Journal of Imaging Science and Technology® 66(3): 030506-1–030506-13, 2022.

A Fast Fourier Transform with Brute Force Algorithm for Detection and Localization of White Points on 3D Film Pattern Images

John Mlyahilu and Jongnam Kim
Department of IT Convergence and Application Engineering, Pukyong National University, Busan 48513, Korea
E-mail: jongnam@pknu.ac.kr

Abstract. *The procedures of white points detection and localization are practically complex on noisy images. In this paper, we propose an algorithm that detects and localizes white points on 3D film images. The proposed algorithm uses the fast Fourier transform to convert the binarized image into real and imaginary parts to obtain the number of white points along the horizontal and vertical. We determine the sorted coordinates of the white points by adding a brute-force solution to the coordinates obtained from the real part of the image. These sorted coordinates are obtained by subtracting the error between the Euclidean distances of the normalized coordinates along the vertical and horizontal direction. The proposed algorithm with and without brute-force achieved an average detection ratio of 0.98 and 0.88 respectively, while the others underperformed. We perform various experiments using the existing algorithms such as template matching, thresholding, and an iterative method to validate the performance of our algorithm. We also compare the rule-based algorithms that detect and localize objects in noisy images with the proposed one to determine the reliability of our algorithm. The experimental results indicate that the proposed algorithm performs better than the template matching, thresholding, and iterative algorithm.* © *2022 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2022.66.3.030506]

1. INTRODUCTION

Industrially produced tools need an ad hoc observation because they may harm personnel during operations. Close observation of these tools can permit the determination of their quality and use. The necessity to determine their attributes allows the user to classify them as either good or bad, proper or improper, perfect or imperfect, genuine or fake, etc. 3D films are one among many industrially produced tools that have multiple applications. Academically, they are used for enhancing learners about the depth perception of certain concepts presented through images [1]. They are also used for medical devices when tracing bone fragments in injured persons. They are also used in construction fields for 3D reconstruction of demolished buildings by simulations [2]. In general, they are used to bridge the gap between human thinking and real-life situations. 3D printing tools are realized in all undertakings due to their speed in production, flexibility in implementation, and cost-benefit analysis.

Received July 21, 2021; accepted for publication Dec. 3, 2021; published online Jan. 24, 2022. Associate Editor: Doo-Hyun Choi.
1062-3701/2022/66(3)/030506/13/$25.00

White point detection problems on 3D films are not common in the field of pattern recognition. The most common problems stated in [3–5] include coin detections and counting, detection of white cancer spots on the skin, detection of multiple bright spots in an image, white dots on black background, etc. These problems can practically be solved by using edge detection methods, thresholding techniques, color texture methods among others. Moreover, there are various occasions where the mentioned techniques fail to detect the designated object in an image. In this paper, we present a problem concerning the detection of white points on 3D films, where most of the rule-based methods fail to achieve the desired goal. This problem involves the detection and localization of white points on the entire image. After points localization, for every opposite and adjacent corner points we crop the recognized patterns for future use in solving either machine learning or deep learning problems. We also propose an algorithm that solves the problem as shown in Figure 1.

Fig. 1, represents a 3D film image that consists of white points to be detected and localized by the proposed algorithm. The white points are the intersections of the rectangles that bear a star-like shape in the image. Additionally, the image has varying intensities, making it to appear faded or shiny at different points. Moreover, there is no clear distinction between the points to be detected (object) and the scene (background). Based on these limitations, the presented problem cannot be easily addressed by most of rule-based algorithms.

Various algorithms are developed to address computer vision problems similar to the one at hand. Of many approaches, template matching, contour, edge detection, and iterative methods are popular frameworks for detecting and localizing blobs on images [6]. Recently, various template matching-based algorithms have been published in the field of image processing such as white point detection and white balance for color images [7], adaptive white point extraction based on dark channel prior for automatic white balance, and method and computer system of white point detection [8, 9]. These algorithms are popular because of their accurate prediction quality, reduced mathematical complexities, and simple programming implementation. Even though these algorithms have been reported to have outstanding performance on detecting and localizing white

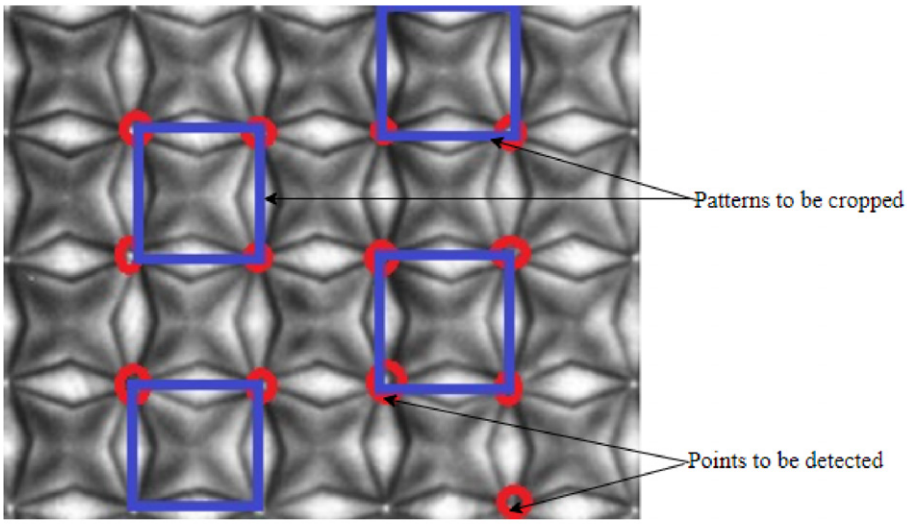

Figure 1. White points to be detected on a 3D film and the patterns to be recognized.

points, they seem to be disadvantageous when applied to highly contrasted images [10]. Use of these algorithms in noisy images can result in poor detection and localization of the white points that may lead to serious problems in devices or tools manufacturing.

We are interested in developing an automated system that assesses and evaluates a 3D film printing robot that designs various patterns and prints them on canvases for spacecraft, airplanes, modern houses, car interior design and decoration. For the printed canvases, there might be both regular and irregular patterns that lead to the acceptance or rejection of the printed 3D films. In this study, we develop and implement an automated multi-camera system with four cameras that assess and evaluate four 3D films as per standards. The assessment deals with every individual pattern in a large 3D film based on their histograms of the intensity pixel values. The evaluation deals with the whole large 3D film by classifying it as either good or bad provided that there exists single or many patterns to make it being accepted or rejected by the producer. In this paper, we eliminate the varying contrast problem by applying histogram equalization and adaptive thresholding. We also eliminate the varying column width problem using a fast Fourier algorithm that localizes the detected white points on a 3D film. The image processing techniques eliminate the noise on an image and detect the white points, and the fast Fourier approach with brute-force algorithm localizes the white points based on a distance filter (d) and minimum error of the euclidean distance between the normalized coordinates and the rounded off coordinates which is subtracted from the coordinates obtained through the fast Fourier. The brute-force algorithm lists all the possible candidates of the coordinates from the unsorted coordinates based on the distance filters minus the minimum error from the euclidean distances. From the combined algorithms, we further detect and localize the white points on the 3D films. The proposed algorithm detects and localizes the white points using the coordinates which are used to crop the recognized pattern even though the lengths of the patterns are irregular. Based on camera calibrations, the robot can produce regular and irregular star-like rectangles in a large 3D film whose length and width are not uniform. Thus, in this paper, we are interested in determining the points of intersections of every star-like rectangle in the entire image. The rest of the procedures and evaluation will be presented in the future versions of this paper with both rule-based, machine learning, and deep learning algorithms.

