

## Modified iterative method with red-black ordering for image composition using poisson equation

Zakariah Aris<sup>1</sup>, Nordin Saad<sup>2</sup>, A'qilah Ahmad Dahalan<sup>3</sup>, Andang Sunarto<sup>4</sup>, Azali Saudi<sup>5</sup>

<sup>1,2,5</sup>Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia

<sup>3</sup>Centre for Defence Foundation Studies, National Defence University of Malaysia, Kuala Lumpur, Malaysia

<sup>4</sup>Institut Agama Islam Negeri, Bengkulu, Indonesia

### Article Info

#### Article history:

Received Jun 27, 2020

Revised Apr 30, 2021

Accepted May 27, 2021

#### Keywords:

Finite difference

Image composition

MAOR

Modified iterative method

MSOR

MTOR

Poisson equation

Red-black ordering

### ABSTRACT

Image composition involves the process of embedding a selected region of the source image to the target image to produce a new desirable image seamlessly. This paper presents an image composition procedure based on numerical differentiation using the laplacian operator to obtain the solution of the poisson equation. The proposed method employs the red-black strategy to speed up the computation by using two acceleration parameters. The method is known as modified two-parameter over-relaxation (MTOR) and is an extension of the existing relaxation methods. The MTOR was extensively studied in solving various linear equations, but its usefulness in image processing was never explored. Several examples were tested to examine the effectiveness of the proposed method in solving the poisson equation for image composition. The results showed that the proposed MTOR performed faster than the existing methods.

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



### Corresponding Author:

Azali Saudi

Faculty of Computing and Informatics

Universiti Malaysia Sabah, Kota Kinabalu, Malaysia

Email: azali@ums.edu.my

## 1. INTRODUCTION

The image composition process computes the image gradients of both source and target images to generate a new desirable image of two composed images seamlessly. Other image processing that employs a similar approach includes image matting [1], image stitching [2], surface reconstruction [3], sharpening filter [4], image completion [5], and image inpainting [6]. In this paper, we present our work in using numerical methods to solve the image composition problem. The Poisson equation is used to model the image composition process, in which the image gradients are computed with the Laplacian operator. The problem is transformed into a large and often sparse linear system which is suitable to be solved using numerical differentiation. In the proposed numerical method, additional acceleration parameters are considered to provide more flexibility during parameter tuning and to further improve the execution time. The proposed method also employed red-black ordering into the modified variants of the relaxation method, in which pixels involved during the composition process are labeled in red or black color nodes. Since red and black nodes are computed independently, the computation to obtain the image gradient can be executed in parallel. To ensure that all tested methods would generate comparably the same final images, several image similarity tests are conducted.

The use of the Poisson equation in image processing for computing image gradients was introduced by Patrick Pérez, Michel Gangnet, and Andrew Blake [7]. Since then several works had been conducted in applying image gradients for image processing. R. Raskar, Adrian Ilie, and Jingyi Yu [8] employed an

approach based on the gradient domain method which preserves local perceptual properties and minimizes usual aliasing, haloing, and ghosting problems. In 2004, Sengupta *et al.* [1] proposed an approach to the natural image matting problem by solving the Poisson equation through the matte gradient domain. It was shown in their work that the proposed approach produced encouraging results for various natural images which previously were problematic. After that Wang and Yang [2] proposed an image stitching method to combines several images that have some overlapped images. A set of the cost function is defined where the similarity of the images and visibility of the seam were constructed in the gradient domain. By working in the gradient domain rather than the pixel intensities of the images, the authors showed that the proposed approach was capable of reducing global inconsistencies and minimizing problems generated by the illumination effect. In the work by Gao *et al.* [5], a completion algorithm based on the gradient domain that applied a patch-based technique was reported. This work involved two phases: firstly, they used the gradient maps to complete the removed area by applying a patch-based filling algorithm. Secondly, the image was then reconstructed from the gradient maps by solving the Poisson equation. To achieve better image completion, a matching criterion based on the gradient and color was proposed.

Image editing based on the gradient domain was also used in the study conducted by Kazhdan *et al.* [3] for surface reconstruction by applying an algebraic approach. In their work, a comprehensive study of different integration methods and the problem when working with noisy gradient fields are presented. A similar study was proposed by Zhouyu Du, Antonio Robles-Kelly, and Fangfang Lu [9], Dikpal Reddy, Amit Agrawal, and Rama Chellappa [10]. Vijayakumar [4] studied the applications of the poisson equation to devise several image filters such as sharpening and flattening. They also proposed the solutions in the fourier domain which showed a more direct, exact, and efficient approach. This idea was also extended and applied to video sequences by employing temporal values as constraints. The image composition approach to reduce bleeding artifacts in the generated images was proposed by Michael W. Tao, Micah K. Johnson, and Sylvain Paris [11]. The gradient-based method is used for video editing using the variational model by Rida Sadek, Gabriele Facciolo, Pablo Arias, *et al.*, [12]. J.-M. Morel, A. B. Petro, and C. Sbert [13] proposed a fourier implementation to solve the poisson equation for image processing. Using neumann boundary conditions, the poisson equation is solved in the retinex algorithm through fourier implementation by Nicolas Limare, Ana Bel'en Petro, Catalina Sbert, and Jean-Michel Morel [14]. Zeng *et al.* [6] proposed a variational framework based on a unified four non-local schemes for their work in image inpainting. They applied the gradients between patches to generate the inpainted image. The image pyramid method for solving the poisson equation was suggested by Hussain and Kamel [15]. Afifi and Hussain [16] proposed the modified Poisson to solve the image blending problem. The iterative method to solve the Poisson equation for image editing was suggested in [17]. In [18] and [19], the modified over-relaxation and block method were developed. In the previous study, the TOR method had been applied to solve path planning problem [20] and many other partial differential equations [21]-[24].

