Electronic Letters on Computer Vision and Image Analysis 14(2):45-57; 2015

Efficient Poisson Image Editing

Khaled F. Hussain[*] Rasha M. Kamel[†]

[*]*Faculty of Computers and Information, Assiut University, Egypt*
[†]*Faculty of Science, Assiut University, Egypt*

Received 4th May 2015; accepted 11th Dec 2015

Abstract

Image composition refers to the process of composing two or more images to create a natural output image. It is one of the important techniques in image processing. In this paper, two efficient methods for composing color images are proposed. In the proposed methods, the Poisson equation is solved using image pyramid and divide-and-conquer methods. The proposed methods are more efficient than other existing image composition methods. They reduce the time taken in the composition process while achieving almost identical results using the previous image composition methods. In the proposed methods, the Poisson equation is solved after converting it to a linear system using different methods. The results show that the time for composing color images is decreased using the proposed methods.

*Keywords***:** image processing, gradient methods, image fusion

1. Introduction

In recent years, many researches and applications have been developed for composing images because the number of people that are interested in changing their images is increased [1]. Image composition is the process of composing two images to create a new image. In the composition process, the user draws a boundary around the desired object in the source image which needs to be cloned into the destination image. A binary mask image is generated for the selected region.

There are several techniques for image composition such as alpha blending and gradient domain methods. In alpha blending methods, images are composed by using an accurate alpha matte for each element. Recent alpha blending techniques can be divided into two categories: propagation-based matting [2-8] and sampling-based matting [9-13]. Gradient domain compositing is an important technique in computer vision. It is the process of combining two images by copying the gradients of the pixels from the source image into the target image. It is used in many applications such as seamless cloning [14-22], seamless video editing [23-24], seamless stitching [25-27], shadow removal [28], inpainting [29-31], gradient domain painting [32,33], scene completion [34,35], and image smoothing [36-38]. Some of gradient domain methods compose images by solving the Poisson equation in the specified region. The Poisson equation is solved by transforming it into a large sparse linear system of equations

$$Ax = b \tag{1}$$

Poisson Image Editing (PIE) Perez et al. [14] is a gradient domain image editing method. This method solves the Poisson equation with the Dirichlet boundary conditions. This method calculates the Laplacian of an unknown function for the pixels in the edited region and the unknown function values over the boundary of the region. The PIE method [14] may give unacceptable results where the user draws a boundary around the object in the source image that intersects with prominent regions in the destination image. Jia et al. [15] suggested a new method to solve this issue. This method makes the user draw a boundary around the desired object and then search for an optimized boundary which is not intersected with any prominent regions of the source and target images. Qin et al. [16] introduced an improved image editing method to solve the color

Correspondence to: khussain@aun.edu.eg

Recommended for acceptance by Filiberto Pla
http://dx.doi.org/10.5565/rev/elcvia.765
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

inconsistency problem in PIE [14] by adding an inner Drichilet boundary condition which divides the unknown region into two regions with two boundaries. The outer boundary values are equal to the values of the destination image and the inner boundary values are equal to source image values. The pixels in the region inside the inner boundary are equal to object values. Then, the values of the pixels in the region between inner and outer boundaries are computed by solving the Poisson equation. So the Poisson equation is solved for the region, which is not too large.

The PIE method in [14] solves the Poisson equation for the color image using each color channel (R, G, B) independently. Dizdaroğlu and İkibaş [17, 18] introduced a method for editing color images by using color information. This method takes into account that each color channel has an effect on other color channels on calculating the gradient of the image. This minimizes the color inconsistency resulting from PIE [14]. Tao et al. [19] presented an approach for image compositing for decreasing the bleeding artifacts in the result image without changing the boundary that the user draws.

The PIE method [14] requires solving a large sparse linear system which is time consuming and memory intensive. Farbman et al. [20] introduced a new approach for seamless image cloning based on Mean Value Coordinates (MVC) [39]. This method avoids solving such linear system by using an interpolation technique to compute the intensity of the pixels in the selected region. The main advantage of the MVC method [20] is that it takes a small amount of memory and all the computation can be parallelized which makes it fast.

Seamless image composition methods can be expressed as boundary value problems. These problems can be solved by constructing a smooth membrane that interpolates the differences between source and destination images across a specified region. Farbman et al. [21] constructed such membrane by approximating Shepard's scattered data interpolation method [40] using a convolution pyramid. Jia and Wu [22] presented a new method for Seamless Image Cloning based on Derivative and Intensity Interpolation (SICDII). This method uses the Discrete Wavelet Transform (DWT) to decompose the image into four parts. This method solves the Poisson equation for the approximation parts and uses the MVC method on the other details parts. The SICDII method [22] takes less amount of memory for computation and is faster than the MVC method [20].

In this paper, two efficient methods for composing color images are proposed. The proposed methods solve the Poisson equation using image pyramid and divide-and-conquer methods. The proposed methods are efficient because they reduce the time taken in the composition process while achieving almost identical result. The results show that the time taken by the proposed methods to generate the output image is less than the time taken by other existing methods.

The remainder of this paper is structured as follows. The main concept of the PIE method is explained in Section 2. A detailed explanation of the proposed methods is introduced in Section 3. The experimental results and conclusions of the paper are shown in Sections 4, 5, respectively.