This paper is organized as follows. In Section 2, we describe the related works that are based on the methods for the detection and localization of white points. In Section 3, we describe the proposed algorithm, which is based on histogram equalization, adaptive thresholding, and a fast Fourier transform. In Section 4, experimental results for various rule-based algorithms are discussed and followed by the conclusion in Section 5.

2. RELATED WORKS

The easiest method to address white point detection problems is the template matching algorithm. This method requires a predefined feature to be used for detection and localization [11]. This category of algorithms use the spatial relationship of the template and the input for matching. They work on the assumption that the environments are always standard and constrained such that the datasets are free from noise [12]. Additionally, they allow easy computations because they use correlational procedures in their implementations. Even though they have promising performances in addressing these problems, their performance is questionable when subjected to noisy datasets [13].

To address the limitations experienced by the template matching algorithm, the histogram of the oriented-gradients algorithm is required. It is a technique that counts the frequencies of the gradient orientation in the localized segments of the input image [14]. This method works as similar to scale-invariant feature transform and edge-oriented histograms. This method performs better than the template matching algorithm because it uses uniformly spaced cells and normalized local contrast values to detect the objects on an image. The algorithm has a lot of advantages but is impractical for rotated images.

For fixed object sizes with known shapes, the iterative algorithms are suitable. The iterative algorithms are mathematical procedures that use an initial value to generate a sequence of approximate solutions for a problem in which the nth approximation is derived from the previous approximations [15]. They perform better in recognizing patterns that have invariant column width sizes [16]. These algorithms are not easily affected by noisy images because they require a proper assignment of the initial point. Iterative algorithms can outperform template matching algorithms on object detection tasks depending on the nature of the problem. These algorithms are also not suitable for rotated images.

The other algorithm discussed in this excerpt is Hough transform algorithm that finds imperfect instances within a class of shapes by a voting procedure [17]. This procedure requires a parameter space from which the candidates are located as local maxima that explicitly computes the transformation. The advantages of this algorithm are; it detects lines, shapes, and objects using pixels lying on one line. The algorithm performs better in detecting objects however, it can miss the object caused by the misleading of multiple lines aligned on one another. Additionally, the mislead can be caused by the detected lines being infinite lines as described by the image size and other parameters, rather than finite lines with defined endpoints [18].

Of late, the most trending algorithms for objects detection and pattern recognition are the neural networks. Neural networks are mathematical models that use learning algorithms inspired by the brain to store information [19]. They use a set of neurons with mathematical computations for feature extraction to determine the expected pattern of the object [20]. The algorithms are used widely in science and engineering because of their efficiency and reliability in performing various computer vision tasks. Neural networks offer many advantages, including requiring less formal statistical training, the ability to implicitly detect complex nonlinear relationships between dependent and independent variables, the ability to detect all possible interactions between predictor variables, and the availability of multiple training algorithms [21]. The main disadvantage is that they are prone to overfitting, computational complexities, and they are black-box in nature. Additionally, they require a lot of data for the network to learn from the inputs for making correct and proper predictions.

Other popular methods to solve this problem are edge detection methods and thresholding methods. These methods aim to identify points on an image at which the brightness changes sharply [22]. These methods are simple to implement but they are sensitive to noise images and tend to be inaccurate. Additionally, the selection of the threshold is not always straightforward, as these methods involve a lot of trial and error. Of all the mentioned methods, including the ones to be used in the experimental section, they seem to be limited in solving this problem. Based on the existing drawbacks for the discussed algorithms, this paper suggests a fast Fourier transform(FFT) algorithm that detects and localizes white points on a binarized 3D film image with an addition of a brute force algorithm. The essence of the brute-force is to calculate the minimum error of the euclidean distance between the normalized coordinates and the rounded-off normalized coordinates. The proposed algorithm detects the white points from the binary images in the Fourier domain as explained in the next section.

3. PROPOSED ALGORITHM

The proposed algorithm stems from the challenge that the existing rule-based methods still face some challenges for detecting and localizing objects on noisy images. To tackle the drawback mentioned thereof, we propose an algorithm that detects all the white points along the horizontal and vertical in a Fourier space. In this section we describe the concepts of the Fourier space and the coordinates of the white points.

Let $I(i, j)$ be $N \times N$ binarized image in spatial domain whose frequency is sampled from Discrete Fourier Transform (DFT) and is defined as $F(k, l)$ in 2-dimension as shown in Eq. (1):

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j) e^{-2\pi t \left(\frac{ki}{N} + \frac{lj}{N}\right)}, \tag{1}$$

where the exponential term is the basis function that represents the sines and cosine waves with increasing frequencies corresponding to each $F(k, l)$ in the Fourier space. The Fourier image can be re-transformed to spatial domain $I(i, j)$ using Eq. (2):

$$I(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) e^{2\pi t \left(\frac{ka}{N} + \frac{lb}{N}\right)}, \tag{2}$$

where $\frac{1}{N^2}$ is the normalizing term in the image re-transformation from the Fourier domain to the spatial domain.

Since the Fourier transform is separable, it can be decomposed into two equations whereby the Eq. (3) represents first transformation of the spatial image to an intermediate image using N one-dimensional Fourier transformations and the Eq. (4) represents the transformation of the intermediate

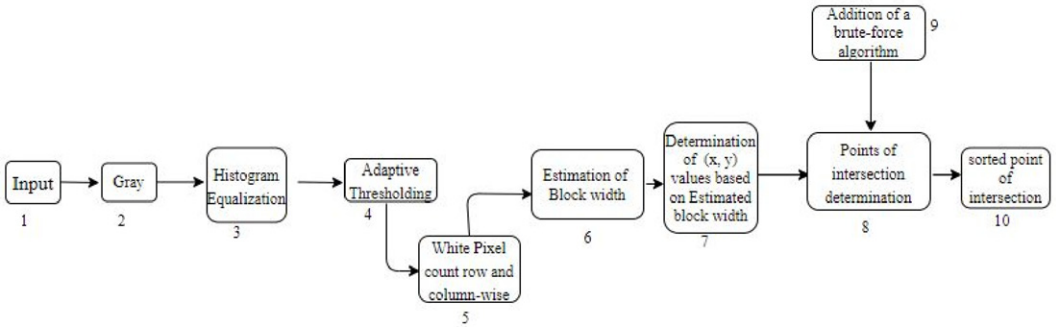

Figure 2. Procedures of the proposed algorithm.

image to the final image in a series of 2N one-dimension:

$$F(k,l) = \frac{1}{N}\sum_{b=0}^{N-1} J(k,b)e^{-2\pi t\left(\frac{lb}{N}\right)}, \tag{3}$$

where

$$J(k,b) = \frac{1}{N}\sum_{a=0}^{N-1} I(a,b)e^{-2\pi t\left(\frac{ka}{N}\right)}. \tag{4}$$

We reduce the complexity of the problem by applying fast Fourier transform to deduce the N^2 one-dimension to $N\log_2 N$. This procedure produces a complex valued output image that can be displayed with two images either real part and imaginary part with amplitude and angle phase. From Eq. (4), we use the real part of the processed image because the imaginary part is nearly zero and images are always real-valued functions. We filter the white pixels of the real part image along the horizontal $I_x(x,y)$ and vertical $I_y(x,y)$ to determine the points of intersections of the signal lines formed by white points:

$$I_x(x,y) = \begin{cases} 255, & J(k,b) > 0 \\ 0, & \text{elsewhere} \end{cases}, \tag{5}$$

$$I_y(x,y) = \begin{cases} 255, & J(k,b) > 0 \\ 0, & \text{elsewhere.} \end{cases} \tag{6}$$

The pixel coordinates are obtained by the intersection of the resulting white lines in Eqs. (5) and (6) as shown in Eq. (7):

$$(x_{id}, y_{id}) = (I_x(x,y) \cap I_y(x,y)), \tag{7}$$

where i represents the position of the points of intersection on an image, and d is the distance filter or sometimes referred to as an order in the Fourier domain. From the unsorted points (x_{id}, y_{id}) in the Fourier domain, we introduce a brute-force solution which determines the minimum error between the normalized euclidean distances between the normalized coordinates and its rounded off coordinate values. The procedures of the proposed algorithm is presented in the following Fig. 1. The blocks in each step are labeled sequentially such that every block is connected to the previous for producing optimal solution.

