Determining the number of batik motif object based on hierarchical symmetry detection approach

Article

Published Version

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

www.reading.ac.uk/centaur

❐   141

Determining the Number of Batik Motif Object based on Hierarchical Symmetry Detection Approach

Ida Nurhaida[1], Remmy A. M. Zen[2], Vina Ayumi[3], Hong Wei[3]
[1,3]Faculty of Computer Science, Universitas Mercu Buana, Indonesia
[2]School of Computing, National University of Singapore, Singapore
[4]Department of Computer Science, University of Reading, United Kingdom

Article Info	ABSTRACT
	In certain conditions, symmetry can be used to describe objects in the batik motif efficiently. Symmetry can be defined based on three linear transformations of dimension n in Euclidian space in the form of translation and rotation. This concept is useful for detecting objects and recognising batik motifs. In this study, we conducted a study of the symmetry effect to determine the number of batik motif objects in an image using symmetry algorithm through a hierarchical approach. The process focuses on determining the intersection line of the batik motif object. Furthermore, by utilising intersection line information for bilateral and rotational symmetry, the number of objects carried out recursively is determined. The results obtained are numbers of batik motif objects through symmetry detection. This information will be used as a reference for batik motif detection. Based on the experimental results, there are some errors caused by the axis of the symmetry line that is not appropriate due to the characteristics of batik motifs. The problem is solved by adding several rules to detect symmetry line and to determine the number of objects. The additional rules increase the average accuracy of the number of object detection from 66.21% to 86.19% (19.99% increase).

Corresponding Author:

Ida Nurhaida
Faculty of Computer Science,
Universitas Mercu Buana,
Jalan Raya Meruya Selatan no. 1 Kebon Jeruk Jakarta Barat 11650.
Email: ida.nurhaida@mercubuana.ac.id

1. INTRODUCTION

There are many variations of batik motifs based on [1]. Batik motifs consist of a geometric pattern and non-geometric pattern. The geometric patterns can be recognised based on the particular shape in horizontal, vertical, and diagonal direction. The non-geometric patterns have free design without any attempt at symmetry. The geometric patterns may be divided into five main classes namely *ceplok, kawung, parang, lereng*, and *nitik*. In contrast, the non-geometric patterns can be grouped into four main categories, *Lung-lungan, Semen, Pagersari*, and *Taplak Meja*. The characteristics of the group are distinctive, respectively.

Batik from Indonesia is the most popular culture as the inspiration in the world. This situation has inspired batik artists to produce many types of batik motif designs. On the other hand, western-style has influenced traditional dresses to be more fashionable. The design may contain several motifs from a different class, colour variation, and pattern combination in one fabric. Batik designers are often placing motif in a new context and making them in the luxury textile. Batik motif counting is essential since the numbers of batik motif found in the fabric or image will be classified into specific classes. Fig. 1 shows the templates sample from each class.

(a). Ceplok Sekar Bangah   (b). Kawung Beton   (c). Parang Rusak Barong

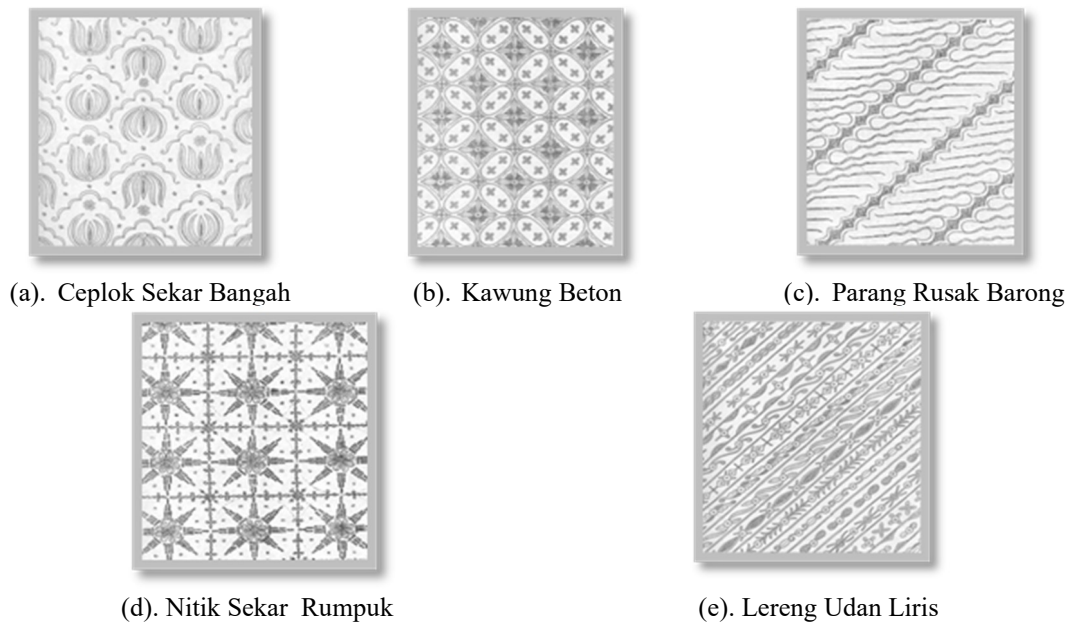(d). Nitik Sekar Rumpuk   (e). Lereng Udan Liris

Figure 1. Primitive motifs from the geometric pattern (source: [1])

The characteristic of features from batik motifs, namely spatial location, scale, rotation, and other transformations, is employed to implement object recognition in Batik Motif Recognition System (BMRS). The primary goal of a BMRS is to identify a batik motif based on its class and to specify the composition of batik motifs in an image. The critical issue in batik motif recognition is to find their initial nature of patterns.

Objects or individual patterns in some batik motifs are highly symmetrical. The symmetrical parts are repeated in several locations. Furthermore, a combination of batik motifs from different classes in one fabric will increase the recognition process's complexity (see Fig. 2). Therefore, it is necessary to analyse each component of the underlying primitive motif within an image. The recognition process in this study is different from the previous researches which are based on texture features. This study aims to determine the number of object occurrences in batik motif images. The property of symmetry of batik motif patterns is utilised for the purpose. This study observed a hierarchical symmetry detection approach to classifying the object motif batik related to its class.