More recently, Cao *et al.* [25] proposed a self-embedding image watermarking scheme based on reference sharing and the poisson equation. In their work, they established the relationship of each pixel and its neighborhood in the original image via Laplacian operator and be converted to compression bits. The approach increased the recovered area from the tampered image by reconstructing the relationship between each compression bit and each reference bit. In [26], the poisson blending algorithm was applied in the cloud removing algorithm for a large area of landsat data. The gradient-domain compositing method is accelerated by utilizing a quad-tree approach. The proposed method demonstrated promising results for experimental images with cloud coverage of less than 80%. Dongbo Min, Sunghwan Choi, Jiangbo Lu, *et al.*, [27] presented an edge-preserving image smoothing method known as fast global smoother. In their method, the sparse laplacian matrices are efficiently solved through global objective functions. By combining the efficient edge-preserving filters and optimization-based smoothing, the approach obtained comparable runtime to the fast edge-preserving filters and also overcame many limitations of the existing local filtering approaches. Yang Pan, Mingwu Jin, Shunrong Zhang, *et al.*, [28] proposed an approach that combines deep convolutional generative adversarial network (DCGAN) and Poisson blending (PB) to perform image completion tasks. Most recently, Nordin Saad, Andang Sunarto, and Azali Saudi [29] employed a modified method by combining red-black strategy with accelerated over-relaxation, known as MAOR method, demonstrated encouraging results. Works on image editing that involves blending, composition, or completion using the Poisson equation are summarized in Table 1.

Table 1. Works on poisson image editing

Authors	Application	Method	Authors	Application	Method
[7]	Poisson image editing	Successive over-relaxation (SOR)	[19]	Image blending	SOR and group iteration
[13]	Poisson image editing	Fourier transform	[26]	Cloud removal using image blending	Gauss-Seidel
[15]	Poisson image editing	SOR	[28]	Map Completion	DCGAN and SOR
[18]	Image blending	Modified SOR	[29]	Image composition	MAOR

*Modified iterative method with red-black ordering for image composition using poisson ... (Zakariah Aris)*

## 2. IMAGE COMPOSITION

An image is constructed based on a two-dimensional array of pixels as illustrated in Figure 1. In the image composition process, the procedure begins by selecting the region of interest  $R$  from source image  $S$ ,  $dR$  denotes the boundary of  $R$ . The selected region is then copied to the desired position of a target image  $T^*$  to produce a new image  $T$ . Then, the vector field  $V$  of the source image  $S$  is computed. Using the Poisson equation as a constraint to model the pixels inside the region  $dR$ , new pixel values are computed by minimizing the gradient difference of  $V$  and the selected region of the target image  $T^*$ .

The formulation of this minimization problem can be written as:

$$\min_f \iint_O |\nabla f - v|^2 \text{ with } f|_{\partial O}. \quad (1)$$

The solution of the minimization problem (1) is the unique solution of the Poisson equation with Dirichlet boundary condition.

$$\Delta f = \text{div } v \text{ at } 0 \text{ with } f|_{\partial O} = f^*|_{\partial O} \quad (2)$$

In the composition process, the vector field  $v$  is a gradient field directly extracted from the source image. Therefore, the (2) is rewritten as.

$$\Delta f = \nabla g \text{ at } 0 \text{ with } f|_{\partial O} = f^*|_{\partial O} \quad (3)$$

where the  $\Delta$  denotes the Laplacian operator.

### 2.1. Formulation of the proposed method

To begin the image composition process, the gradient fields in the source image need to be computed by solving the Poisson equation that is used to model the problem. Let us consider the standard Poisson equation in 2D written as (4).

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x, y) \quad (4)$$

The discretization of (4) using the five-point Laplacian operator will result in the following approximation (5).

$$U_{i,j} = \frac{1}{4} [U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - h^2 f_{i,j}] \quad (5)$$

where  $h = \Delta x = \Delta y$ . The iterative scheme of (5) is given as (6) [30].

$$U_{i,j}^{(k+1)} = \frac{1}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k+1)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k+1)} - h^2 f_{i,j}] + (1 - \omega) U_{i,j}^{(k)} \quad (6)$$

where  $\omega$  represents a weighted parameter. In [31], the acceleration over-relaxation (AOR) scheme was developed and described as (7).

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + \frac{\tau}{4} [U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}] + (1 - \omega) U_{i,j}^{(k)}. \quad (7)$$

where additional parameter,  $\tau$ , is added. An extension to the SOR and AOR is the two-parameter over-relaxation (TOR) iterative method [32] which employs two acceleration parameters to provide more flexibility and weight tuning choices during the iteration process. The finite-difference approximation equation for the TOR method is given as (8).

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + \frac{\tau}{4} [U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)}] + \frac{\gamma}{4} [U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}] + (1 - \omega) U_{i,j}^{(k)}, \quad (8)$$

The optimal value for the three parameters ( $\omega$ ,  $\tau$ , and  $\gamma$ ) is in the range between 1 and 2.