2. Poisson Image Editing

PIE is the process of extracting an object from source image g and pasting it into a destination image f^* to create a new realistic image f. Let Ω be the selected region on the source image with the boundary $\partial\Omega$. P´erez et al. [14] creates a guidance vector field v which is based only on source image called seamless cloning, source and destination images called mixed seamless cloning. Then, P´erez et al. [14] set a new intensity value f for all the pixels in the region Ω that minimize the difference between the gradient of f (∇f) and the vector field v. This method assigns the intensity values of f at the boundary of the region $\partial\Omega$ to the values of the destination image f^* ($f = f^*$ over $\partial\Omega$), as shown in figure 1.

$$min_f \iint_\Omega |\nabla f - v|^2 \; with \; f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{2}$$

The solution of the minimization problem is the unique solution of the Poisson equation with the Dirichilet boundary condition.

$$\Delta f = \operatorname{div} v \; over \;\; \Omega \;\; with \;\; f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{3}$$

For a simple seamless cloning, the guidance field v is set to be the gradient of the source image: $\mathrm{v} = \nabla g$. Thus, the previous equation becomes

$$\Delta f = \Delta g \; over \; \Omega \; with \;\; f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{4}$$

where Δ is the Laplacian operator. The discrete solution to the previous equation (4) is equal to the solution of the linear system of equation (1) where A is $(n \times n)$ matrix, n is the number of the unknown pixels in the selected region. This system is solved three times for the RGB images for each color channel separately. The final image results from merging the solved color channel values.

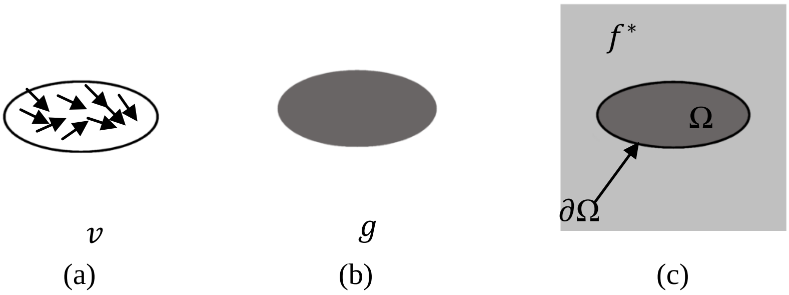


Figure 1. (a) Guided vector field. (b) Source image. (c) Destination image.

3. Efficient Poisson Image Editing Using Pyramid

The composition of two images based on the Poisson equation requires solving a linear system with n unknowns to compute the values of n pixels. In this work, an improved image composition method based on the Poisson equation is introduced to decrease the time taken in the composition process. The proposed methods solve the Poisson equation using image pyramid and divide-and-conquer methods [41]. Image pyramid and divide-and-conquer methods are used to speed up many different types of image operations.

Using the proposed methods, the time taken in solving a large problem at one time is greater than the time taken if it is divided into smaller problems. In the proposed methods, two and three pyramid levels are constructed to increase the efficiency of the composition. The proposed methods start from the third pyramid level. At the third level of the pyramid, the source and destination images are combined together to generate the composite image. PIE method is used to compose the source and destination images in the third level of the pyramid. This method sets the intensity values of the pixels at the boundary of the selected region to the values of the destination image. Then, the Poisson equation is solved to compute the intensity values for the pixels in the interior region. Thus, the border of the cloned object in the composite image is smooth. After generating the composite image from the third pyramid level, it is used in the second level of the pyramid. At the second level, the source and destination images are combined by solving the Poisson equation taking the Dirichlet boundary condition from the composite image results from the third level. So, the composite image generated from the second level is realistic and smooth. At the second level, the unknown region is divided into a number of small unknown regions and the sparse linear system is solved for each small region separately. Finally, the process that occurred in the second level is repeated in the first level to generate the final composite image.

As explained before, the proposed methods are based on partitioning the unknown region into a number of small regions solving the sparse linear system for each small region separately. For the unknown region, the complexity of the Lower-Upper (LU) factorization method for solving the linear system in equation (1) is approximately $O\left(\frac{2}{3}n^3\right)$ [42] where the size of A $is\, n \times n$. If the unknown region is divided into small regions, then the time cost of solving the system in equation (1) for the unknown region using LU factorization is equal to the summation of the time cost for solving it for each small region separately, as shown in the following equation:

$$\sum_{i=1}^{r} \frac{2}{3} m_i^3 = \frac{2}{3} \sum_{i=1}^{r} m_i^3 = \frac{2}{3}(m_1^3 + m_2^3 + \cdots + m_r^3) = T_1 \tag{5}$$

where m_i is the number of the unknown pixels in the region i and r is the number of small regions. The complexity T for solving the linear system for all regions once is given by the following equation:

$$T = \frac{2}{3} n^3 \tag{6}$$

Since the total number of the unknown pixels equal to the summation of the unknown pixels in all small regions $n = \sum_{i=1}^{r} m_i$, then the equation (6) becomes:

$$T = \frac{2}{3}\left(\sum_{i=1}^{r} m_i\right)^3 = \frac{2}{3}(m_1 + m_2 + \cdots + m_r)^3 \tag{7}$$

The expansion of equation (7) gives:

$$\begin{aligned} T = \frac{2}{3}(m_1^3 + m_2^3 + \cdots + m_r^3 + 3m_1^2 m_2 + 3m_1^2 m_3 + \cdots + 3m_1^2 m_r + 3m_2^2 m_1 + 3m_2^2 m_3 + \cdots \\ + 3m_2^2 m_r \ + 3m_r^2 m_1 + 3m_r^2 m_2 + \cdots + 3m_r^2 m_{r-1} + 6m_1 m_2 \ldots m_r = T_1 + T_2 \end{aligned} \tag{8}$$

The above equation shows that the time complexity for solving the linear system for all unknown pixels at one time T equals the time cost of solving it for all small regions separately T_1 plus additional term T_2. Thus, the time cost of solving the linear system for the unknown region after dividing the unknown region into small regions is lower than the time cost of solving it at one time.