(a). Kawung and Nitik   (b). Kawung and Ceplok   (c). Parang and Kawung
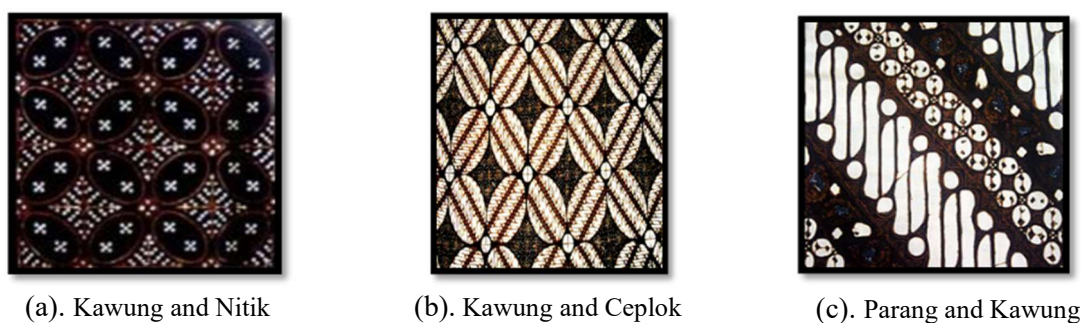
Figure 2. The combination of batik motifs from different classes in batik cloth (Source: Courtesy of Batik Museum Jakarta, Indonesia)

In some cases, symmetry can be used to describe objects in batik motif images more efficiently. For example, if one-half of an object is mirroring to the other half, only one-half needs to be described. Symmetry can be defined regarding two linear transformations in n-dimensional Euclidean space, i.e. translation and rotation. It is a powerful concept that facilitates object detection and recognition in many situations [2]–[4]. This study investigates the symmetry effect in determining the number of objects in batik motif images. Hence, we exploit the symmetry detection algorithm based on [5]. Hereafter, the process focuses on determining intersection lines of an object in a batik motif recursively so that it detects the number of multiple objects [2], [3], as shown in Fig. 3.

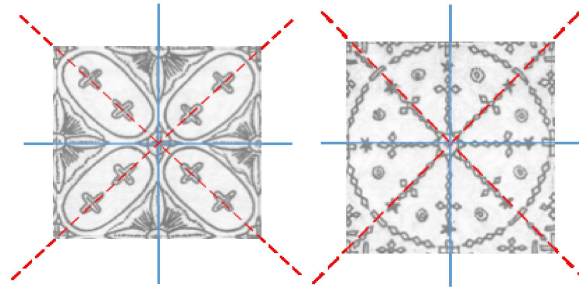Figure 3. Symmetry pattern in batik motifs

There has been an increasing interest in the problem of transformation object in the object recognition area in recent years. The critical aspect of batik motif recognition for the batik motif transformation (rotation, scaling, and translation) may match the image query's batik motif. Batik motif has specific features related to highly symmetric in each part of an object. It can be found in several positions within an image repeatedly. The symmetric property and multiple occurrences of batik motifs that have similar patterns make the recognition process more difficult. The characteristic is illustrated in Figure 3. This problem is the main focus of batik motif recognition in this study. Generally, batik fabric uses a similar motif, but they have differences in terms of position, scale, and direction. Therefore, a reliable method is required for identifying the batik motif. The method must figure out the problem by using the appropriate features that will be employed for batik motif recognition. It should be invariant to translation, scale, rotation, and multiple objects [6][7].

Sanabila et al. [8] employed a template matching based approach. This study detects the batik motif based on the template image by applying the Generalised Hough Transform. The experiment result shows that the Hough transform has a good performance but consuming vast computational resources. The classification and retrieval images build upon texture features have become a concern of several studies.

Nurhaida et al. [9] discuss a batik motif identification method that compared extracted texture features to another batik motif in the Batik recognition task database. Two motif images are selected randomly from the dataset in each class for testing data, while the rest will be used as training data. The distance and the similarity of feature vectors are calculated using the Euclidean distance method. The Grey Level Co-occurrence Metrics as a feature extraction method demonstrate the best result of 80% classification accuracy.

Research conducted by [10] showed that the threshold algorithm gives precision and recall value at 70% and 75% based on the similarity of shape and texture characteristics for batik image retrieval. The threshold algorithm is derived from Fagin Algorithm. Threshold value extracted from the query image will be compared with the threshold's value in all the images in the image database using Euclidean function. The images that have a similar threshold value will be displayed in sequence.

In a batik image retrieval study, [11] acquired the best performance of precision by 74% while recall at 89%. The combination feature of edge orientation and with microstructure descriptor is applied to enhance retrieval performance. The performance comparison between SIFT, voting Hough, and Invariant Generalized Hough Transform (IGHT) has been done to classify particular batik motifs [12]. SIFT earns the best accuracy between 80% - 87%. From this study, it can be proved that SIFT is appropriate for batik motif classification. Research by [13] discusses object detection of batik motif based on Invariant Generalized Hough Transform (the) method. The rotation angle of the batik motifs is computed and stored in the accumulator array. Hill Climbing and low pass filter technique are implemented for smoothing the accumulator array. This study gained accuracy represented by precision and recall 42% and 94% respectively.

Multiple objects detection from images is reported. Research [14] determined the number of multiple objects in a particular class based on voting independently in Hough space. This study revealed the drawback of Hough transform regarding object properties. Research by [15] employed sliding windows for independent features to resolve the drawback and performed votes into probabilistic Hough voting. Both SIFT-like descriptors and Earth Mover's Distance are utilised to detect multiple instances in shorter time [16]. The statistical approach for matching procedure obtained the number of objects based on each query descriptor to the database's variety and size.

To assess whether and how feature fusion is produced and retrieved the batik motif image, we measured the descriptor of Grey Level Co-Occurrence Matrices, Local Binary Pattern, Gabor, and Log-Gabor [17]. Data were gathered from the features extraction process for each image. In summary, these results show that feature fusion outperforms than individual features in this retrieval process. The evidence presented in the study [18] suggests that SFFS methods are used to select the features, whereas the PCA is used to undertake reduction feature on the batik retrieval process.

This paper begins with the feature extraction methods and symmetry detection in Section 2. It will then go on Section 3 that presents the proposed methods. Section 4 analyses the results of symmetry detection and discusses the findings. Finally, this article ends up with a conclusion in Section 5.

2.   FEATURE EXTRACTION AND SYMMETRY DETECTION

2.1. Feature Extraction: Scale Invariant Feature Transform (SIFT)

The real application needs information from interest points for object recognition, object tracking, and image alignment [15]. Image feature proposed by Lowe called Scale Invariant Feature Transform (SIFT) [19]. This technique is based on the selection of the feature which represented specific interest points. The interest points are invariant to the position, scale and orientation. The object recognition is executed using local descriptors around an interesting point.