**2.2. Red-black strategy**

Figure 2 illustrates the implementation of a red-black ordering strategy, in which the red and black nodes are computed alternatively, thus very suitable for parallel implementation [33]. By applying the red-black strategy, the computations of red and black nodes involve two phases. The procedure begins by computing red nodes where (i+j) is even. In the second phase, black nodes, where (i+j) is odd, are computed. The iteration procedures of these two phases continue until the stopping criterion is satisfied.

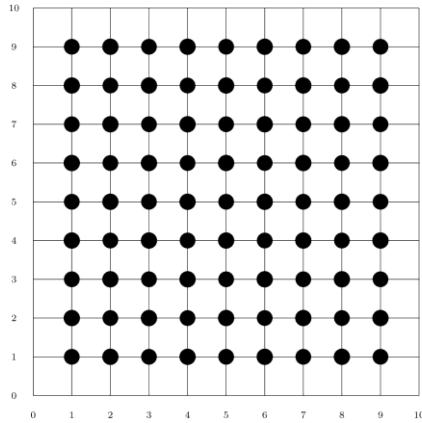


Figure 1. The standard finite grid of size, N=10

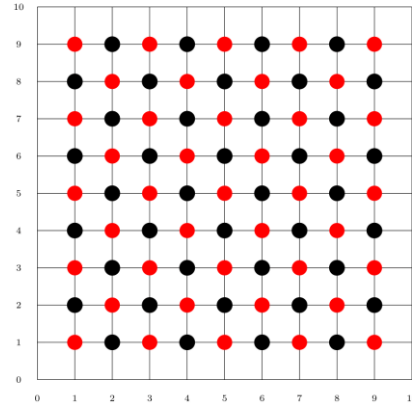


Figure 2. The finite grid with red-black ordering

Based on the iteration scheme given in (6), the red-black SOR (RBSOR) formula are given as.

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + (1 - \omega)U_{i,j}^{(k)} \text{ for all } i+j \text{ even} \quad (9)$$

and

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k+1)} + U_{i-1,j}^{(k+1)} + U_{i,j+1}^{(k+1)} + U_{i,j-1}^{(k+1)} - h^2 f_{i,j}] + (1 - \omega)U_{i,j}^{(k)} \text{ for all } i+j \text{ odd} \quad (10)$$

The red-black AOR (RBAOR) formula are based on the iteration scheme given in (7) and can be expressed as:

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + (1 - \omega)U_{i,j}^{(k)} \text{ for all } i+j \text{ even} \quad (11)$$

and

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + \frac{\tau}{4} [U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i+1,j}^{(k+1)} - U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)} + U_{i,j+1}^{(k+1)} - U_{i,j+1}^{(k)}] + (1 - \omega)U_{i,j}^{(k)} \text{ for all } i + j \text{ odd} \quad (12)$$

Lastly, the red-black TOR (RBTOR) formula that is based on (8) are given as.

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + (1 - \omega)U_{i,j}^{(k)} \text{ for all } i+j \text{ even} \quad (13)$$

and

$$U_{i,j}^{(k+1)} = \frac{\omega}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + \frac{\tau}{4} [U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i+1,j}^{(k+1)} - U_{i+1,j}^{(k)}] + \frac{\gamma}{4} [U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)} + U_{i,j+1}^{(k+1)} - U_{i,j+1}^{(k)}] + (1 - \omega)U_{i,j}^{(k)} \text{ for all } i+j \text{ odd} \quad (14)$$

The computational molecules for red and black nodes are illustrated in Figure 3. Red nodes are calculated using the (9), (11), and (13). While, the (10), (12), and (14) are used to calculate black nodes. The implementation of these equations is executed iteratively, in which the execution continues until the tolerance error rate  $\epsilon$ , is satisfied.

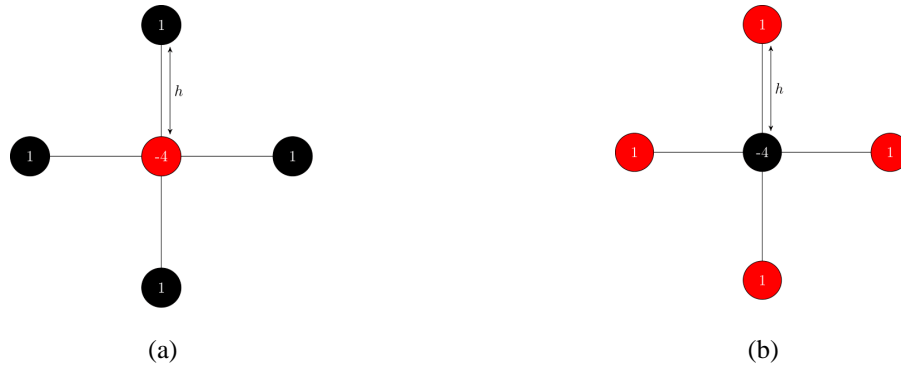


Figure 3. Computational molecules for, (a) red, (b) black nodes, respectively

### 2.3. Modified iterative methods

An extension to the RBSOR method is the modified successive over-relaxation (MSOR) [34] that can be expressed as:

$$U_{i,j}^{(k+1)} = \frac{\alpha}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + (1 - \alpha)U_{i,j}^{(k)} \text{ for all } i+j \text{ even} \quad (15)$$

and

$$U_{i,j}^{(k+1)} = \frac{\beta}{4} [U_{i+1,j}^{(k+1)} + U_{i-1,j}^{(k+1)} + U_{i,j+1}^{(k+1)} + U_{i,j-1}^{(k+1)} - h^2 f_{i,j}] + (1 - \beta)U_{i,j}^{(k)} \text{ for all } i+j \text{ odd} \quad (16)$$