The first block with "Input" has 3D film-colored images which are converted to gray and stored in the second block. The third block with "Histogram Equalization" enhances the contrast of the gray-scale images in the previous block. The essence of applying histogram equalization is to reduce the manual labour in parameter settings when applying thresholding techniques in the fourth block. The fourth block with "Adaptive thresholding" binarizes all the input images with constant parameters (block size and a constant for weighted mean subtraction). In this block, a black and white image is produced to allow the procedures in the next block to take place. The fifth block with "white pixel count row-wise and column-wise" allows counting of all white pixels in each row and column which will be used to estimate the width of each block provided that the number of white points is known and defined in every image. The block with "Estimation of Block width" uses the prior information from every input image such that the height and width will guide the determination of the number of white pixels in each row and column. The block width will be determined by finding the ratio between the width of the image and the number of blocks in an image for horizontal (rows). A ratio between the height of the image and the number of blocks along the vertical (columns) will also be used to determine the block height. The block with "Determination of (x,y)-values based on the estimated block width" determines the x and y-values based on the estimated block width in the total count of the white pixels determined in the fifth block. The block with "Determination of points of intersections" obtains the points of intersections from the results in the seventh block performing it iteratively along the horizontal and vertical to determine the corner points of each block in the entire image hereinafter referred to as unsorted points of intersection. In the ninth block, we introduce a brute-force algorithm that determines the minimum error between the normalized points and the rounded off coordinate values from the eighth block. In this block, we define brute force as a random search algorithm that exhaustively searches by generating and testing technique [23]. The significance of introducing the brute-force is to determine the minimum error by listing all possible candidates that will be used to determine the

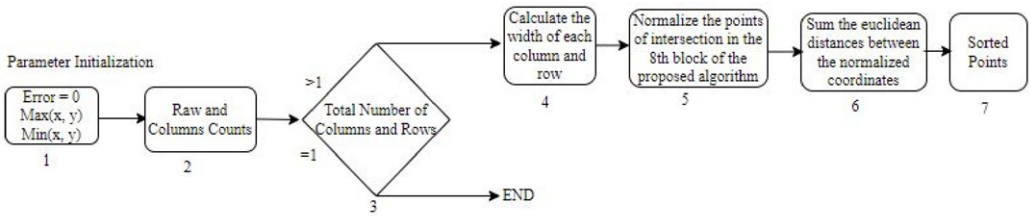

Figure 3. The expansion of the brute-force in the 8th block of the proposed algorithm.

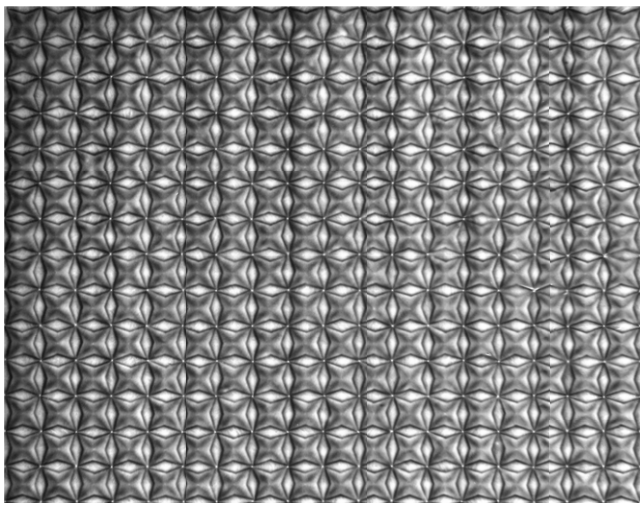

Figure 4. Grayscale image for experimentation.

Table I. Mathematical notations.

Symbols	Definitions
R_c	Row count obtained direct from the image
C_c	Column count obtained direct from the image
C_w	Column width
R_w	Row height
$\hat{x}_n$	Normalized x-value coordinates
$\hat{y}_n$	Normalized y-value coordinates
i and j	Any arbitrary constants obtained in a binary image

minimum error. The extension of the ninth block in Figure 2 is shown here below.

The algorithm for the brute-force has 7 blocks staring with "parameter initialization" to "sorted points". The first block starts the program by initializing the minimum error to zero, minimum x- and y-coordinate values, and maximum x- and y-coordinate values based on the unsorted coordinates in block eighth of the proposed algorithm. The second block with "Row and Column counts" determines the number of rows and columns that the image has based on image dimensions. The third block with "Number of Columns for an Image" defines an if-else condition that if the number of columns in the entire image is equal to 1 the program ends with just four coordinates else if the number of columns is greater than 1, the program calculates the width of each column and row storing the results in the fourth block. The fifth block of the brute-force performs normalization of the points of intersections in the eighth block in the proposed algorithm by using the column width and row heights to determine the normalized coordinates. The sixth block sums the Euclidean distances between the normalized coordinates and the rounded-off coordinates to determine the minimum incremental error. This error is subtracted from the unsorted coordinates in the eighth block of the proposed algorithm as shown in Figure 3. The last block with "sorted points of intersections" consists of coordinates from the eighth block affected with the error calculated by the brute-force in the ninth block yielding the accurate coordinates. The pseudocodes for the extended algorithm is provided below and the mathematical notations used are listed in Table I.

The main idea is to generate all possible dimensions of the grid in every block given that each block has a different size along the horizontal and along the vertical. Lastly, we define the correctness of the proposed algorithm (DR) by the ratio of the total correctly detected and localized points to the total white points in the 3D film as shown in Eq. (8):

$$DR = \frac{\text{Detected points} - \text{mislocated points}}{\text{Actual white points}}. \tag{8}$$

4. EXPERIMENTAL RESULTS AND DISCUSSION

The experiments of this work were performed on a gaming computer operating on Windows 10 installed with Python 3.8. The libraries used are Numpy, OpenCV, and Scipy for pixel manipulation, image processing, and white points' determination respectively. We used four different images to justify the performance of our algorithm and compare it with the other two rule-based algorithms.

We start the experiments by following the procedures of the proposed algorithm; we convert the colored image to gray to simplify the determination of the white points on an image.

In Figure 4, the image is presented as a single channel of a multichannel image by representing the amount of light intensity as black and white. They are the monochrome images that are exclusive of shades whereby, the weakest

Pseudo codes for the brute-force algorithm
D is the set of points such that D[j][0] represent the x-coordinate and D[j][1] represent the y-coordinate
Let d = |D| or the number of points in D.
Let max_value_x = the maximum value of x-coordinate in D >1 symbolized as max_x.
Let max_value_y = the maximum value of y-coordinate in D >0 1 symbolized as may_x..
Minimum values for x, y in D are assumed to be zero.
Initialize minimum error distance = +infinity.
Initialize min error in columns = 0.
Definition of symbols

// compute for integer number of rows and columns only.