3.1 Partitioning the unknown Region into Thin Slices (PRTS)

In this method, the unknown region is divided into thin slices, as shown in figure 2. Then, the Poisson equation is solved for each slice separately taking the Dirichlet boundary condition from the composite image that results from the previous pyramid level. The steps of this method are described in the following algorithm:

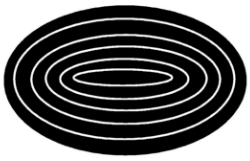


Figure 2. The unknown region is divided into thin slices

PRTS Algorithm: Efficient PIE based on partitioning the unknown region into thin slices

```
Inputs: mask of the unknown region, source, and destination images.
Output: final composite image.
1. Construct the pyramid levels for source, destination, and mask images using Gaussian pyramid
   method in [41].
2. Compose the source and destination images at the third level of the pyramid using PIE method [14].
3. At the second level of the pyramid do the following:
    a. Divide the unknown region in the mask into thin slices using the following steps:
        i. Erode the unknown region using erode morphological operation.
        ii. Apply the exclusive-or (XOR) between the eroded unknown region and the original
            unknown region.
    b. for each slice do
        i. Compose source and destination images using the PIE method [14].
        ii. Solve the Poisson equation with the Dirichlet boundary condition taken from the generated
            composite image from step 1.
       end for
4. Compute the final composite image by repeating the previous step at the first level of the pyramid.
```

3.2 Partitioning the unknown Region into Small Square Blocks (PRSSB)

In this method, the unknown region is divided into small square blocks. The shape of the regions in the boundary of the unknown region is incomplete square (not all pixels in the block are unknowns) and the shape of the regions in the interior is square (all pixels are unknowns), as shown in figure 3. Then, the

Poisson equation is solved for each region separately taking the Dirichlet boundary condition from the output image of the previous pyramid's level. The steps are described in the following algorithm:

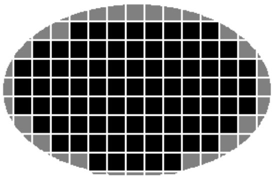


Figure 3. The unknown region is divided into small square blocks

PRSSB Algorithm: Efficient PIE based on partitioning the unknown region into small square blocks

Inputs: mask of the unknown region, source and destination images.

Output: final composite image.

```
1. Construct the pyramid levels for source, destination, and mask images using Gaussian pyramid method in [41].
2. Compose the source and destination images at the third level of the pyramid using PIE method [14].
3. At the second level of the pyramid do the following:
   a. Divide the mask of the unknown region into small square blocks each block of size $BS$.
   b. Create the sparse matrix $SA$ that will be used if all the pixels in the block are unknowns by doing the following steps:
      i. Initialize the $SA$ matrix with size ($BS^2 \times BS^2$) and diagonal elements 4s, where $BS^2$ is the number of the pixels in the block.
      ii. for $r = 0$ to $BS - 1$ do
            for $c = 0$ to $BS - 1$do
               Determine the set of neighbor pixels $NB$ around the pixel $P(r, c)$
               for each neighbor pixel $NB1$ do
                  if $NB1$ is a pixel on the boundary then
                        Set $SA(r, c) = 0$
                  else
                        Set $SA(r, c) = -1$
                  end if
               end for
            end for
          end for
   c. for each block in the mask do
      i. Compose source and destination images using the PIE method [14].
      ii. Solve the Poisson equation with the Dirichlet boundary condition taken from the generated composite image from step 1.
      iii. if all the Pixels on the block are unknowns then
              Use the pre-computed matrix $SA$ in solving the linear system.
           else
              Compute the sparse matrix $A$ for solving the linear system.
           end if
      end for
4. Compute the final composite image by repeating the previous step at the first level of the pyramid.
```

The proposed methods are more efficient than previous image composition methods because they minimize the processing time.

4. Experimental Results

In this section, the two proposed methods (PRSSB and PRTS) were applied to a number of instances [43] with different number of unknown pixels, as shown in table 1. Each instance consists of two color images (source and destination) and mask image, as shown in figure 4.