The parameters  $\alpha$  and  $\beta$ , are fixed such that  $0 < \alpha < 2$  and  $0 < \beta < 2$ . The corresponding modified variant of RBSOR, namely the modified accelerated over-relaxation (MAOR), are given as.

$$U_{i,j}^{(k+1)} = \frac{\alpha}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + (1 - \alpha)U_{i,j}^{(k)} \text{ for all } i+j \text{ even} \quad (17)$$

and

$$U_{i,j}^{(k+1)} = \frac{\beta}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + \frac{\tau}{4} [U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i+1,j}^{(k+1)} - U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)} + U_{i,j+1}^{(k+1)} - U_{i,j+1}^{(k)}] + (1 - \beta)U_{i,j}^{(k)} \text{ for all } i+j \text{ odd} \quad (18)$$

Lastly, the modified variant of RBTOR, known as the modified two-parameter over-relaxation (MTOR) are given as (19) and (20).

$$U_{i,j}^{(k+1)} = \frac{\alpha}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + (1 - \alpha)U_{i,j}^{(k)} \text{ for all } i+j \text{ even} \quad (19)$$

and

$$U_{i,j}^{(k+1)} = \frac{\beta}{4} [U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)} - h^2 f_{i,j}] + \frac{\tau}{4} [U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i+1,j}^{(k+1)} - U_{i+1,j}^{(k)}] + \frac{\gamma}{4} [U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)} + U_{i,j+1}^{(k+1)} - U_{i,j+1}^{(k)}] + (1 - \beta)U_{i,j}^{(k)} \text{ for all } i+j \text{ odd} \quad (20)$$

Algorithm 1 describes the main procedures that are used in the implementation of the six methods considered in this study.

---

#### Algorithm 1: The image composition algorithm for the six methods

---

**Input** : Source image  $\mathbf{S}$ , Target image  $\mathbf{G}$ , and pixels inside region  $\partial\mathbf{S}$   
**Output** : New edited image  $\mathbf{F}$

---

```

/* Initialize U and V (to store the previous and updated pixels inside region  $\partial S$  */
initPixels(V, U, S, G)
/* Initialize parameters  $\alpha$ ,  $\beta$ ,  $\tau$ ,  $\gamma$  */
initParams( $\alpha$ ,  $\beta$ ,  $\tau$ ,  $\gamma$ )
while checkConverge(V, U) do
  /* For red nodes (i+j) is even */
  for i := 1 to n-2 do
    for j := 1 to m-2 do
      if isEven(i+j) then
        V(i,j) := computePixels(U,  $\alpha$ ,  $\tau$ ,  $\gamma$ )
      end
    end
  end
  /* For black nodes (i+j) is odd */
  for i := 1 to n-2 do
    for j := 1 to m-2 do
      if isOdd(i+j) then
        V(i,j) := computePixels(U,  $\beta$ ,  $\tau$ ,  $\gamma$ )
      end
    end
  end
end
F := copyPixels(V)
return F

```

Algorithm 1 is only applied to the pixels involved in the composition process. All other pixels outside the region of the composition are ignored during the computation process, thus speed up the overall execution time. With MTOR, two acceleration parameters,  $\omega$ , and  $\tau$ , are available for fine-tuning, thus provide more choice during the composition process. Additionally, by employing the red-black strategy, two additional parameters,  $\alpha$ , and  $\beta$ , can be tuned and thus providing additional parameter values option to further improve the computational time.

### 3. RESULTS AND DISCUSSION

In this work, three sets of images are used to measure the performance of the considered methods. Figure 4 shows all images sets that comprise of source and target images. Table 2 shows the number of pixels inside the image masks that were applied for the three image sets.



SET A (a)



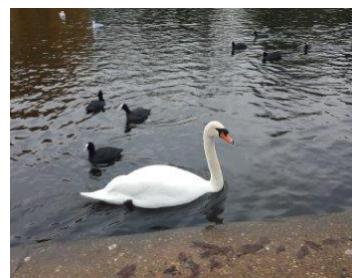
SET B (a)



SET C (a)



SET A (b)



SET B (b)



SET C (b)

Figure 4. These figures are; (a) target, (b) source images for set A, B, and C, respectively

Table 2. Number of pixels inside the image masks

Mask	Ballon	Goose	Eland
Number of pixels	12.392	7.600	7.939

### 3.1. Computational cost measurements

Table 3, Table 4 and Table 5 show the performance of the six tested methods. The proposed MTOR method gives the lowest number of iterations, thus consequently provides the best execution time. All modified variants, namely MSOR, MAOR, and MTOR, perform faster than their corresponding standard variants of RBSOR, RBAOR, and RBTOR. As shown in the three tables, the modified variants reduce the number of iterations by approximately 7% to 14% compared to the standard variants. It is also shown that the TOR variants decrease the number of iterations by approximately 8% to 11% and 34% to 43% compared to the AOR and SOR variants, respectively.