$for\ i\ in\ range(i = 1,\ i \le d)$:
{
$if\ \big((d\%C_c) == 0\big)$:
$R_c = \dfrac{n}{C_c}$
}

{
$if\ \big((d\%R_c) == 0\big)$:
$C_c = \dfrac{n}{R_c}$
}

// Compute the width of the columns and height of the rows. If the number of columns is 1, let the column width be maximum value of x (and for all rows).

{
$if\ (C_c > 1)$:
$C_w = \dfrac{max_x}{C_c - 1}$
$else: C_w = max_x$
}
{
$if\ (R_c > 1)$:
$R_h = \dfrac{max_y}{R_c - 1}$
$else: R_h = max_y$
}

// If the condition in the previous is satisfied then reset the error for the new configuration.

$error = 0.0$

// For the current point, normalize x- and y-coordinates so that it is in the range of (0, number of columns – 1) and (0, number of rows – 1).

$for\ j\ in\ range(j = 1, j \le n)$:
{
$\hat{x}_{nj} = \dfrac{D[j][0]}{C_w}$
$\hat{y}_{nj} = \dfrac{D[j][1]}{R_h}$
}

// Error is the sum of the squares of the distances between the normalized values and the rounded off values the x and y direction.

$$error = \sqrt{\left(\hat{x}_{nj} - round\left(\hat{x}_{nj}\right)\right)^2 + \left(\hat{y}_{nj} - round\left(\hat{y}_{nj}\right)\right)^2}$$

// Return new points as a difference between the points obtained in equation (7) and the error as shown below.

$$(x_{is}, y_{is}) = (x_{id}, y_{id}) - error$$

// End algorithm.

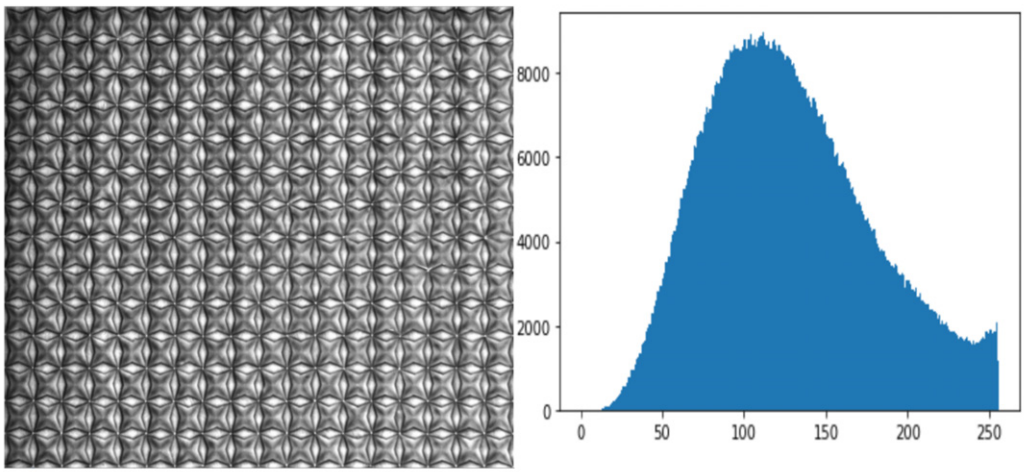

Figure 5. The gray image with its pixel distribution.

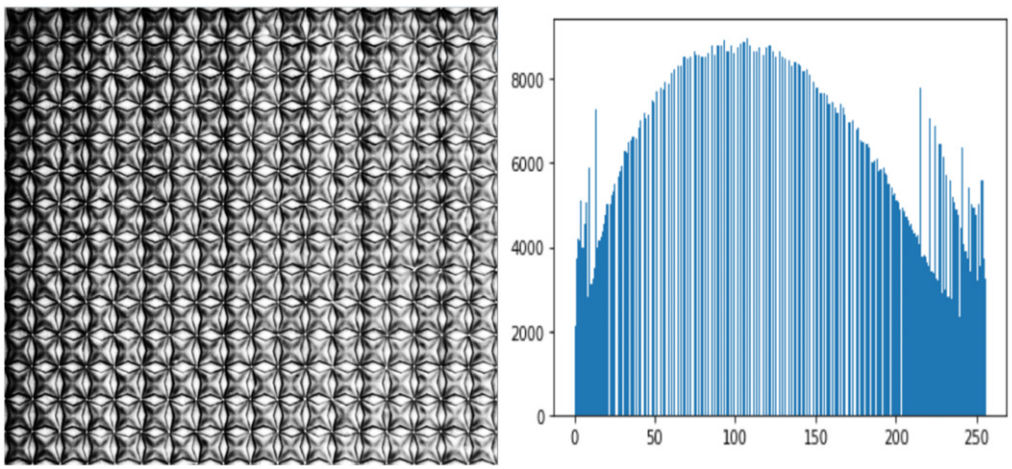

Figure 6. Histogram equalized image with its pixel distribution.

part becomes black and the strongest part becomes white. We present the first three procedures as shown in the proposed algorithm by presenting the distribution of the gray, histogram equalization, and its distribution as shown in Figures 5–7.

The gray-scale image's distribution is determined and then subjected to histogram equalization to standardize the pixels' intensity as shown in Figs. 5–7. From the top-right position in Fig. 7, we count the total white pixels based on a binarized image and apply a distance filter of order d ranging from 45 to 55 to determine the points of intersections without brute-force algorithm. The following figure presents all the results from different images with their respective algorithms.

From Figure 8, the left side with images (a) and (c) are subjected to all procedures of the proposed algorithm except histogram equalization whose results are presented in the right position with the image (b) and (d). The image labelled with (b) indicates that the algorithm was able to detect and localize all-white points except for the last column of the image. Additionally, the proposed algorithm without image processing was not able to detect and localize the white points in the third row of the image labelled (d). This implies that the images have varying intensities, which is why there are some regions where the points are available but they are not easily detected and localized. We present the comparison results of one image against template matching, the iterative algorithm, proposed without brute force.

From Figure 9, the image labelled (a) represents the input image and the rest from (b) to (d) represents the results for template matching iterative method, and the proposed algorithm without brute-force. The image in (b) shows that the template matching method was not able to detect all of the white points because of a change in position and orientation. The method in (b) was not able to detect all of the white points in each row and column and was not able to localize

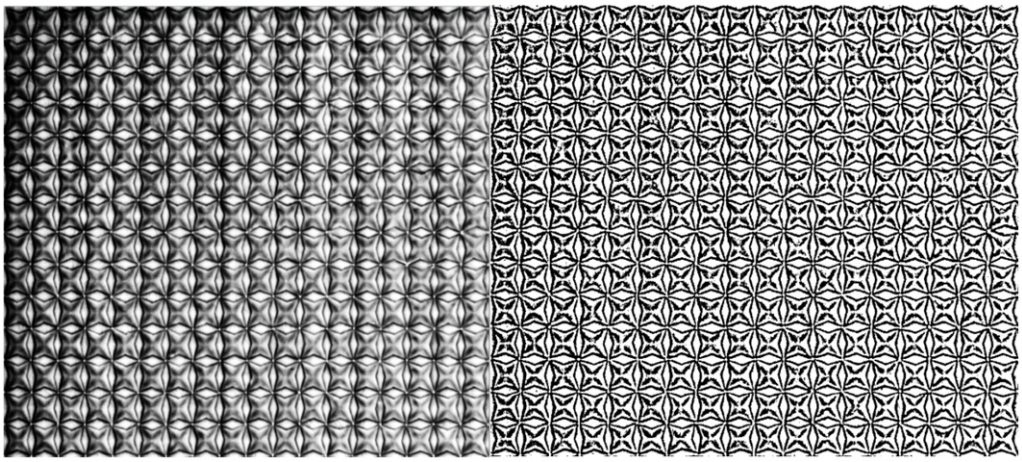

Figure 7. Binary image of a gray image after the application of histogram equalization.