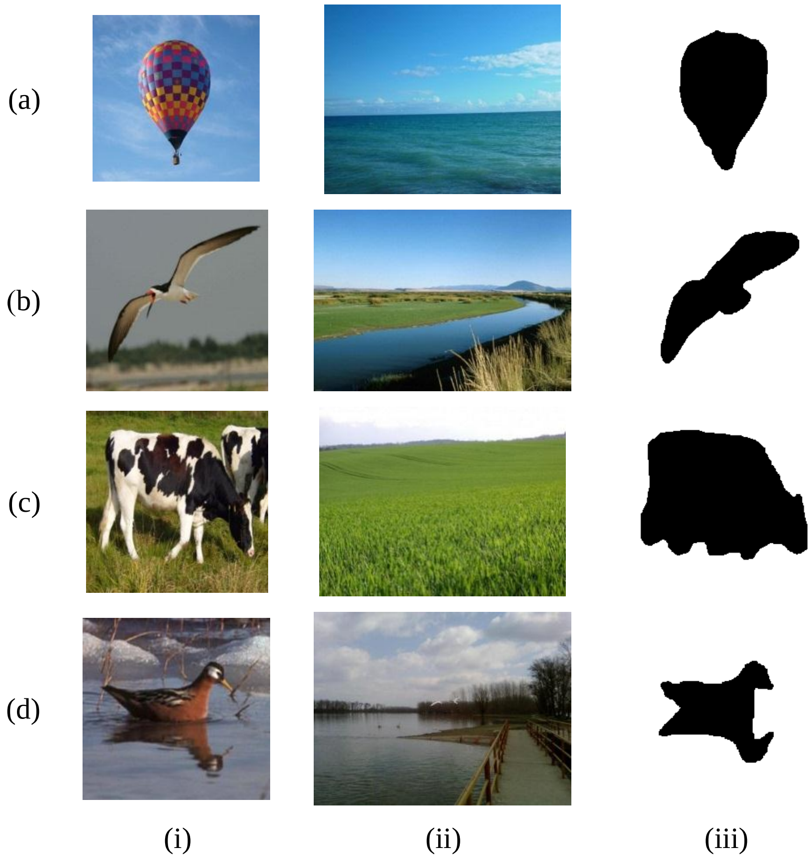


Figure 4. (i) Source image. (ii) Destination image. (iii) Object mask.

Table 1. Number of pixels in the unknown region for all instances

Instance	(a) Balloon	(b) Bird	(c) Cow	(d) Phalarope
Number of pixels	356,409	74,529	146,689	42,983

The proposed methods with two pyramid levels (PRSSB2 and PRTS2) and with three pyramid levels (PRSSB3 and PRTS3) were compared with methods in [14, 20, 22]. In the composition process, the sparse linear system equation (1) was solved three times for R, G, and B colors separately using different methods [44, 45]:

1. Preconditioned Conjugate Gradient (PCG).
2. Bi-Conjugate Gradient (BCG).
3. Successive-Over-Relaxation (SOR).
4. Lower-Upper factorization (LU).
5. Lower-Diagonal-Lower factorization (LDL).

4.1 Time cost measurements

This section shows the composition time of methods in [14, 20, 22] and the composition time of the two proposed methods with two and three pyramid levels, as shown in tables 2-5.

Table 2. Average time in seconds for PIE method

Method	Balloon	Bird	Cow	Phalarope
PCG	546.84	25.47	96.18	14.68
BCG	736.82	32.79	208.77	15.38
SOR	750.33	33.98	209.03	18.67
LU	> 60 min	40.21	219.16	26.71
LDL	**518.47**	**24.53**	**88.33**	**13.28**

Table 2 shows that LDL method has the lowest composition time compared to other tested methods, especially for high resolution image.

Table 3. Average time in seconds for MVC and SICDII methods

MVC				SICDII			
Balloon	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
183.49	15.61	42.38	9.44	100.96	9.11	24.02	7.86

Results in table 3 show that the time taken to compose images using the MVC method [20] is lower than the time taken from the PIE method [14]. Results also show that composing images using the SICDII method [22] takes lower time than the time taken using either PIE method [14] or MVC method [20].

Table 4. Average time in seconds for PRTS2 and PRTS3 methods

	PRTS2				PRTS3			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
PCG	71.72	7.42	20.91	4.41	41.07	8.36	17.36	3.94
BCG	83.61	8.07	21.81	4.95	46.14	8.75	19.02	4.81
SOR	115.29	11.86	38.28	6.12	75.64	8.50	27.24	5.99
LU	85.49	8.14	23.68	5.38	41.03	6.95	15.26	4.28
LDL	**56.06**	**7.03**	**13.73**	**3.61**	**38.51**	**7.14**	**16.18**	**4.11**

Table 5. Average time in seconds for PRSSB2 and PRSSB3 methods

	PRSSB2				PRSSB3			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
PCG	47.21	4.88	13.79	3.71	20.01	3.79	7.52	2.88
BCG	78.98	12.54	21.45	3.63	41.16	11.48	15.97	4.06
SOR	116.92	12.43	32.77	3.88	41.66	8.83	15.85	3.26
LU	62.93	4.43	17.22	3.36	12.58	2.61	5.18	2.41
LDL	**36.60**	**4.25**	**11.24**	**1.88**	**12.24**	**2.56**	**4.51**	**1.86**