Table 3. Computational cost for image set A

METHODS	Number of iterations	Execution time
RBSOR	1404	19s 271ms
RBAOR	1134	15s 507ms
RBTOR	1046	14s 342ms
MSOR	1298	18s 759ms
MAOR	1019	16s 189ms
MTOR	928	13s 70ms

Table 4. Computational cost for image set B

METHODS	Number of iterations	Execution time
RBSOR	488	5s 893ms
RBAOR	389	4s 748ms
RBTOR	357	4s 393ms
MSOR	450	5s 493ms
MAOR	348	4s 240ms
MTOR	315	3s 818ms

Table 5. Computational cost for image set C

METHODS	Number of iterations	Execution time
RBSOR	575	6s 977ms
RBAOR	465	5s 674ms
RBTOR	429	5s 442ms
MSOR	531	7s 457ms
MAOR	418	5s 866ms
MTOR	380	5s 355ms

It is clearly shown from the results shown in Table 3, Table 4 and Table 5 that the MTOR outperforms the other methods. All variants of TOR are superior to their AOR and SOR variants, in which the modified variants performed slightly faster than their corresponding standard variants. Also, since all methods applied red-black ordering, the proposed methods can be implemented for parallel execution.

### 3.2. Image quality measurements

The image quality measurement process is used to compare the similarity between the final output images. Based on the statistical method analysis of variance (ANOVA) [35], seven metrics are used namely mean square error (MSE), structural similarity index (SSIM) [36], structural content (SC), normalized cross-correlation (NCC), normalized absolute error (NAE), average difference (AD) and maximum difference (MD) [37]. The simple MSE can be written as (19).

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (A_{ij} - B_{ij})^2. \quad (19)$$

where A and B represent the pixel values of reference and target images, and  $(m \times n)$  is the size of the image. MSE is used to measure the difference between pixel values in A and B, in which a smaller value means higher similarity. The ideal MSE value of 0 is obtained when the two images A and B are identical. Another similarity test that can be used to compare the two images is SSIM that is written as (20) [35].

$$SSIM(x, y) = \frac{(2\phi_x\phi_y + C_1)(2\tau_{xy} + C_2)}{(\phi_x^2 + \phi_y^2 + C_1)(\tau_x^2 + \tau_y^2 + C_2)} \quad (20)$$

where  $\phi_x$ ,  $\phi_y$  and  $\tau_x$ ,  $\tau_y$  denote the mean intensity and standard deviation set of image block x and image block y, respectively, while  $\tau_{xy}$  denote their cross-correlation.  $C_1$  and  $C_2$  are small constants value to avoid instability problems when the denominator is too close to zero [38]. If the obtained SSIM is equal to 1, it means the two images are identical. Similarly, SC and NCC values that are equal to 1 can be used to indicate that the two images are identical. SC value can be obtained using the (21).

$$SC = \frac{\sum_{i=1}^m \sum_{j=1}^n (A_{ij})^2}{\sum_{i=1}^m \sum_{j=1}^n (B_{ij})^2} \quad (21)$$

NCC measurement compares the two images and is given as (22).

$$NCC = \sum_{i=1}^m \sum_{j=1}^n \frac{A_{ij} \times B_{ij}}{A_{ij}^2} \quad (22)$$

In contrast, NAE and AD values that are close to 0 means the two images are identical. Both values can be obtained as.

$$NAE = \frac{\sum_{i=1}^m \sum_{j=1}^n (|A_{ij} - B_{ij}|)^2}{\sum_{i=1}^m \sum_{j=1}^n (A_{ij})^2} \quad (23)$$

and

$$AD = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n [A_{ij} - B_{ij}], \quad (24)$$

where A and B are the reference and target images, respectively. Lastly, the MD value that is used to obtain the maximum difference of pixel value between the two images A and B. MD is given the (25).

$$MD = \max (A_{ij} - B_{ij}) \quad (25)$$

If the obtained MD value is very small, it means the similarity between the two images is higher. All these seven metrics are used to compare the similarity of the output images generated by the six tested methods. As shown in Figure 5, Figure 6, and Figure 7, the generated images are all visually identical. The results of the seven metrics measurement are given in Table 6, Table 7 and Table 8.