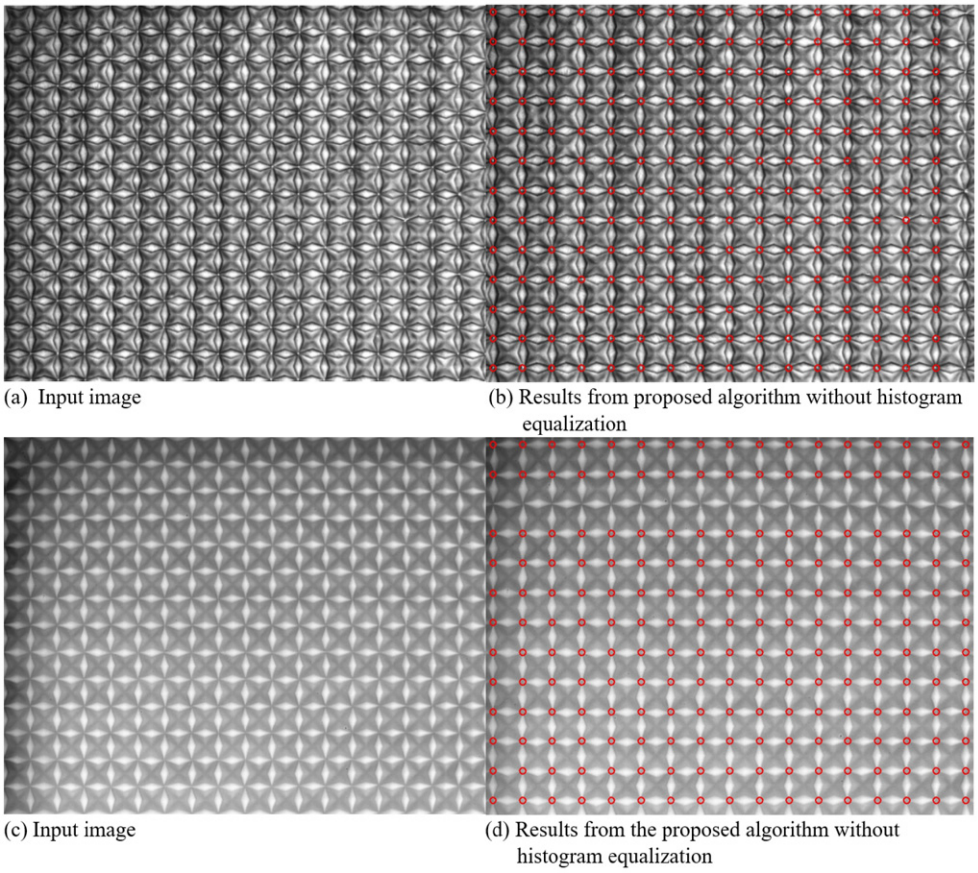

Figure 8. (a) and (c) represents input images, (b) and (d) presents the proposed algorithm without histogram equalization.

the correct position of the detected points. In this method, a total number of 47 white points were detected in the entire image but 10 white points only were correctly detected and localized even though the total number of white points in the entire image is 252. The method in (c) is easily affected by images whose patterns have varying column width sizes because during experimentation it was fixed to a size of 169 pixels. Method (c) and (d) were able to detect an equal number of white points in the entire image but method (d) performed better than method (c) however, it mislocated a

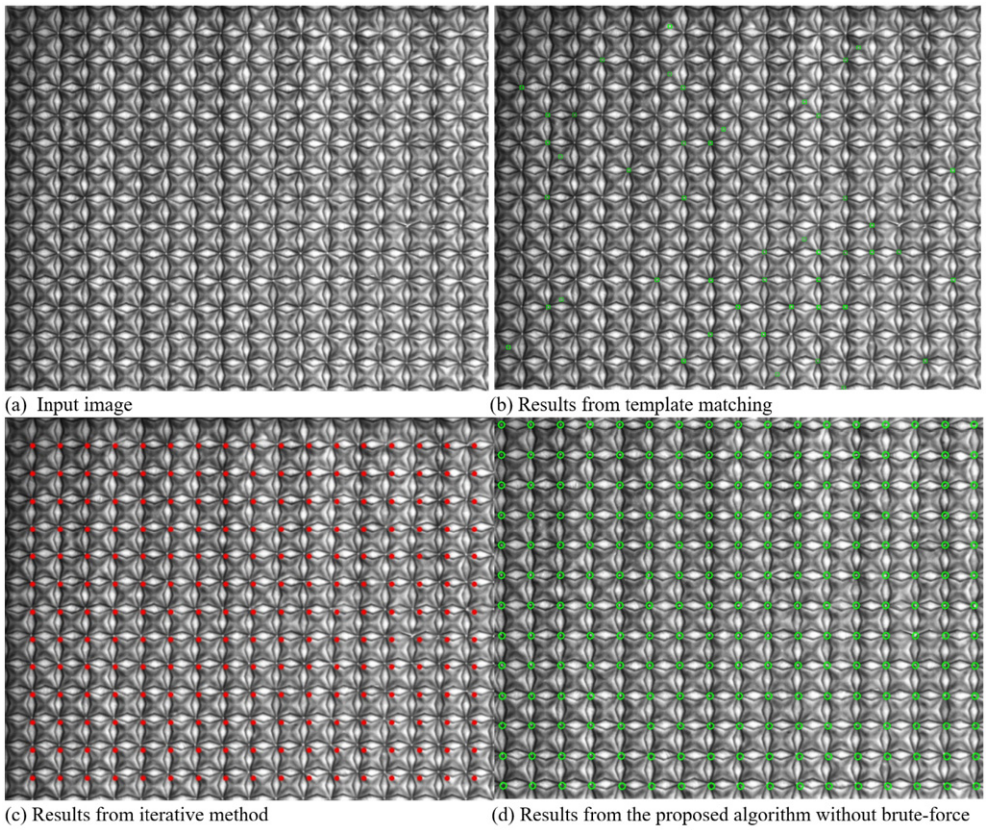

Figure 9. (a) input image (b) points detected and localized by template matching, (c) points detected and localized by iterative method.

few white points in the last three columns of the image. It should be noted that the centroid of the green circle should be seen as a white point within the green circle and not otherwise.

Our proposed algorithm performed better than the other methods because it initially counters the nonuniform contrast problem. Additionally, it applies a thresholding technique to filter the existing noise in the image signal. Then the binarized image is transformed by a fast Fourier transform to determine unsorted coordinates of the white points on the 3D film images. Lastly, the unsorted coordinates are filtered by the minimum error of the normalized coordinates and the rounded-off coordinates whose error value is subtracted from the coordinates obtained by the fast Fourier algorithm. The following figure compares the performance of three different images for the proposed algorithms. The other algorithms such as template matching, iterative algorithm, proposed without histogram equalization are not included in this figure.