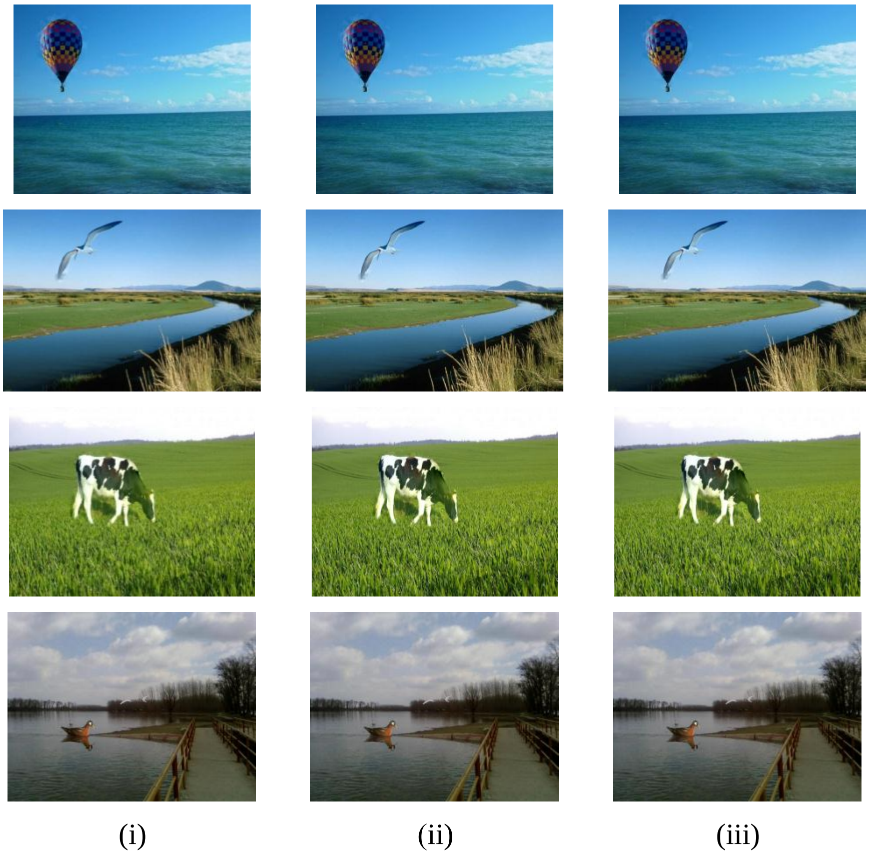
Results in tables 4 and 5 show that the time taken in the composition process for all instances is decreased when applying PRTS and PRSSB methods with two and three pyramid levels compared to methods in [14, 20, 22]. Results also show that LDL method has the lowest composition time compared to other tested methods.

4.2 Quality measurements

The quality of the images is measured using subjective and objective methods. In subjective image quality assessment the visual quality of the images is assessed by humans. In objective methods, the quality is measured by algorithms such as Mean Square Error (MSE) [46] and Structural Similarity Index (SSIM) [47]. Subjective image quality assessment methods are based on the presentation of a questionnaire form to a number of observers. Each observer is asked to evaluate the quality of the image by giving a score between 0 and 10.

In this paper, the quality of the composite images is measured using two objective metrics (MSE and SSIM) and one subjective metric using a questionnaire form. The MSE and SSIM are used to compare between the composite images generated by the two proposed methods and output images produced by PIE method [14]. The SSIM test generates values between -1 and 1. If the SSIM value between two images is equal 1, this means that the images are identical. In the MSE test, the images are identical if the MSE value is equal 0.

A comparison between the output images from PRTS and PRSSB methods and the output images from PIE method [14] are shown in figure 5. The output images in figure 5 show that visually there are no differences between the images generated by PRTS or PRSSB methods and output images from the PIE method in [14].



(i) (ii) (iii)

Figure 5. (i) Image produced by the PIE method [14]. (ii) Image produced by the PRSSB method. (iii) Image produced by the PRTS method.

The quality of the output images from PRTS2, PRTS3, PRSSB2, and PRSSB3 methods was measured by computing the MSE and SSIM. The MSE and SSIM were computed using the output images from PIE method in [14] where the system in equation (1) was solved by the LDL method as a reference. Tables 6-10 show the MSE and SSIM for the output images that are result from MVC, SICDII methods in [20, 22], and the two proposed methods with two and three pyramid levels.

Table 6. MSE and SSIM for output images from MVC and SICDII methods

	MSE				**SSIM**			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
MVC	0.0364	0.056	1.71	0.0462	0.9999	0.999	0.998	0.9999
SICDII	0.0223	0.681	1.14	0.0418	0.9997	0.999	0.998	0.9998

As shown in table 6, the MSE and SSIM for the images generated from MVC and SICDII methods in [20, 22] are approximately close to 0 and 1 for all instances, respectively. This means that there is a small difference between these images and the images produced by PIE method [14].