Figure 5. The output images of set A



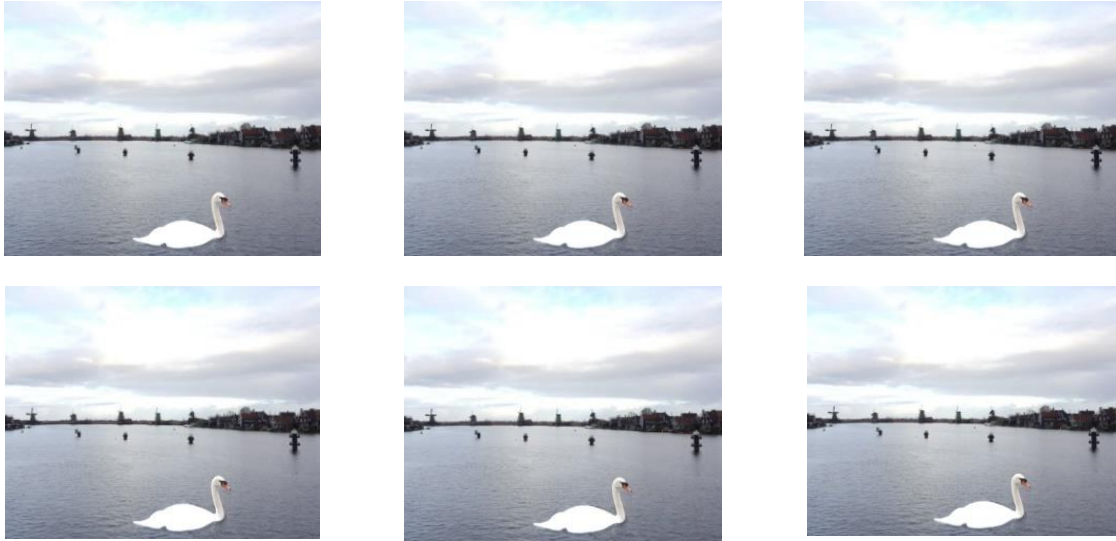


Figure 6. The output images of set B



Figure 7. The output images of set C

Based on the similarity measurement shown in Table 6, Table 7 and Table 8, the MSE, NAE, and AD values are approximately near to 0. Meanwhile, the SSIM and NCC are very close to 1, which means the generated images are all almost identical. The maximum difference (MD) in pixel values is very small (between 2 to 8). The obtained values show that the six methods produce identical images.

The results obtained by the considered algorithms show that red-black strategy provides an alternative to the existing approaches for solving image composition problem. By applying a chessboard like point coloring, the red-black strategy is very suitable for parallel implementation, since the computation for red and black nodes can be done independently.

Table 6. Similarity measurement for image set A

METHODS	SSIM	MSE	SC	NCC	NAE	AD	MD
RBSOR	0.99993	0.23408	0.99912	1.00044	0.00039	0.059655	7
RBAOR	0.99992	0.27711	0.99904	1.00047	0.00042	0.064970	8
RBTOR	0.99991	0.28855	0.99902	1.00048	0.00043	0.066310	8
MSOR	0.99992	0.24488	0.99910	1.00045	0.00039	0.060975	7
MAOR	0.99991	0.28701	0.99902	1.00048	0.00043	0.066140	8
MTOR	0.99991	0.29958	0.99900	1.00049	0.00044	0.067555	8

Table 7. Similarity measurement for image set B

METHODS	SSIM	MSE	SC	NCC	NAE	AD	MD
RBSOR	0.99999	0.00591	0.99995	1.00002	0.00003	0.005580	2
RBAOR	0.99999	0.00646	0.99995	1.00003	0.00003	0.006050	3
RBTOR	0.99999	0.00667	0.99995	1.00003	0.00003	0.006235	3
MSOR	0.99999	0.00601	0.99995	1.00002	0.00003	0.005670	2
MAOR	0.99999	0.00665	0.99995	1.00003	0.00003	0.006220	3
MTOR	0.99999	0.00678	0.99994	1.00003	0.00003	0.006330	3

Table 8. Similarity measurement for image set C

METHODS	SSIM	MSE	SC	NCC	NAE	AD	MD
RBSOR	0.99999	0.11480	0.99900	1.00050	0.00039	0.049680	4
RBAOR	0.99998	0.13669	0.99890	1.00055	0.00042	0.054495	4
RBTOR	0.99998	0.14307	0.99888	1.00056	0.00043	0.055795	5
MSOR	0.99999	0.11947	0.99898	1.00051	0.00039	0.050710	4
MAOR	0.99998	0.14243	0.99888	1.00056	0.00043	0.055670	5
MTOR	0.99998	0.14919	0.99885	1.00057	0.00044	0.057025	5

#### 4. CONCLUSION

For solving the image composition problem, the red-black ordering implementation is used for the six iterative methods considered in this study. The experimental results show that the MTOR method provides the fastest rate of convergence compared to the previous MSOR, MAOR, and all Red-Black variants. Most of the previous works did not utilize the red-black strategy that supports parallelism. Seven image similarity indexes were used to compare the generated images. The similarity tests conducted on the images show that the suggested methods obtained almost identical images with no noticeable differences. In the future, the work can be extended by applying more advanced point iterative methods such as half- and quarter-sweep iteration approaches. The iterative methods based on block can also be considered.

#### ACKNOWLEDGEMENTS

The authors would like to thank the Universiti Malaysia Sabah for financially supporting this research under grant no SGA0117-2019.