From Figure 10, the images labelled as (a) up to (f) represents the results from the proposed algorithm with and without histogram equalization. The results are characterized by green and yellow colored circles for the two compared methodologies. For images (a) and (b), the green circles are affected by the distance factor (order = 40) such that the last two rows bear circles with missing the centroid (white points). This indicates that the algorithm was able to detect the existence of a white point but failed to locate the exact position of a point on the image. The introduction of brute force to the proposed algorithm nullified the negative impact of the order in the fast Fourier transformed image and attain outstanding results. Of all the above-extracted figures, the yellow circles have centroids in themselves while few of the green circles lack centroids.

We evaluated our proposed algorithm over different images based on the number of white points detected on the 3D films, number of mislocated points, filter, and threshold. We used a constant threshold of 0.8 for the template matching algorithm to avoid feature overlapping. This situation is influenced by varying intensities in every image. We also used a column width of 169 for the iterative method and a range between 40 and 50 for the rest of the algorithms. The summary of the comparative results is shown in Table II whose pictorial results are annexed in the appendix section (Figures A.1–A.4). This table presents template matching, iterative, proposed without brute-force, and proposed with brute-force algorithms.

From Table I, the template matching algorithm detected a few white points in the four images compared to the rest of the algorithms involved in the experiments. Even though the algorithm detected a few white points but it has mislocated a few points too. The iterative method detected many white points but has mislocated many of them on an image depicting the worst performance of all the algorithms.

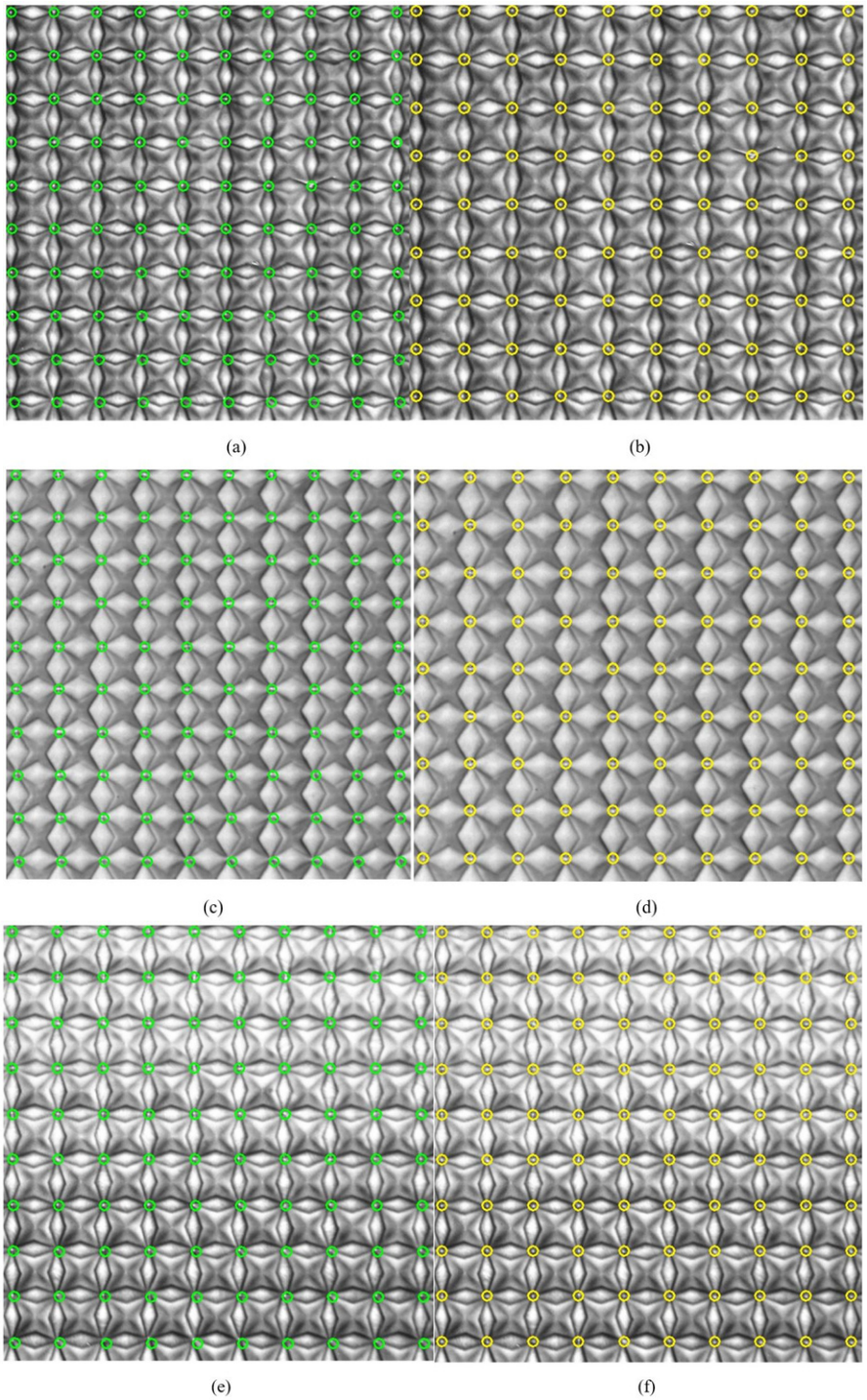

Figure 10. Comparison between the proposed algorithm with and without a brute-force.

The proposed algorithm with and without brute-force has comparable results such that the significance of adding a brute-force reduced the mislocation of white points.

Based on the experimental results, we could establish that the proposed algorithm has shown better results compared to other methods regardless of the effect of contrast and column width between the white points on the image. Most of the methods were able to detect but failed to properly locate the detected white points on the digital images. The proposed algorithm can detect all white points and correctly locate them on the image. Therefore, the proposed algorithm can be recommended to solve problems similar to this one.

Table II. Evaluation of the methods for white points detection based on detection ratio.

Method	Image	Threshold/ Filter	Actual points	Detected points	Mislocated points	Detection ratio
Template matching	1	0.80	252	47	10	0.15
	2	0.80	270	108	17	0.34
	3	0.80	270	189	112	0.29
	4	0.80	270	109	9	0.37
Iterative method	1	169	252	221	201	0.01
	2	169	270	221	204	0.01
	3	169	270	221	189	0.12
	4	169	270	221	204	0.01
Proposed method 1	1	40–55	252	252	37	0.80
	2	40–55	270	270	11	0.96
	3	40–55	270	270	24	0.91
	4	40–55	270	270	22	0.90
Proposed method 2	1	40–55	252	252	3	0.99
	2	40–55	270	270	4	0.98
	3	40–55	270	270	3	0.99
	4	40–55	270	270	1	0.99

5. CONCLUSIONS

We proposed an algorithm that detects and localizes white points on 3D film images using the fast Fourier transform with and without a brute force algorithm. The proposed algorithm uses the fast Fourier transform to transform a binarized image for obtaining the white points which are unsorted because of the application of the distance filter ranging from 40 to 55. The coordinates of the white points were obtained by determining the points of intersections of the coordinates of the white points along the horizontal and vertical. The unsorted points of intersection are then refined by using a brute force algorithm that minimizes the error between the normalized coordinates and the rounded-off coordinates as determined through the algorithm. The proposed algorithm with a brute force solution outperformed the one without brute force by 10%. We performed various experiments using the existing algorithms such as template matching, and an iterative method to validate the performance of our algorithm. The experimental results indicated that the proposed algorithm performs better than the template matching, and the iterative algorithm. Therefore, we recommend the proposed algorithm for solving problems similar to this. Based on the coordinates of the detected and localized white points, we will extract the star-like feature rectangle to evaluate their properties and categorize them into various classes by using either machine learning or neural networks.