Two pairs of matching keypoint are compared from query image against to template image using the nearest neighbour technique [20] [21] [22]. The comparison between the smallest distance and the second smallest distance must be below the threshold. The threshold ratio 0.8 is recommended by Lowe for point matching. The descriptor is 128-dimension which can be used for feature matching. In particular, the strategy was problematic since it requires demanding computational power and memory consumption. This condition is the constraint of SIFT for implementation in a real-time with large numbers of images.

2.2. Symmetry Detection

in this study, symmetry detection based on the matching features of the symmetric point [5]. We choose this algorithm because it can conduct multiple symmetry detections from a non-segmented real image. The numbers of matching symmetry point show the related position (x,y coordinate), scale (σ) and orientation (θ) of the pair features. The Hough space is utilised to store the accumulation number of the vote of potential symmetry within an image. The symmetric grouping of features is carried out for symmetry detection in the image plane. Each pair of feature points indicated the numbers of symmetry keypoint. Potential mirror-symmetric matches can be resulted from a build up a collection of mirrored feature descriptors and matching these against the native feature descriptors. Mirrored feature descriptors are defined as descriptors of mirrored copies of the local image patches associated with the original feature points [23].

Matching pairs are described reflected bundled features of features point. Symmetry will be calculated immediately from these pairs of matching vectors. In this study, the method proposed by Loy et al [5] was implemented in Matlab, with feature points detected and described using Lowe's SIFT code and applied to identify bilateral and rotational symmetries in batik motif images. Bilateral symmetry refers to one side is the mirror image of the other side. They essentially look just like their mirror images. However, a bit of rotating may be necessary to superimpose the two shapes precisely upon each other if the proper mirror plane is not chosen. The rotational symmetry likely looked the same after being rotated by 180° around its centre. However, it does not have mirror symmetry. Batik motif patterns appear in both rotational and bilateral symmetries.

3.   RESEARCH METHOD

3.1. Dataset

The experiment was conducted on a dataset including 211 motif templates (each template has 150x150 pixels) generated from batik images. We generated this dataset using templates from [1]. These templates were cropped into primitive patterns. The images were first grouped into five groups based on the motif composition. Each primitive pattern is formed into images with some symmetry objects from 2 (1 x 2; 2 x 1), 4 (2 x 2), and 8 (4 x 2; 2 x 4). The total number of generated query images is 1,055, and they have a size of from 150 x 300 pixels to 600 x 600 pixels (see Table 1).

Table 1. The Generated Dataset Batik Motif

No.	Motif Composition (row x column)	Number of data set	Size (height x width)
1	1 x 2	211	150 x 300
2	2 x 1	211	300 x 150
3	2 x 2	211	300 x 300
4	2 x 4	211	300 x 600
5	4 x 2	211	600 x 300
T o t a l		1.055	

The query images are generated based on a composition object in a regular pattern. In this experiment, the data set only for a primitive pattern from a geometric pattern. Based on [1], the geometric pattern consists of five

major geometric Indonesian batik motifs from class Ceplok, Kawung, Nitik, Lereng, and Parang. We only used five class of geometric pattern based on its classification in this experiment [1]. Examples of each motif compositions can be seen in Figure 4.

3.2. Pre-processing

In this experiment, we performed pre-processing steps before going into feature extraction. We did the pre-processing steps in this experiment: converting images to the greyscale level and reducing the images' resolution. We used greyscale images instead of colour images because we only focused on geometric feature and feature extraction in this experiment: SIFT and Symmetric. Besides, we used grayscale images (as depicted in Fig. 4) to prevent heavy computation due to colour features and because the SIFT algorithm is already greedy.
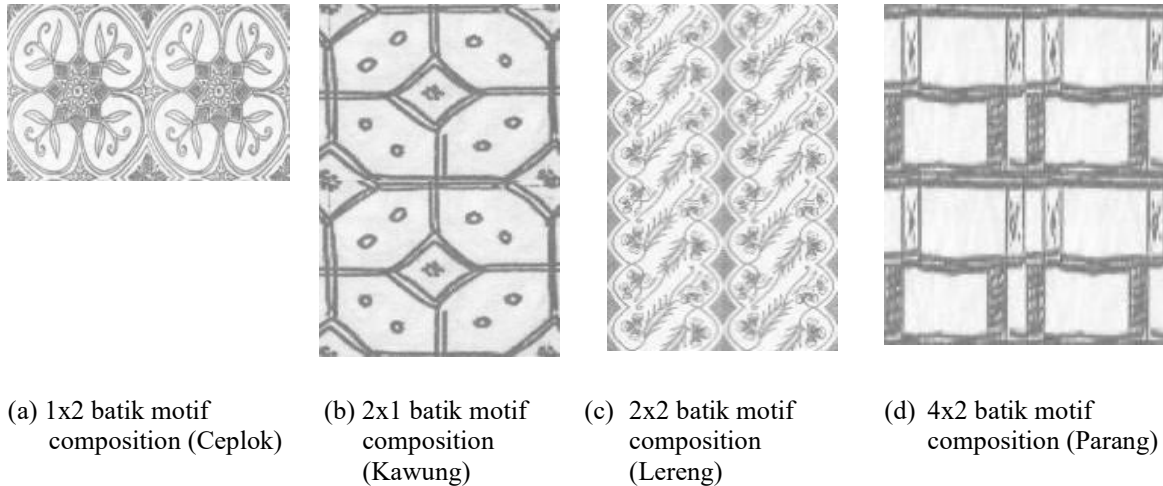
| (a) 1x2 batik motif composition (Ceplok) | (b) 2x1 batik motif composition (Kawung) | (c) 2x2 batik motif composition (Lereng) | (d) 4x2 batik motif composition (Parang) |

Figure 4. Batik motif in several compositions

3.3 Hierarchical Symmetry Detection Approach

The proposed algorithm is intended to find the nature of the batik motif. We call it *template motif.* The samples of template motif with symmetry can be seen in Figure 5 namely bilateral symmetry (Fig. 5.(a)), rotational symmetry (Fig 5.(b)) and both bilateral and rotational symmetry (Fig. 5.(c)).