#### REFERENCES

- [1] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman, "Background matting: The world is your green screen," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2291-2300.
- [2] Z. Wang, and Z. Yang, "Review on image-stitching techniques," *Multimedia Systems*, vol. 26, pp. 413-430, 2020, doi: 10.1007/s00530-020-00651-y.
- [3] Misha Kazhdan, Ming Chuang, Szymon Rusinkiewicz, and Hugues Hoppe, "Poisson surface reconstruction with envelope constraints," *Computer Graphics Forum*, vol. 39, no. 5, pp. 173-182. 2020, doi: 10.1111/cgf.14077.
- [4] B. Vijayakumari, "Noise removal with filtering techniques," *Advancements in Security and Privacy Initiatives for Multimedia Images*, pp. 133-156. IGI Global, 2021, doi: 10.4018/978-1-7998-2795-5.ch006
- [5] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf, "Flow-edge guided video completion," *European Conference on Computer Vision*, Springer, Cham, 2020, pp. 713-729.
- [6] Hui Zeng, Anjie Peng and Xiangui Kang, "Hiding traces of camera anonymization by poisson blending," *International Conference on Artificial Intelligence and Security*, Springer, Cham, 2020, pp. 98-108, doi: 10.1007/978-3-030-57881-7\_9.
- [7] Patrick Pérez, Michel Gangnet, and Andrew Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313-318, 2003, doi: 10.1145/882262.882269.
- [8] R. Raskar, Adrian Ilie, and Jingyi Yu, "Image fusion for context enhancement and video surrealism," *The 3rd International Symposium on Non-Photorealistic Animation and Rendering, Association for Computing Machinery*, 2004, pp. 85-152.
- [9] Zhouyu Du, Antonio Robles-Kelly, and Fangfang Lu "Robust surface reconstruction from gradient field using the L1 norm," *The 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, 2007, pp. 203-209, doi: 10.1109/DICTA.2007.4426797.
- [10] Dikpal Reddy, Amit Agrawal, and Rama Chellappa, "Enforcing integrability by error correction using l1-minimization," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2350-2357, doi: 10.1109/CVPR.2009.5206603.
- [11] Michael W. Tao, Micah K. Johnson, and Sylvain Paris, "Error-tolerant image compositing," *Int. J. Comput. Vision*, vol. 103, no. 2, pp. 31-44, 2010, doi: 10.1007/978-3-642-15549-9\_3.
- [12] Rida Sadek, Gabriele Facciolo, Pablo Arias, and Vicent Caselles, "A variational model for gradient-based video editing," *International Journal of Computer Vision*, vol. 103, pp. 127-162, 2013, doi: 10.1007/s11263-012-0597-5.

- [13] J.-M. Morel, A.B. Petro, and C. Sbert, "Fourier implementation of poisson image editing," *Pattern Recogn. Lett.*, vol. 33, pp. 342-348, 2012, doi: 10.1016/j.patrec.2011.10.010.
- [14] Nicolas Limare, Ana Bel'en Petro, Catalina Sbert, and Jean-Michel Morel, "Retinex poisson equation: a model for color perception," *Image Processing On Line*, vol. 1, pp. 39-50, 2011, doi: 10.5201/ipol.2011.lmpsrpe.
- [15] K.F. Hussain, and R.M. Kamel, "Efficient poisson image editing," *Electronic Letters on Computer Vision and Image Analysis*, vol. 14, no. 2, pp. 45-57, 2015, doi: 10.5565/rev/elcvia.765.
- [16] M. Afifi, and K.F. Hussain, "MPB: A modified Poisson blending technique," *Computational Visual Media*, vol. 1, no. 4, pp. 331-341, 2015, doi: 10.1007/s41095-015-0027-z.
- [17] J. Mat'ias Di Martino, Gabriele Facciolo, and Enric Meinhardt-Llopis, "Poisson image editing," *Image Processing On Line*, vol. 6, pp. 300-325, 2016, doi: 10.5201/ipol.2016.163.
- [18] Eng Jeng Hong, Azali Saudi, and Jumat Sulaiman, "Numerical assessment for poisson image blending problem using MSOR iteration via five-point Laplacian operator," *Journal of Physics: Conference Series*, vol. 890, 2017, doi: 10.1088/1742-6596/890/1/012010.
- [19] Jeng Hong Eng, Azali Saudi, and Jumat Sulaiman, "Performance analysis of the explicit decoupled group iteration via five-point rotated laplacian operator in solving Poisson image blending problem," *Indian Journal of Science and Technology*, vol. 11, no. 12, 2018, doi: 10.17485/ijst/2018/v11i12/120852.
- [20] Saudi, A., and J. Sulaiman, "Indoor path planning using harmonic functions via half-sweep arithmetic mean method," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 524-528, 2019, doi: 10.12785/ijcds/080510.
- [21] N A Syafiq, M Othman, N Senu and F Ismail, "An experimental study of the modified accelerated overrelaxation (MAOR) scheme on stationary helmholtz equation," *Journal of Physics: Conference Series*, vol. 1366, 2019, doi: 10.1088/1742-6596/1366/1/012093.
- [22] N.H.M. Ali, and L.S. Chong, "Group accelerated OverRelaxation methods on rotated grid," *Applied Mathematics and Computation*, vol. 191, pp. 533-542, 2007, doi: 10.1016/j.amc.2007.02.131.
- [23] W.S. Yousif, and M.M. Martins, "Explicit de-coupled group AOR method for solving elliptic partial differential equations," *Neural, Parallel Sci. Comput.*, vol. 16, no. 4, pp. 531-541, 2008.
- [24] J. L. Santos, W. S. Yousif and M. M. Martins, "The explicit group TOR method," *Neural Parallel and Scientific Computations*, vol. 20, no. 3-4, pp. 459-474, 2012.
- [25] Fang Cao, Tianwei Wu, Chuan Qin, Zhenxing Qian, and Xinpeng Zhang, "New paradigm for self-embedding image watermarking with poisson equation," *International Workshop on Digital Watermarking*, 2020, pp. 184-191.
- [26] Changmiao Hu, Lian-Zhi Huo, Zheng Zhang and Ping Tang, "Multi-temporal landsat data automatic cloud removal using poisson blending," *IEEE Access*, vol. 8, pp. 46151-46161, 2020, doi: 10.1109/ACCESS.2020.2979291.
- [27] Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsub Ham, Kwanghoon Sohn, and Minh N. Do, "Fast global image smoothing based on weighted least squares," *IEEE Transactions on Image Processing*, vol. 23, pp. 5638-5653, 2014, doi: 10.1109/TIP.2014.2366600.
- [28] Yang Pan, Mingwu Jin, Shunrong Zhang, and Yue Deng, "TEC map completion using DCGAN and poisson blending," *Space Weather*, vol. 18, no. 5, 2020, doi: 10.1029/2019SW002390.
- [29] Nordin Saad, Andang Sunarto, and Azali Saudi, "Accelerated red-black strategy for image composition using laplacian operator," *International Journal of Computing and Digital Systems*, 2020.
- [30] D. Young, "Iterative methods for solving partial difference equations of elliptic type," *Transactions of the American Mathematical Society*, vol. 76, pp. 92-111, 1954, doi: 10.1090/S0002-9947-1954-0059635-7.
- [31] A. Hadjidimos, "Accelerated overrelaxation method," *Mathematics of Computation*, vol. 32, no. 141, pp. 149-157, 1978, doi: 10.2307/2006264.
- [32] J. Kuang, and J. Ji, "A survey of AOR and TOR methods," *Journal of Computational and Applied Mathematics*, vol. 24, no. 1-2, pp. 3-12, 1988, doi: 10.1016/0377-0427(88)90340-8.
- [33] D. J. Evans, "Parallel SOR iterative methods," *Parallel Computing*, vol. 1, pp. 3-18, 1984, doi: 10.1016/S0167-8191(84)90380-6.
- [34] D. R. Kincaid, and D.M. Young, "The modified successive overrelaxation method with fixed parameters," *Mathematics of Computation*, vol. 26, pp. 705-717, 1972, doi: 10.1090/S0025-5718-1972-0331746-2.
- [35] S. H. A. Hashim, F. A. Hamid, J. J. Kiram, and J. Sulaiman, "The relationship investigation between factors affecting demand for broadband and the level of satisfaction among broadband customers in the South East Coast of Sabah, Malaysia," *Journal of Physics: Conference Series*, vol. 890, 2017, doi: 10.1088/1742-6596/890/1/012149.
- [36] Zhou Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004, doi: 10.1109/TIP.2003.819861.
- [37] Farida Memon, Mukhtiar Ali Unar, and Sheeraz Memon, "Image quality assessment for performance evaluation of focus measure operators," *Mehran University Research Journal of Engineering and Technology*, vol. 34, no. 4, pp. 379-386, 2016.
- [38] K. H. Thung, and P. Raveendran, "A survey of image quality measures," *2009 IEEE International Conference for Technical Postgraduates (TECHPOS)*, 2009, pp. 1-4, doi: 10.1109/TECHPOS.2009.5412098.