APPENDIX.

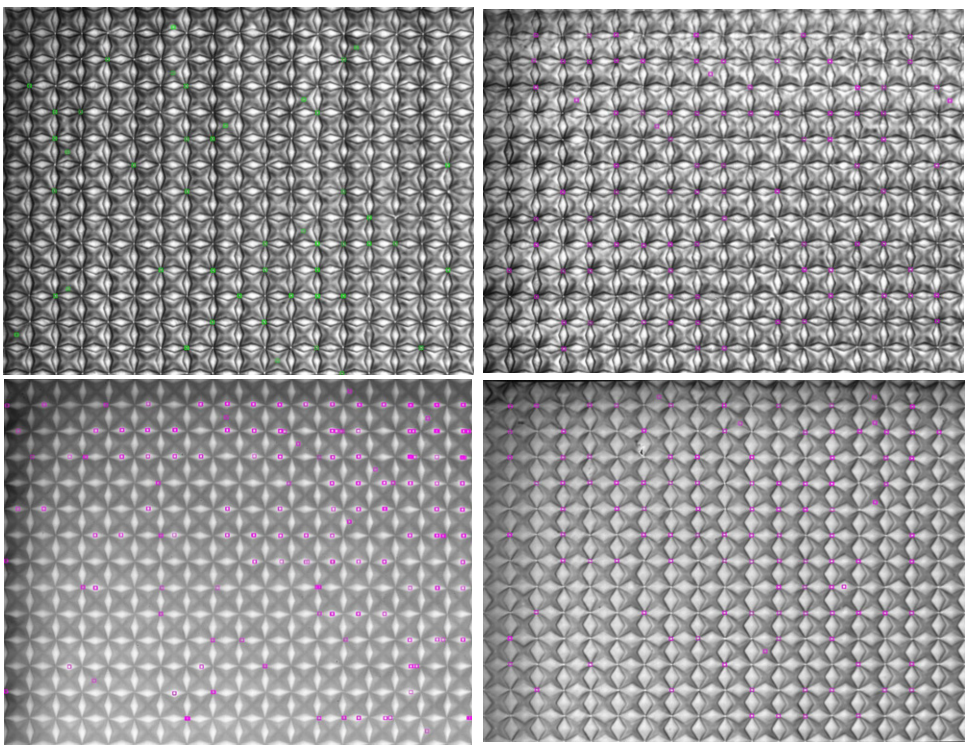

Figure A.1. Template matching results from image 1–4.

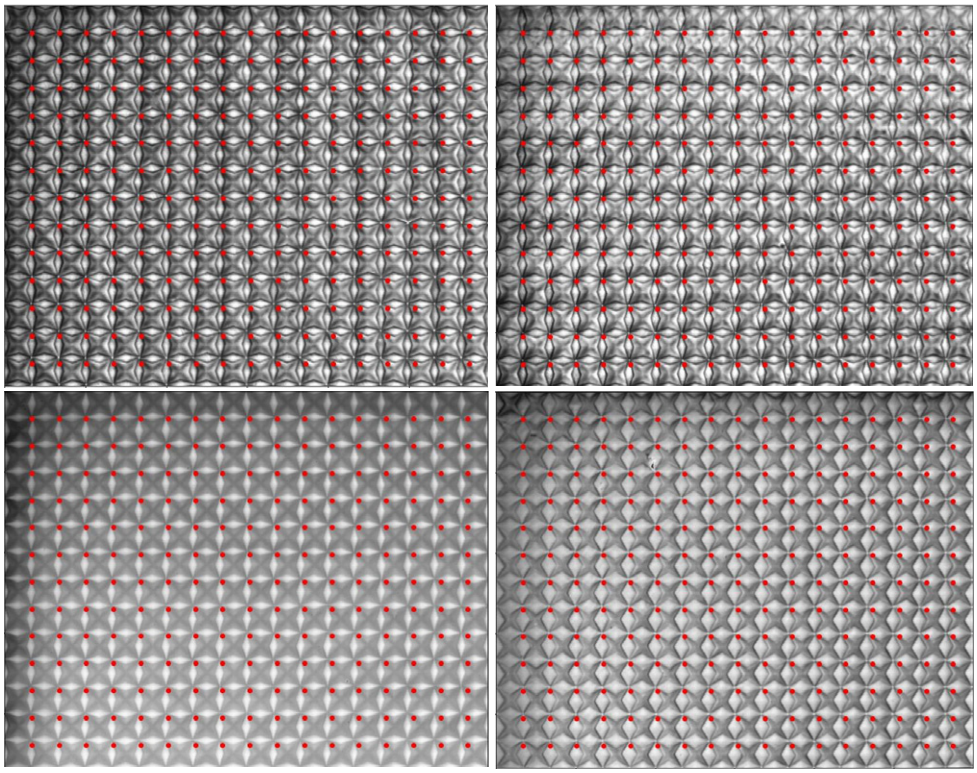

Figure A.2. Iterative method results for images from 1–4.

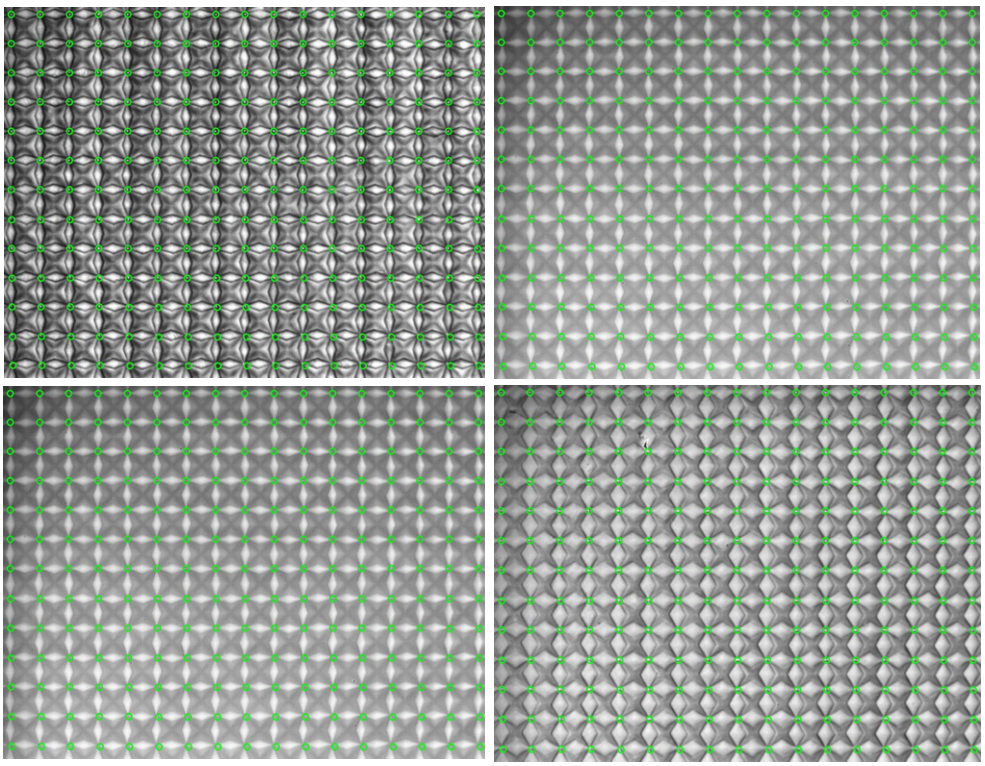

Figure A.3. Results for the proposed algorithm without brute force for image 1–4.