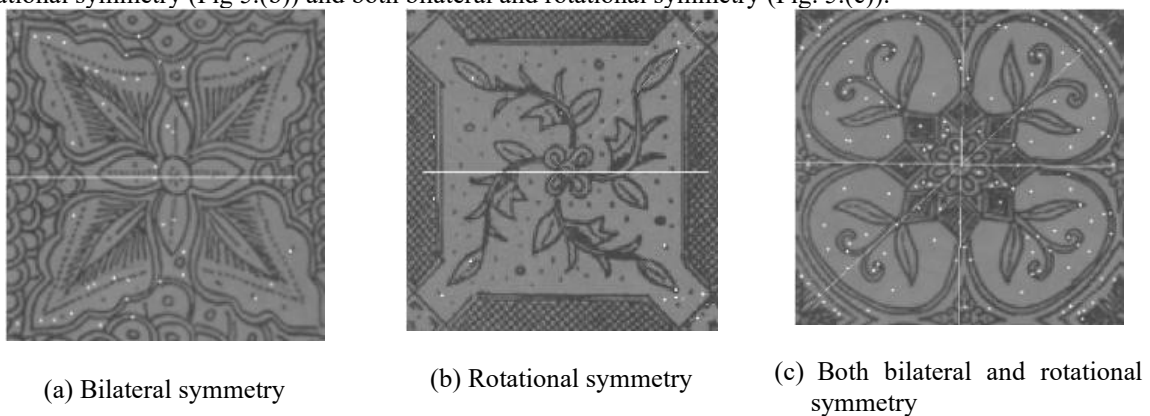| (a) Bilateral symmetry | (b) Rotational symmetry | (c) Both bilateral and rotational symmetry |

Figure 5. Symmetry in template motif

The template motifs are stored in the database. The algorithm takes a batik image as an input and determines the number of template motif occurrences in the image. The algorithm works recursively. First, the algorithm will verify the size of the input image. If it has equal size with template motif, then the algorithm will result in only one object found. However, if the input image's size is bigger than the template motif size, then we apply symmetry detection to the image using the symmetry function provided by [5].

Figure 6 describes the result of the symmetry detection of batik image. We use the image in Figure 6.(a) as a template and created a 2 x 1 (row x column) input image in Figure 6.(b). Figure 6.(c) show the result of symmetry function by [5].

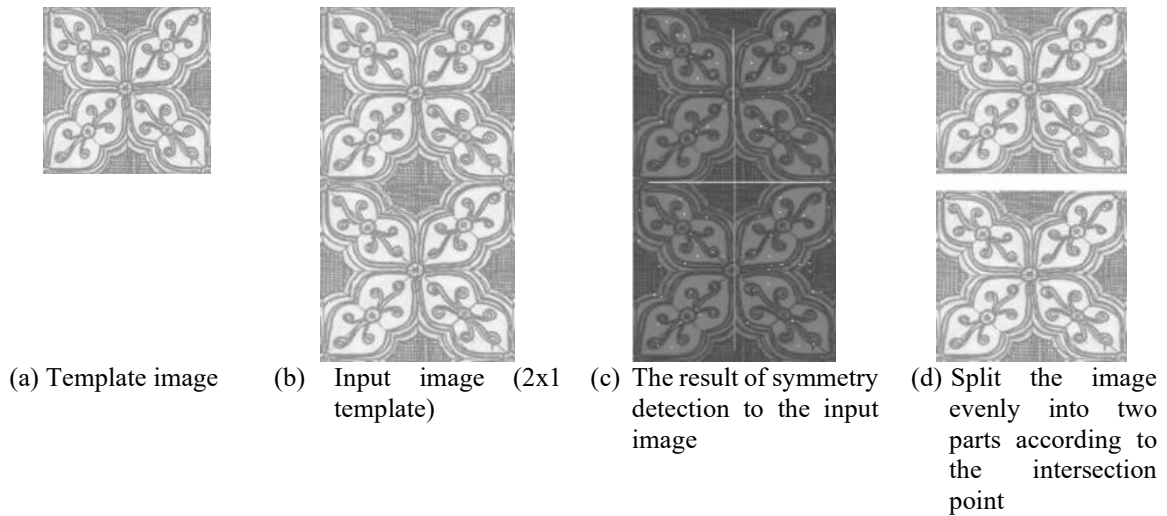|			
(a) Template image	(b) Input image (2x1 template)	(c) The result of symmetry detection to the input image	(d) Split the image evenly into two parts according to the intersection point

Figure 6. Symmetry Detection Result

The dots indicate the key-points, while the white lines indicate the symmetry line. We modified the function, so it returns two coordinates ($[x_1, y_1]; [x_2, y_2]$) at the end of each symmetry line found. We can get the equation of a line from two points by using:

$$y = mx - mx_1 + y_1 \tag{1}$$

Where m is the gradient. Then, we can find the intersection point between two lines equations by solving equation:

$$\text{x:} \begin{bmatrix} -m_1 & 1 \\ -m_2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} -m_1 x_{1,1} + y_{1,1} \\ -m_2 x_{1,2} + y_{1,2} \end{bmatrix} \tag{2}$$

Where m_i means gradient from equation i, $x_{i,j}$ means x_i from equation j, and $[x_0, y_0]$ is the intersection point.

In case the input image width is the same as the input image height, we use the intersection point to crop the image into four parts evenly, and each image becomes the input of this function recursively. If the input image width differs from the height, this is the case where the number of the object is 2 or 8, so we divided the image into two parts evenly. In case there is no key point found, the algorithm will be terminated. The final value returned from this function is the number of template occurrences in the image. Pseudocode of the algorithm explained above can be seen in Algorithm 1.

Algorithm 1

```
func split_rec (image)
        if image size equals template image size
                return 1
        end
        lineq = symmetry(image)
        if lineq is empty
                return 0
        end
        intersect = find_intersection(lineq)
        object = 0
        if image width equals image height
                object = object + split_rec(upper left image)
                object = object + split_rec(upper right image)
                object = object + split_rec(lower left image)
                object = object + split_rec(lower right image)
        elseif image width greater than image height
                object = object + split_rec(left image)
                object = object + split_rec(right image)
        elseif image height greater than image width
                object = object + split_rec(upper image)
                object = object + split_rec(lower image)
        end
        return object
endfunc
```

4. RESULTS AND DISCUSSION

We evaluate the algorithm by matching the number of objects returned by the algorithm with the image's real number of objects. If the number of objects is equal, we say the answer is correct or 1 and wrong or 0 otherwise. Then, we can get the accuracy with the following equation:

$$accuracy = \frac{\sum correct images}{number of images} * 100 \qquad (3)$$

The results matrix grouped by the motif composition and class can be seen in Table 2.