Table 7. MSE and SSIM for output images from PRTS2 method

	MSE				**SSIM**			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
PCG	0.263	0.118	6.69	0.797	0.999	0.999	0.997	0.9997
BCG	0.263	0.118	6.69	0.797	0.999	0.999	0.997	0.9997
SOR	0.293	0.114	6.69	0.797	0.999	0.999	0.997	0.9997
LU	0.263	0.118	6.69	0.797	0.999	0.999	0.997	0.9997
LDL	0.263	0.114	6.69	0.797	0.999	0.999	0.997	0.9997

Table 8. MSE and SSIM for output images from PRTS3 method

	MSE				**SSIM**			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
PCG	0.285	0.400	15.31	4.955	0.999	0.999	0.995	0.9988
BCG	0.285	0.400	15.31	4.955	0.999	0.999	0.995	0.9988
SOR	0.448	0.400	15.31	4.955	0.999	0.999	0.995	0.9988
LU	0.285	0.400	15.31	4.955	0.999	0.999	0.995	0.9988
LDL	0.285	0.400	15.31	4.955	0.999	0.999	0.995	0.9988

Table 9. MSE and SSIM for output images from PRSSB2 method

	MSE				**SSIM**			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
PCG	0.0937	0.074	4.43	0.653	0.9998	0.9999	0.998	0.9998
BCG	0.0937	0.074	4.33	0.653	0.9998	0.9999	0.998	0.9998
SOR	0.0937	0.074	4.56	0.6529	0.9998	0.9999	0.998	0.9998
LU	0.0937	0.074	4.43	0.653	0.9998	0.9999	0.998	0.9998
LDL	0.0937	0.074	4.43	0.653	0.9998	0.9999	0.998	0.9998

Table 10. MSE and SSIM for output images from PRSSB3 method

	MSE				**SSIM**			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
PCG	0.1600	0.177	10.61	4.955	0.9998	0.999	0.997	0.9988
BCG	0.1600	0.177	10.61	3.509	0.9998	0.999	0.997	0.9992
SOR	0.1600	0.177	10.61	3.509	0.9998	0.999	0.997	0.9992
LU	0.1600	0.177	10.61	3.509	0.9998	0.999	0.997	0.9992
LDL	0.1600	0.177	10.61	3.509	0.9998	0.999	0.997	0.9992

Results in tables 7-10 show that the MSE and SSIM values for the images generated by the two proposed methods with two and three pyramid levels are approximately near to 0 and 1, respectively. This means that these images are almost identical to the images produced by PIE method [14]. In the composition process for each instance, the linear system equation (1) is solved using direct and iterative methods. Iterative methods are adjusted to give approximately the exact solution for the linear system as direct methods. Thus, the MSE and SSIM values for each composite image when equation (1) is solved using direct and iterative methods are approximately the same.

The quality of the output images is measured visually by presenting a form to 30 people, then everyone gives a score between 0 and 10 (0 is the lowest score and 10 is the highest score) for each image for all methods. Finally, the average of the scores for each method is computed, as shown in table 11.

Table 11. A quality comparison between different methods

Method	Quality (%)
PIE [14]	65
MVC [20]	67
SICDII [22]	70
PRTS2	71
PRTS3	70
PRSSB2	68
PRSSB3	67

4.3 Time cost and Quality measurements

Results in subsections 4.1 and 4.2 showed that the composition time for the PRTS and PRSSB methods with two and three pyramid levels is less than the time taken from methods in [14, 20, 22]. Results also show that the quality of the output images from the two proposed methods is approximately the same as the quality of the images generated from methods in [14, 20, 22]. Tables 12, 13 show a run time, MSE, and SSIM comparison between PIE, MVC, SICDII methods in [14, 20, 22], and the two proposed methods.

Table 12. A run time comparison between different methods

Method	Balloon	Bird	Cow	Phalarope
PIE [14]	518.47	24.73	88.33	13.28
MVC[20]	183.49	15.61	42.38	9.44
SICDII[22]	100.96	9.1088	24.02	7.86
PRTS2	**56.06**	**7.03**	**13.73**	**3.61**
PRTS3	**38.51**	**7.14**	**16.18**	**4.1**
PRSSB2	**36.60**	**4.25**	**11.24**	**1.88**
PRSSB3	**12.24**	**2.56**	**4.51**	**1.86**

The highlighted values in table 12 show that the two proposed methods (PRTS2, PRTS3, PRSSB2, and PRSSB3) take the lowest composition time compared to the other three methods. The composition time for PRSSB2 and PRTS2 methods is lower than the time taken using the PIE method [14] by approximately a factor of 1/14 and 1/9, respectively. Results also show that PRSSB3 takes the lowest composition time for all images compared to other methods.

As shown in table 11, the quality of the output images from PRSSB2 and PRSSB3 methods is approximately the same as the quality of the output images from PIE, MVC, and SICDII methods [14, 20, 22]. Results in table 11 show that PRTS2 is the best method and the quality of the output images from PRTS3 method is approximately the same as the quality of the output images from MVC and SICDII methods [20, 22], and better than PIE method [14].

Table 13. MSE and SSIM comparison between different methods

	MSE				SSIM			
Method	**Balloon**	**Bird**	**Cow**	**Phalarope**	**Balloon**	**Bird**	**Cow**	**Phalarope**
MVC[20]	0.0364	0.056	1.71	0.0462	0.9999	0.999	0.998	0.9999
SICDII[22]	0.0223	0.681	1.14	0.0418	0.9997	0.999	0.998	0.9998
PRTS2	0.263	0.114	6.69	0.797	0.999	0.999	0.997	0.9997
PRTS3	0.285	0.400	15.31	4.955	0.999	0.999	0.995	0.9988
PRSSB2	0.0937	0.074	4.43	0.653	0.9998	0.9999	0.998	0.9998
PRSSB3	0.1600	0.177	10.61	3.509	0.9998	0.999	0.997	0.9992

Results in table 13 show that the quality of the output images from the two proposed methods are almost the same as the quality of the images generated by methods [20, 22].