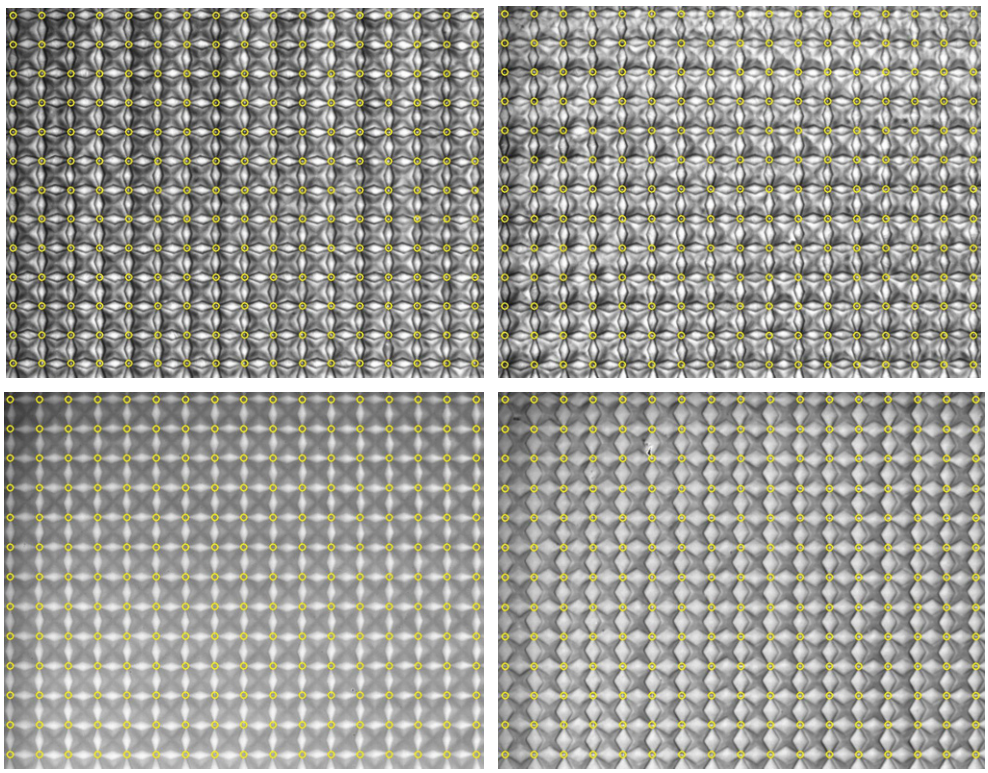

Figure A.4. Results for the proposed algorithm with brute force for image 1–4.

ACKNOWLEDGMENT

This work was supported by Basic research program of NRF, Fund growing technology program of SMTECH, and Regional SW service program of NIPA.

REFERENCES

[1] A. G. Solimini, "Are there side effects to watching 3D movies? A prospective crossover observational study on visually induced motion sickness," PLoS One **8**, 1–8 (2013).

[2] K. Huda, M. Mahmoud, and M. Ra'ed, "A review of stereoscopic 3D: home entertainment for the twenty first century," 3D Review **5**, 1–9 (2014).

[3] R. Dhanabal, S. K. Sahoo, V. Bharathi, K. Dowluri, B. S. R. P. Varma, and V. Sasiraju, "FPGA based image processing unit usage in coin detection and counting," *Proc. Int'l. Conf. on Circuits, Power and Computing Technologies* (IEEE, Piscataway, NJ, 2015), pp. 1–5.

[4] S. M. M. Roomi and R. B. J. Rajee, "Coin detection and recognition using neural networks," *Proc. Int'l. Conf. on Circuits, Power and Computing Technologies* (IEEE, Piscataway, NJ, 2015), pp. 1–6.

[5] D. Mehta and A. Sagar, "A Survey on various techniques of coin detection and recognition," Int. J. Comput. Appl. **6**, 29–32 (2013).

[6] T. Sooruth and M. V. Gwetu, "Automatic south african coin recognition through visual template matching," *Proc. Int'l. Conf. on Advances in Big Data, Computing and Data Communication Systems* (IEEE, Piscataway, NJ, 2018), pp. 1–6.

[7] A. V. Durg and R. Oreg, "Global white point detection and white balance for color images," *Intel Corporation* (Taylor & Zafman LLP, Georgia, USA, 2000), pp. 1–2.

[8] J. Jo, J. Im, J. Jang, Y. Yoo, and J. Paik, "Adaptive white point extraction based on dark channel prior for automatic white balance," IEIE Trans. Smart Process. Comput. **5**, 383–389 (2016).

[9] C.-A. Lin, "Method and computer system of white point detection," Samsung Electron., U.S. Patent 10,803,341. 1–3 (2019).

[10] N. S. Hashemi, R. B. Aghdam, A. S. B. Ghiasi, and P. Fatemi, "Template matching advances and applications in image analysis," Am. Sci. Res. J. Eng., Technol. Sci. **26**, 91–108 (2016).

[11] Q. I. Pham, R. Jalovecky, and M. Polášek, "Using template matching for object recognition in infrared video sequences," *Proc. 34th Digital Avionics Systems Conf.* (IEEE, Piscataway, NJ, 2015), pp. 1–15.

[12] J. L. Latecki, "Template matching," Lecture Notes (Tempele University, 2017).

[13] G. Koutaki and K. Uchimura, "Template matching robust to intensity variations and noise using eigen-decomposed template," Electron. Commun. Japan **95**, 27–36 (2012).

[14] T. Kobayashi, A. Hidaka, and T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection," in *Neural Information Processing*, edited by M. Ishikawa, K. Doya, H. Miyamoto, and T. Yamakawa, Lecture Notes in Computer Science (Springer, Berlin, Heidelberg, 2007), 4985, pp. 598–607.

[15] R. Bourezak and G. Bilodeau, "Object detection and tracking using iterative division and correlograms," *Proc. 3rd Canadian Conf. on Computer and Robot Vision* (IEEE, Piscataway, NJ, 2006), pp. 38–46.

[16] K. Cheng, Y. Chen, and W. Fang, "Improved object detection with iterative localization refinement in convolutional neural networks," IEEE Trans. Circuits Syst. for Video Technol. **28**, 2261–2275 (2018).

[17] K. Zhao, Q. Han, C. B. Zhang, J. Xu, and M. M. Xheng, "Deep Hough transform for semantic line detection," IEEE Trans. Pattern Anal. Mach. Intell. 1–14 (2021).

[18] P. Bachiller-Burgos, L. Manso, and P. Bustos, "A variant of the Hough transform for the combined detection of corners, segments, and polylines," J. Image Video Process. **32**, 1–26 (2017).

[19] K. Kinsley and D. Kukiela, *Neural Networks from Scratch in Python* (Harrison Kinsley, Enterprises Inc, Houston, USA, 2020), pp. 1–7.

[20] L. Marvin, *Neural Networks with MATLAB* (CreateSpace Independent Publishing Platform, California, USA, 2016), pp. 1–14.

[21] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L Marques, "Recent advances and applications of machine learning in solid-state materials science," npj Comput. Mater. **5**, 1–36 (2019).

[22] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," J. Clin. Epidemiol. **49**, 1225–1231 (1996).

[23] E. Fox and C. Guestrin, *Machine Learning: Clustering and Retrieval,* Causera Course, Retrieved June (University of Washington, Washington, USA, 2018).