Table 2. Accuracy by Motif Composition

Composition Class	1 x 2	2 x 1	2 x 2	2 x 4	4 x 2	Average
Ceplok	80.21%	94.79%	94.79%	44.79%	71.88%	77.29%
Kawung	94.12%	88.24%	100%	35.29%	88.24%	81.18%
Lereng	61.90%	85.71%	40.00%	18.18%	14.29%	44.02%
Nitik	76.19%	97.62%	98%	30.95%	78.57%	76.19%
Parang	85.29%	52.94%	64.71%	29.41%	29.41%	52.35%
Average	79.54%	83.86%	79.42%	31.73%	56.48%	**66.21%**

Figure 7 demonstrates the graphic of average accuracy divided by the motif composition and class. Overall our method gave 66.21% accuracy. As we can see in Figure 7 (a) that the accuracy slowly decreased as the number of objects in the image increase. It is trivial that detecting symmetry in small images with a few objects is easier than that in larger images with many objects. From the chart, it can be seen that Lereng motif gave the lowest accuracy. This is due to the nature of Lereng motif that does not have a clear symmetry pattern and more complex than other motifs.
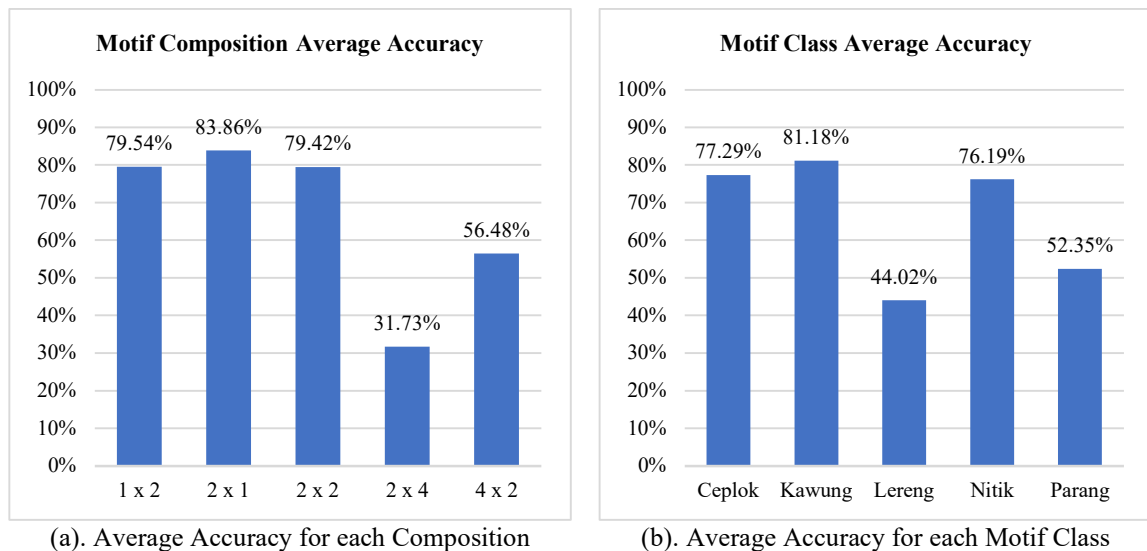

(a). Average Accuracy for each Composition   (b). Average Accuracy for each Motif Class

Figure 7. Average Accuracy for Motif Composition and Motif Class

Error analysis was carried out, and four aspects were taken into account as follow:
1) Symmetry Detection Function by [5] gave symmetry lines that are not the symmetry of the whole image but symmetry lines for each batik template. In this case, it will split the template, not the whole image. These errors occurred mostly because the batik template data itself are symmetry by nature (Fig 8.a).
2) The intersection coordinate could not be found because no symmetry lines are intersecting with each other. In other words, symmetry lines formed a parallel line (Fig 8.b). The error also occurred when the intersection coordinate located outside the image boundaries. If the intersection coordinate could not be found, the process stops and return object found as zero.
3) There are also cases that only one-line equation found. Therefore, no intersection can be found to split the image (Fig 8.c).
4) The symmetry lines are not centred, leading to a false splitting process and giving a wrong number of objects (Fig 8.c).

Visualisation of each error respectively are shown in Fig. 8.

(a). Symmetry Lines not for the
     whole object (Case 1)

(b). Symmetry lines form a
     parallel line (Case 2)

(c). Only One Symmetry Line and
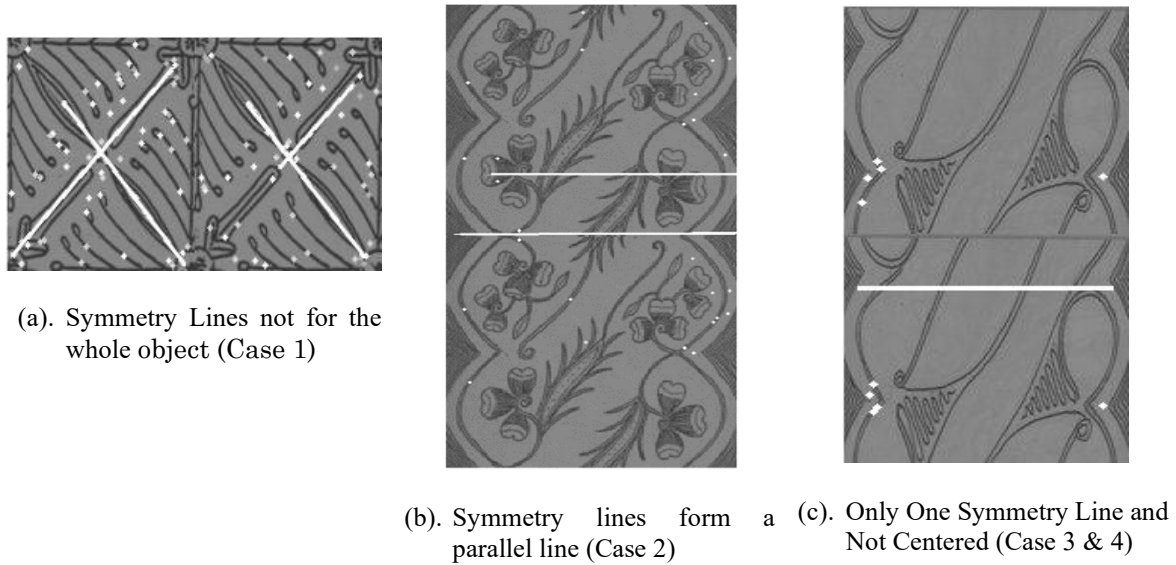   Not Centered (Case 3 & 4)

Figure 8. Visualisation of Errors in the Experiments

The errors in Cases 2 and Cases 3 can be handled with an additional rule: if no intersection coordinate or only one line found, then take the main line equation and take the midpoint from that equation as the intersection point wanted. This additional rule improves accuracy significantly. The algorithm with this new rule is presented in Algorithm II (see Algorithm 2). However, the error can still happen if the rule chooses the wrong line equation. For example, in Figure 8 (b), if the main equation line is the top one, we will have a Case 4 error and a wrong result. To deal with this, we simply stop the algorithm and say that two objects are found.

