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

Zakariah Aris¹, Nordin Saad², A’qilah Ahmad Dahalan³, Andang Sunarto⁴, Azali Saudi⁵
¹,²,⁵Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia
³Centre for Defence Foundation Studies, National Defence University of Malaysia, Kuala Lumpur, Malaysia
⁴Institut Agama Islam Negeri, Bengkulu, Indonesia

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.

Keywords:
Finite difference
Image composition
MAOR
Modified iterative method
MSOR
MTOR
Poisson equation
Red-black ordering

This is an open access article under the CC BY-SA license.

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
device 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,
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
dge-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

<table>
<thead>
<tr>
<th>Authors</th>
<th>Application</th>
<th>Method</th>
<th>Authors</th>
<th>Application</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>[7]</td>
<td>Poisson image editing</td>
<td>Successive over-</td>
<td>[19]</td>
<td>Image blending</td>
<td>SOR and group iteration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>relaxation (SOR)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>blending</td>
<td></td>
</tr>
<tr>
<td>[18]</td>
<td>Image blending</td>
<td>Modified SOR</td>
<td>[29]</td>
<td>Image composition</td>
<td>MAOR</td>
</tr>
</tbody>
</table>

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*. 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:

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

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

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

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

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

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].

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

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

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

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).

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

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](image1.png)

![Figure 2: The finite grid with red-black ordering](image2.png)

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

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

and

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

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

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

and

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

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

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

and

\[
U^{(k+1)}_{i,j} = \frac{\omega}{4} \left[ U^{(k)}_{i+1,j} + U^{(k)}_{i-1,j} + U^{(k)}_{i,j+1} + U^{(k)}_{i,j-1} - h^2 f_{i,j} \right] + \frac{\tau}{4} \left[ U^{(k+1)}_{i-1,j} - U^{(k)}_{i-1,j} \right] + \frac{\tau}{4} \left[ U^{(k+1)}_{i+1,j} - U^{(k)}_{i+1,j} \right] + (1 - \omega)U^{(k)}_{i,j} \quad \text{for all} \ i+j \text{ odd} \tag{14}
\]

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 \(\varepsilon\) is satisfied.

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

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

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

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

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 $S$, Target image $G$, and pixels inside region $\delta S$  

**Output**: New edited image $F$
Modified iterative method with red-black ordering for image composition using poisson (Zakariah Aris)

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.

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

<table>
<thead>
<tr>
<th>METHODS</th>
<th>Number of iterations</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBSOR</td>
<td>1404</td>
<td>19s 271ms</td>
</tr>
<tr>
<td>RBAOR</td>
<td>1134</td>
<td>15s 507ms</td>
</tr>
<tr>
<td>RBTOR</td>
<td>1046</td>
<td>14s 342ms</td>
</tr>
<tr>
<td>MSOR</td>
<td>1298</td>
<td>18s 759ms</td>
</tr>
<tr>
<td>MAOR</td>
<td>1019</td>
<td>16s 189ms</td>
</tr>
<tr>
<td>MTOR</td>
<td>928</td>
<td>13s 70ms</td>
</tr>
</tbody>
</table>

### Table 4. Computational cost for image set B

<table>
<thead>
<tr>
<th>METHODS</th>
<th>Number of iterations</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBSOR</td>
<td>488</td>
<td>5s 893ms</td>
</tr>
<tr>
<td>RBAOR</td>
<td>389</td>
<td>4s 748ms</td>
</tr>
<tr>
<td>RBTOR</td>
<td>357</td>
<td>4s 393ms</td>
</tr>
<tr>
<td>MSOR</td>
<td>450</td>
<td>5s 493ms</td>
</tr>
<tr>
<td>MAOR</td>
<td>348</td>
<td>4s 240ms</td>
</tr>
<tr>
<td>MTOR</td>
<td>315</td>
<td>3s 818ms</td>
</tr>
</tbody>
</table>

### Table 5. Computational cost for image set C

<table>
<thead>
<tr>
<th>METHODS</th>
<th>Number of iterations</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBSOR</td>
<td>575</td>
<td>6s 977ms</td>
</tr>
<tr>
<td>RBAOR</td>
<td>465</td>
<td>5s 674ms</td>
</tr>
<tr>
<td>RBTOR</td>
<td>429</td>
<td>5s 442ms</td>
</tr>
<tr>
<td>MSOR</td>
<td>531</td>
<td>7s 457ms</td>
</tr>
<tr>
<td>MAOR</td>
<td>418</td>
<td>5s 866ms</td>
</tr>
<tr>
<td>MTOR</td>
<td>380</td>
<td>5s 355ms</td>
</tr>
</tbody>
</table>

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).

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

(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 measures 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. Similarly, SC and NCC values that are equal to 1 can be used to indicate that the two images are identical. Another similarity test that can be used to compare the two images is SSIM that is written as (20) [35].

\[
SSIM(x, y) = \frac{(2\varphi_x\varphi_y + C_1)(2\tau_{xy} + C_2)}{(\varphi_x^2 + \varphi_y^2 + C_1)(\tau_x^2 + \tau_y^2 + C_2)}
\]  

(20)

where \(\varphi_x\), \(\varphi_y\) 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 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).
Modified iterative method with red-black ordering for image composition using poisson ...

(Zakariah Aris)
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>
<thead>
<tr>
<th>METHODS</th>
<th>SSIM</th>
<th>MSE</th>
<th>SC</th>
<th>NCC</th>
<th>NAE</th>
<th>AD</th>
<th>MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBSOR</td>
<td>0.99993</td>
<td>0.23408</td>
<td>0.99912</td>
<td>1.00044</td>
<td>0.00039</td>
<td>0.059655</td>
<td>7</td>
</tr>
<tr>
<td>RBAOR</td>
<td>0.99992</td>
<td>0.27711</td>
<td>0.99904</td>
<td>1.00047</td>
<td>0.00042</td>
<td>0.064970</td>
<td>8</td>
</tr>
<tr>
<td>RBTOR</td>
<td>0.99991</td>
<td>0.28855</td>
<td>0.99902</td>
<td>1.00048</td>
<td>0.00043</td>
<td>0.066310</td>
<td>8</td>
</tr>
<tr>
<td>MSOR</td>
<td>0.99992</td>
<td>0.24488</td>
<td>0.99910</td>
<td>1.00045</td>
<td>0.00039</td>
<td>0.060975</td>
<td>7</td>
</tr>
<tr>
<td>MAOR</td>
<td>0.99991</td>
<td>0.28701</td>
<td>0.99902</td>
<td>1.00048</td>
<td>0.00043</td>
<td>0.066140</td>
<td>8</td>
</tr>
<tr>
<td>MTOR</td>
<td>0.99991</td>
<td>0.29958</td>
<td>0.99900</td>
<td>1.00049</td>
<td>0.00044</td>
<td>0.067555</td>
<td>8</td>
</tr>
</tbody>
</table>
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


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