5. Conclusions

In this paper, two efficient methods are proposed to reduce the time taken in composing color images. Using an image pyramid and divide-and-conquer methods, the proposed methods solve the Poisson equation for color images for each color separately. The proposed methods are based on partitioning the unknown region into small regions. Experimental results show that the composition time is reduced using the proposed methods. Results also show that the quality of the output images is approximately the same as the quality of the images that result from other methods. Thus, the proposed methods can be used efficiently in composing images.

References

[1] Hu, S. M., Chen, T., Xu, K., Cheng, M. M., and Martin, R. R., "Internet visual media processing: a survey with graphics and vision applications," The Visual Computer, 29(5), pp. 393-405, 2013. DOI: 10.1007/s00371-013-0792-6.

[2] Sun, J., Jia, J., Tang, C. K., and Shum, H. Y., "Poisson Matting," ACM Transactions on Graphics, 23(3), pp. 315-321, 2004. DOI: 10.1145/1015706.1015721.

[3] Levin, A., Lischinski, D., and Weiss, Y., "A closed-form solution to natural image matting," IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(2), pp. 228-242, 2008. DOI: 10.1109/TPAMI.2007.1177.

[4] He, K., Sun, J., and Tang, X., "Fast matting using large kernel matting laplacian matrices," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, pp. 2165-2172, 2010. DOI: 10.1109/CVPR.2010.5539896.

[5] Lee, P., and Wu, Y., "Nonlocal matting," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, pp. 2193-2200, 2011. DOI: 10.1109/CVPR.2011.5995665.

[6] Chen, Q., Li, D., and Tang, C. K., "KNN matting," IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(9), pp. 2175-2188, 2013. DOI: 10.1109/TPAMI.2013.18.

[7] He, B., Wang, G., Shi, C., Yin, X., Liu, B., and Lin, X., "Iterative transductive learning for alpha matting," 20th IEEE International Conference on Image Processing (ICIP), Australia, pp. 4282-4286, 2013. DOI: 10.1109/ICIP.2013.6738882.

[8] Shi, Y., Au, O. C., Pang, J., Tang, K., Sun, W., Zhang, H., and Jia, L., "Color clustering matting," IEEE International Conference on Multimedia and Expo (ICME), California, USA, pp. 1-6, 2013. DOI: 10.1109/ICME.2013.6607497.

[9] Gastal, E. S., and Oliveira, M. M., "Shared sampling for real time alpha matting," Computer Graphics Forum, 29(2), pp. 575–584, 2010. DOI: 10.1111/j.1467-8659.2009.01627.x.

[10] He, K., Rhemann, C., Rother, C., Tang, X., and Sun, J.,"A global sampling method for alpha matting," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, pp. 2049-2056, 2011. DOI: 10.1109/CVPR.2011.5995495.

[11] Shahrian, E., andRajan, D., "Weighted color and texture sample selection for image matting," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, pp. 718-725, 2012. DOI: 10.1109/CVPR.2012.6247741.

[12] Shahrian, E., Rajan, D., Price, B., and Cohen, S., "Improving image matting using comprehensive sampling sets," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, pp. 636-643, 2013. DOI: 10.1109/CVPR.2013.88.

[13] Varnousfaderani, E. S., and Rajan, D., "Weighted color and texture sample selection for image matting," IEEE Transactions on Image Processing, 22(11), pp. 4260-4270, 2013. DOI: 10.1109/TIP.2013.2271549.

[14] Pérez, P., Gangnet, M., and Blake, A., "Poisson image editing," ACM Transactions on Graphics (TOG), 22(3), pp. 313–318, 2003. DOI: 10.1145/882262.882269.

[15] Jia, J., Sun, J., Tang, C. K., and Shum, H. Y., "Drag-and-drop pasting," ACM Transactions on Graphics (TOG), 25(3), pp. 631-637, 2006. DOI: 10.1145/1141911.1141934.

[16] Qin, C., Wang, S., and Zhang, X., "Image editing without color inconsistency using modified Poisson equation," International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, pp. 397-401, 2008. DOI: 10.1109/IIH-MSP.2008.28.

[17] Dizdaroglu, B., and İkibaş, C., "A seamless image editing technique using color information," International Conference on Image Processing, Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 216-221, 2009.

[18] Dizdaroglu, B., and İkibaş, C., "An improved method for color image editing," EURASIP Journal on Advances in Signal Processing, Springer, 2011. DOI: 10.1186/1687-6180-2011-98.

[19] Tao, M. W., Johnson, M. K., and Paris, S., "Error-tolerant image compositing," International journal of computer vision, 103(2), pp. 178-189, 2013. DOI: 10.1007/s11263-012-0579-7.

[20] Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D., and Lischinski, D., "Coordinates for instant image cloning," ACM Transactions on Graphics (TOG), pp. 67, 2009. DOI: 10.1145/1531326.1531373.

[21] Farbman, Z., Fattal, R., and Lischinski, D., "Convolution pyramids," ACM Transactions on Graphics (TOG), 30(6), pp. 175, 2011. DOI: 10.1145/2070781.2024209.

[22] Jia, Y., and Wu, T., "Seamless instant image cloning based on derivative and intensity interpolation," Journal of Information and Computational Science, pp. 3019–3028, 2014. DOI: 10.12733/jics20103878.