Algorithm 2

```
func split_rec (image)
        if image size equals template image size
                return 1
        end
        lineq = symmetry(image)
        if lineq is empty
                return 0
        end
        intersect = find_intersection(lineq)
        // the new rule to handle case 2 and case 3 error
        if lineq length == 1 or intersect == null
                intersect = midpoint(lineq[0])
        end
        object = 0
        if image width equals image height
                object = object + split_rec(upper left image)
        object = object + split_rec(upper right image)
                object = object + split_rec(lower left image)
object = object + split_rec(lower right image)
        elseif image width greater than image height
object = object + split_rec(left image)
object = object + split_rec(right image)
        elseif image height greater than image width
object = object + split_rec(upper image)
object = object + split_rec(lower image)
        end
        return object
endfunc
```

The results with the additional rule applied are presented in Table 3. The improved method achieves 86.19% accuracy. It dramatically increases the accuracy by 19.99% compared to the original method. Ceplok,

Kawung and Nitik motif classes always have high accuracy over 95%. Furthermore, Parang and Lereng motif classes have relatively poor performance with an average accuracy of 75.88% and 67%, respectively.

Table 3. Accuracy by Motif Composition (after Improvement)

Composition Class	1 x 2	2 x 1	2 x 2	4 x 2	2 x 4	Average
Ceplok	98.96%	100.00%	98.96%	86.46%	92.71%	95.42%
Kawung	100.00%	94.12%	100%	88.24%	100%	96.47%
Lereng	90.48%	95.24%	56.00%	40.91%	52.38%	67.00%
Nitik	100.00%	100.00%	98%	90.48%	92.86%	96.19%
Parang	94.12%	100.00%	85.29%	44.12%	55.88%	75.88%
Average	96.71%	97.87%	87.57%	70.04%	78.77%	**86.19%**

Table 3 shows that the 2x1 and 1x2 motif compositions always receive high average accuracy compared to other motif compositions. These two motif compositions obtain an average accuracy of 96.71% and 97.87%, respectively. Moreover, the composition 4x2 has the lowest average accuracy of 70.04%. The average accuracy of each composition is getting worse when the number of motifs in an image is increased. The main problem is that the symmetric lines cannot separate the objects in the image precisely. Thus, the algorithm cannot detect the number of objects correctly. Figure 9 presents the details of the experimental result for each motif composition.
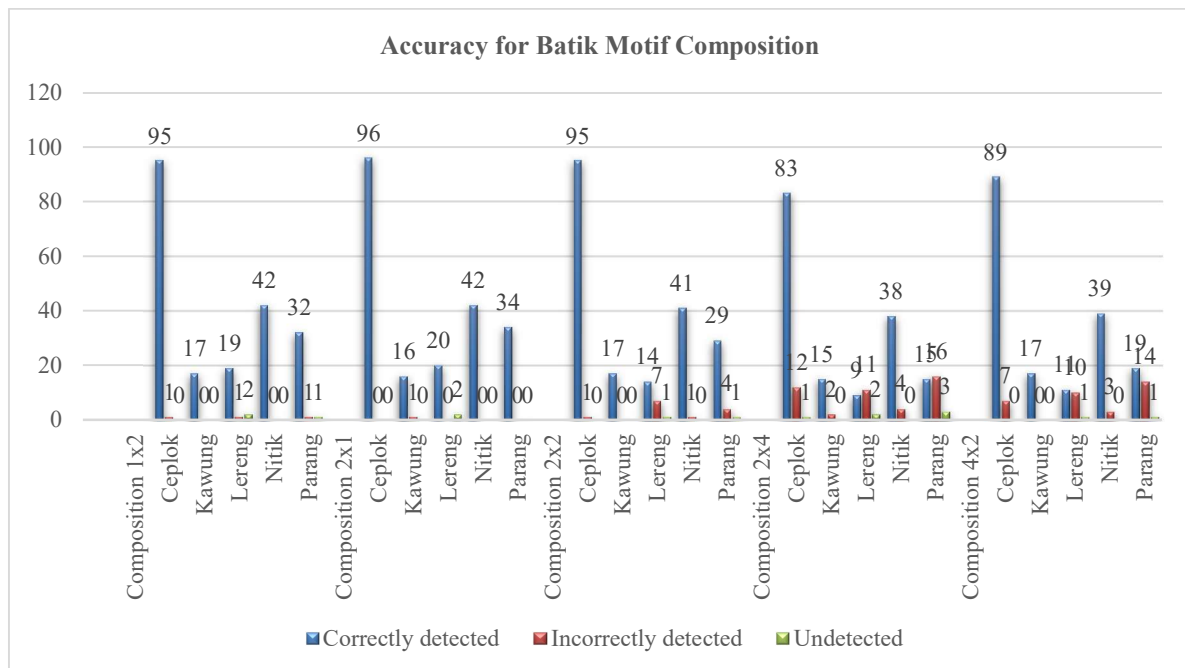

Figure 9. Experiment result based on numbers of batik motif with composition 1x2, 2x1, 2x2, 2x4 and 4x2

In the 1x2 motif composition, Kawung and Nitik motifs have the highest accuracy of 100%. Meanwhile, there is only one image that was incorrectly detected in Ceplok, Parang and Lereng motifs. The current study found that the error had occurred in Parang and Lereng motif symmetry detection. Therefore, the number of objects cannot be correctly detected. For a 2x1 motif composition, the performance is good.

The proposed method is only incorrectly detected in one image of the Kawung motif and two Parang motif images. In contrast to earlier findings, Parang and Lereng motif has low accuracy in a 2x2 composition. The symmetry line in this composition cannot separate the objects precisely so that the number of objects is incorrectly detected. In these motif compositions, the symmetric lines are hard to detect so that the objects are sometimes entirely undetected. In 2x4 and 4x2 motif compositions, the number of images that incorrectly detected is increased. The generated symmetric lines are mainly parallel to each other, and it does not divide objects in the images correctly. In this case, the additional rule is applied to find the keypoint as the intersection point for determining the number of objects. However, Lereng and Parang classes have relatively poor accuracy compared to the other class.