**BIOGRAPHIES OF AUTHORS**

**Zakariah Aris** currently works at the Faculty of Computing and Informatics, Universiti Malaysia Sabah (UMS) as a researcher and Senior Lecturer. My research interest is in Prediction, Artificial Networks. Image Processing, Project Management in Oil & Gas, and Fiber Optic. He has experience in downstream Oil and Gas mega projects in PETRONAS i.e. Peninsular Gas Utilization (PGU) and Gas Processing Plant (GPP); collaborating with Nova Corporation Inc. (Canada), and Toyo Engineering Corporation (Japan) besides controlling pipeline valves with SCADA system.



**Nordin Saad** obtained his master's degree in Information Technology from Universiti Kebangsaan Malaysia (UKM) and his bachelor's degree in Information Technology from Universiti Utara Malaysia (UUM). He joined Universiti Malaysia Sabah (UMS) as a lecturer in Information Technology in May 1999. Currently, he is a senior lecturer in the Faculty of Computing and Informatics, UMS. His research interests include ICT Infrastructure, IoT, AI, and Smart Agriculture. He also works on the modeling and simulation of image compositions.



**A'Qilah Ahmad Dahalan** holds a Master's degree in Applied Sciences and a Bachelor's degree in Mathematics from Universiti Malaysia Sabah. She is currently attached as a lecturer at the Department of Mathematics, Centre for Defence Foundation Studies, National Defence University of Malaysia. She is part of the committee members in Robotics Lab which focuses on the development of intelligent robots. Her research interests include the areas of computational mathematics, fuzzy partial differential equations, and path planning



**Andang Sunarto** obtained his Ph.D. from Universiti Malaysia Sabah, a master's degree in Computer Science from Universitas Gajah Mada, and a bachelor's degree in Statistics from Universitas Islam Indonesia. He joined the Institut Agama Islam Negeri, Bengkulu, Indonesia, as a senior lecturer (Computer Science and Mathematics) in September 2006. He is a researcher and appointed as a fellow working in the field of Numerical Analysis, Algorithm, C++ Programming, and Scientific Calculation, Modelling, and Simulation. His research employs the application of IoT, Numerical Analysis to improve the model in fractional partial differential equations. He also works in the modeling and simulation of numerical analysis.



**Azali Saudi** holds a Ph.D. degree in Mathematics from Universiti Malaysia Sabah (UMS), a Master's degree in Artificial Intelligence from the University of Edinburgh, and a Computer Science degree from Universiti Kebangsaan Malaysia. He is currently attached as Associate Professor at the Faculty of Computing and Informatics, UMS. He also acts as Director of Research and Innovation Management Centre at UMS. His research interests include the areas of computational mathematics, artificial intelligence, and robot vision. He also leads research projects related to intelligent robotics, computer vision, deep learning, and computational mathematics.