[23] Facciolo, G., Sadek, R., Bugeau, A., and Caselles, V., "Temporally consistent gradient domain video editing," Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 59-73, 2011. DOI: 10.1007/978-3-642-23094-3_5.

[24] Chen, T., Zhu, J. Y., Shamir, A., and Hu, S. M., "Motion-aware gradient domain video composition," IEEE Transactions on Image Processing, 22(7), pp. 2532-2544, 2013. DOI: 10.1109/TIP.2013.2251642.

[25] Yan, T., Huang, Z., Lau, R. W., and Xu, Y., "Seamless stitching of stereo images for generating infinite panoramas," Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology, pp. 251-258, 2013. DOI: 10.1145/2503713.2503729.

[26] Ng, M. K., and Wang, W., "A Variational Approach for Image Stitching II: Using Image Gradients," SIAM Journal on Imaging Sciences, 6(3), pp. 1345-1366, 2013. DOI: 10.1137/120872140.

[27] Agarwala, A., "Efficient gradient-domain compositing using quadtrees," ACM Transactions on Graphics (TOG), 26(3), 2007. DOI: 10.1145/1276377.1276495.

[28] Finlayson, G. D., Drew, M. S., and Lu, C., "Entropy minimization for shadow removal," International Journal of Computer Vision, 85(1), pp.35-57, 2009. DOI: 10.1007/s11263-009-0243-z.

[29] Arias, P., Facciolo, G., Caselles, V., and Sapiro, G.,"A variational framework for exemplar-based image inpainting," International journal of computer vision, 93(3), pp. 319-347, 2011. DOI: 10.1007/s11263-010-0418-7.

[30] Doria, D., and Radke, R. J., "Filling large holes in lidar data by inpainting depth gradients," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, Rhode Island, USA, pp. 65-72, 2012. DOI: 10.1109/CVPRW.2012.6238916.

[31] Hareesh, A. S., and Chandrasekaran, V., "Exemplar-based color image inpainting: a fractional gradient function approach," Pattern Analysis and Applications, 17(2), pp. 389-399, 2014. DOI: 10.1007/s10044-012-0316-4.

[32] McCann, J., and Pollard, N. S., "Real-time gradient-domain painting," ACM Transactions on Graphics (TOG), 27(3), pp. 93, 2008. DOI: 10.1145/1360612.1360692.

[33] Liang, L., and Jin, L., "Image-Based Rendering for Ink Painting," IEEE International Conference on Systems, Man, and Cybernetics (SMC), Manchester, UK, pp. 3950-3954, 2013. DOI: 10.1109/SMC.2013.674.

[34] Hays, J., and Efros, A. A., "Scene completion using millions of photographs," Communications of the ACM, 51(10), pp. 87-94, 2008. DOI: 10.1145/1400181.1400202.

[35] Yang, J., Wang, Y., Wang, H., Hua, K., Wang, W., and Shen, J., "Automatic objects removal for scene completion," IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Canada, pp. 553-558, 2014. DOI: 10.1109/INFCOMW.2014.6849291.

[36] Bhat, P., Zitnick, C. L., Cohen, M., and Curless, B.,"Agradient-domain optimization framework for image and video filtering," ACM Transactions on Graphics (TOG), 29(2), pp.1-14, 2010. DOI: 10.1145/1731047.1731048.

[37] Xu, L., Lu, C., Xu, Y., and Jia, J., "Image smoothing via L0 gradient minimization," ACM Transactions on Graphics (TOG), 30(6), pp. 174:1– 174:12, 2011. DOI: 10.1145/2070781.2024208.

[38] Xu, L., Yan, Q., Xia, Y., and Jia, J., "Structure extraction from texture via relative total variation," ACM Transactions on Graphics (TOG), 31(6), pp. 139:1–139:10, 2012. DOI: 10.1145/2366145.2366158.

[39] Floater, M. S., "Mean value coordinates," Computer aided geometric design, 20(1), pp. 19-27, 2003. DOI: 10.1016/S0167-8396(02)00002-5.

[40] Shepard, D., "A two-dimensional interpolation function for irregularly-spaced data," In Proceedings 23rd ACM National Conference, New York, NY, USA, pp. 517–524, 1968. DOI: 10.1145/800186.810616.

[41] Rosenfeld, A. (Ed.)," Multiresolution image processing and analysis," Springer-Verlag, Vol. 12, 1984. DOI: 10.1007/978-3-642-51590-3.

[42] Vismor, T., "Matrix algorithms," Available on the internet https://vismor.com/download/Documents/Network_Analysis/matrix_algorithms.pdf, 2015.

[43] http:// www.public-domain-image.com (last accessed: February 2015).

[44] Davis, T. A., Direct methods for sparse linear systems, Siam, 2006. DOI: 10.1137/1.9780898718881.

[45] Saad, Y., Iterative methods for sparse linear systems, Siam, 2003. DOI: 10.1137/1.9780898718003.

[46] Wang, Z., and Bovik, A. C., "A universal image quality index, "IEEE Signal Processing Letters, 9(3), pp. 81-84, 2002. DOI: 10.1109/97.995823.

[47] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity, "IEEE Transactions on Image Processing, 13(4), pp. 600-612, 2004. DOI: 10.1109/TIP.2003.819861.