Motif Composition Average Accuracy

96.71% 97.87% 87.57% 70.04% 78.77%

1 x 2   2 x 1   2 x 2   2 x 4   4 x 2

(a). The average accuracy for each composition using an improved method

Motif Class Average Accuracy

95.42% 96.47% 67.00% 96.19% 75.88%

Ceplok Kawung Lereng Nitik Parang

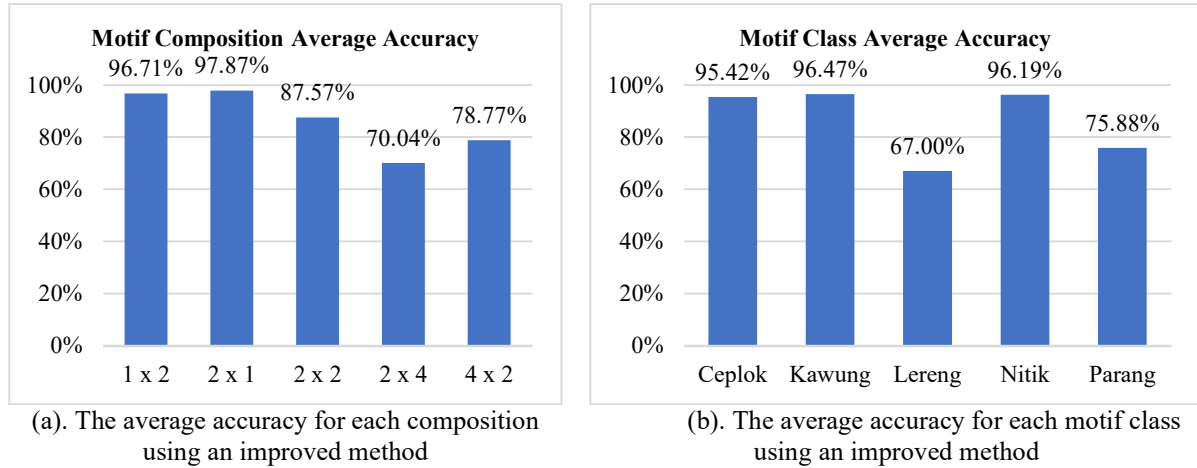(b). The average accuracy for each motif class using an improved method

Figure 10. Average Accuracy on Respective Motif Composition and Motif Class

As shown in Figure 10, the results indicate that the main reason is the ornaments in Lereng class is quite excellent, dense, and has stripes component. Lereng and Parang classes mainly have a diagonal pattern that is hard to divide the objects evenly. In some Parang and Lereng motif images, numerous parallel symmetric lines (horizontal and diagonal) are detected, but they do not represent the boundary lines of the objects. Thus, the number of batik motifs in an image cannot identify correctly. Moreover, the number of symmetric lines in some images is not sufficient to separate objects in an image. Therefore, the proposed algorithms only identify a smaller number of objects. The error of symmetric line detection can be seen in Figure 11.
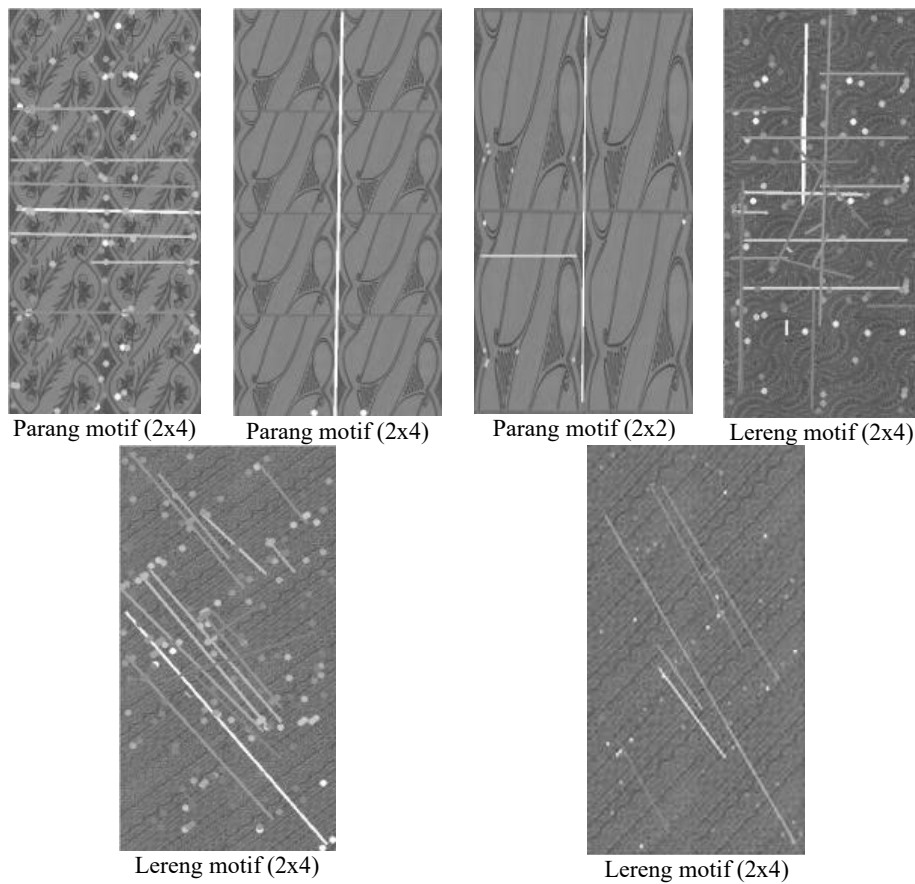
Parang motif (2x4)   Parang motif (2x4)   Parang motif (2x2)   Lereng motif (2x4)

Lereng motif (2x4)   Lereng motif (2x4)

Figure 11. Symmetric Line Error in Parang and Lereng Motif

5. CONCLUSION

This research presented different methods to detect the number of batik motif objects based on symmetry detection using a hierarchical approach. With this method, it is possible to detect local as well as global symmetries. The method can also detect the number and position of the axes of symmetry. Scale, rotational and partial symmetry has been detected successfully. Our method provides a top-down hierarchical approach for estimating the symmetry structure. The algorithm performs well for detection in highly curved or distorted symmetry structures based on batik motif class.

On detecting the symmetric object in an image, we face some problem that can cause the object's number not to be determined and identified. We proposed an improvement to our previous method by putting the additional rule. Based on the experiment, it brings an improvement of up to 19.99%. However, Parang and Lereng motif has lower accuracy because they have a dense ornament, tiny stripes, and diagonal pattern. A motif with a diagonal pattern is hard to determine the object's number because it will form a diagonal symmetric line. On the contrary, Kawung, Nitik and Ceplok with a circle pattern, have better accuracy.

Based on the experiment result, the proposed approach is more suitable to detect a circular pattern rather than a diagonal pattern. Proposing a new rule to mitigate the problem with diagonal pattern and detecting multiple batik templates in one image can be a future direction for this work.

ACKNOWLEDGMENT

REFERENCES

[1] S. S. Samsi, *Techniques, Motifs, Patterns Batik Yogya and Solo*. Titian Foundation, 2011.
[2] C. Sun and J. Sherrah, "3D Symmetry Detection Using The Extended Gaussian Image," *164 IEEE Trans. PATTERN Anal. Mach. Intell.*, vol. 19, no. 2, pp. 164–169, 1997.
[3] K. Kim and S. Lee, "Computerized Medical Imaging and Graphics Vertebrae localisation in CT using both local and global symmetry features," *Comput. Med. Imaging Graph.*, vol. 58, pp. 45–55, 2017.
[4] M. Mignotte, "Symmetry detection based on multiscale pairwise texture boundary segment interactions ☆," *Pattern Recognit. Lett.*, vol. 74, pp. 53–60, 2016.
[5] G. Loy and J. Eklundh, "Detecting Symmetry and Symmetric Constellations of Features," in *ECCV'*, 2006, pp. 508–521.
[6] Z. Liu, S. Bu, and J. Han, "Locality-constrained sparse patch coding for 3D shape retrieval," *Neurocomputing*, vol. 151, pp. 583–592, 2015.
[7] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, and C. Gao, "Object Class Detection: A Survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 10:1–10:53, 2013.
[8] H. R. Sanabila and M. Manurung, "Recognition of Batik Motifs using the Generalised Hough Transform," in *Proceedings of the International Conference on Advanced Computer Science and Information Systems, (ICACSIS)*, 2009, pp. 1–6.
[9] I. Nurhaida, M. Manurung, and A. M. Arymurthi, "Performance Comparison Analysis Features Extraction Methods for Batik Recognition," in *International Conference on Advanced Computer Science and Information Systems, Depok*, 2012.
[10] A. H. Rangkuti, R. B. Bahaweres, and A. Harjoko, "Batik Image Retrieval Based on Similarity of Shape and Texture Characteristics," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2012, pp. 978–979.
[11] A. E. Minarno, Y. Munarko, F. Bimantoro, A. Kurniawardhani, and N. Suciati, "Batik image retrieval based on enhanced micro-structure descriptor," in *2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE), Bali*, 2014, vol. 1, pp. 65–70.
[12] R. Akta, "Batik Motif Classification using Scale Invariant Feature Transform method," Thesis, Universitas Indonesia, 2012.
[13] A. A. Bouty, "Accumulator Array Extraction for Rotation Measurement using Hill Climbing and Low Pass Filter in Batik Digital Image," Thesis, Universitas Indonesia, 2012.
[14] G. Aragon-Camarasa and J. P. Siebert, "Unsupervised clustering in Hough space for recognition of multiple instances of the same object in a cluttered scene," *Pattern Recognit. Lett.*, vol. 31, no. 11, pp. 1274–1284, Aug. 2010.
[15] P. Yarlagadda and A. Monroy, "Voting by Grouping Dependent Parts," in *ECCV'10 Proceedings of the 11th European conference on Computer vision: Part V*, 2010, pp. 97–210.
[16] J. Rabin, J. Delon, and Y. Gousseau, "A Statistical Approach to the Matching of Local Features," *SIAM J. Imaging Sci.*, vol. 2, no. 3, pp. 931–958, Jan. 2009.
[17] I. Nurhaida, H. Wei, R. A. M. Zen, R. Manurung, and A. M. Arymurthy, "Texture Fusion for Batik Motif Retrieval System," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 6, pp. 3174–3187, 2016.
[18] H. Fahmi, R. A. M. Zen, H. R. Sanabila, I. Nurhaida, and A. M. Arymurthy, "Feature Selection and Reduction for Batik Image Retrieval," in *Proceedings of the Fifth International Conference on Network, Communication and Computing (ICNCC)*, 2016.
[19] I. Nurhaida, H. Wei, Z. Remmy Augusta Menzata, M. Manurung, and A. M. Arymurthi, "Texture Fusion for Batik

Motif Retrieval System," *Int. J. Electron. Comput. Eng.*, vol. 6, no. 6, 2016.

[20] M. J. B. Wilhelm Burger, *Priciples of Digital Image Processing*. Springer-Verlag London Limited, 2013.

[21] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[22] Y. Sun, L. Zhao, S. Huang, L. Yan, and G. Dissanayake, "L2-SIFT: SIFT feature extraction and matching for large images in large-scale aerial photogrammetry," *ISPRS J. Photogramm. Remote Sens.*, vol. 91, pp. 1–16, 2014.

[23] N. Bakiah *et al.*, "SIFT-Symmetry : A robust detection method for copy-move forgery with reflection attack q," *J. Vis. Commun. Image Represent.*, vol. 46, pp. 219–232, 2017.

BIOGRAPHY OF AUTHORS

Ida Nurhaida was born in Kuantan (Malaysia) in 1971. She is a researcher and member of Faculty of Computer Science Universitas Mercu Buana, Indonesia. Her areas of interest and research are in the image processing, pattern recognition and image retrieval system. She formalised in her PhD (2010) on this subject in the Faculty of Computer Science, University of Indonesia. She has presented papers at conferences both home and abroad, published articles and papers in various journals, and contributed a chapter to the book.

Remmy A. M. Zen was born in Jakarta, Indonesia, in 1991. He received his Bachelor Degree in Computer Science from Universitas Indonesia in 2013 and a Master Degree in Computer Science from Universitas Indonesia in 2014. He is currently a PhD candidate at the School of Computing, National University of Singapore (NUS). His research interest on the quantum machine learning topic, specifically solving a quantum many-body problem with machine learning.

Dr Hong Wei received her first and Master degrees from Tianjin University, China, in 1983 and 1986, respectively, and the PhD degree from Birmingham University, UK, in 1996. She is author/co-author of over 60 refereed research papers and two text books. She is currently an associate professor in the School of Systems Engineering, University of Reading, UK. Her research areas cover biometrics, image-based pattern classification, and computer vision.

Vina Ayumi is lecturer of Computer Science in Universitas Mercu Buana, Indonesia. She received master degrees from Faculty of Computer Science, Universitas Indonesia. Her research interests are Machine Learning and Computer Vision. She can be contacted at e-mail address: vina.ayumi@mercubuana.ac.id.