International Journal of Computing Academic Research (IJCAR)
ISSN 2305-9184, Volume 7, Number 5 (October 2018), pp.62-67

http://www.meacse.org/ijcar

Android Based Photomontage Application

Dhia DAMKHI[1] and Mouslem DAMKHI[2]
[1]Mathematics Stream, High School of Abbes Laghrour, Batna, Algeria
[2]Department of Computer Engineering, Karadeniz Technical University, Trabzon, Turkey

Abstract

This paper presents two ways to achieve an Android based photomontage. The first one was realizing an Android application which works based on an Android device only. While, the second one was realizing an application which uses a server in order to optimize the required time for the photomontage process. This process is done through solving a Poisson equation using iterative methods. First, a general idea of the proposed approach was given. Secondly, an approach considering the iterative methods for solving the Poisson equation based on the Dirichlet boundary conditions was presented. Later, a general idea about the Android only based application was given followed by a presentation of its test's results. Finally, the client-server based application was presented followed by a discussion about its test's results compared to those of the previouse case.

Keywords: Poisson Equation, Iterative Methods, Photomontage, Distributed Systems, Network Programming.

Introduction

A photomontage is a photographs assembling by sticking, printing or using a software to give a photo a different appearance, by incorporating one or more parts or the totality of another photo and allowing any retouching and special effects. A number of approaches have been developed in order to allow the merging of the digital photographs without leaving noticed traces. However, these approaches are usually implemented as computer based applications. In that case, the sizes of the computer devices limit the places and the times of the use of these applications. On the other hand, the availability and the wide spreading of the Android based devices can be helpful to widely exploit the developed image processing approaches through implementing Android based applications. Yet, the performance limitation of the Android based devices prevents them from solving big and complex problems, which leads to the use of a server with a higher level of performance. After the server receives the related data of the photomontage process from the client (the Android based device), it achieves the necessary operations and sends back the final results to the client.

Other than the classical approaches, the new approaches of photomontage are mainly mathematical approaches where they generally base on the Poisson equation [1]. This equation actually behaves similarly to the Laplace equation. The Laplace equation blurs the image because it behaves like a low-pass filter. However, in order to overcome this obstacle, the Poisson equation approaches take into account additional data of the image section to be adjusted. Moreover, several methods have been developed in the literature for the solution of the Poisson equation. Although iterative methods such as SOR (Sequential Over-Relaxation) [2] produce a slightly slower result than direct methods, they are preferred because additional libraries are not needed in the solution of the linear equation sets while the computer program is being written. In addition, the Poisson equation can also be solved based on the boundary conditions of both Dirichlet and Neumann [3].

In this study, two Android based application are presented to make photomontage widely provided. The first application achieves the necessary calculation for the photomontage process locally, while the second uses a server to get the results. Both applications uses iterative methods to solve the linear equations in order to accelerate the process.

The Used Approach

Poisson photomontage is a technique for merging two images together in a fully automatic way. The mathematical expression behind this method is the following [1]:

$$min_{f \in \Omega} \int |\nabla f - \nabla g|^2 \, dx \, dy, \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{1}$$

In Eq. 1, g is the source image containing the object to be selected, f presents the resulted iamge after the photomontage process, and f * denotes the target image to be photomontaged. Also, the boundary of the selected region is denoted by $\partial\Omega$. Besides, ∇ is the gradient operator and is expressed as:

$$\nabla . = \left(\frac{\partial .}{\partial x}, \frac{\partial .}{\partial y}\right)^T \tag{2}$$

Several methods have been developed in the literature for solving Eq. 1 depending on the Dirichlet boundary conditions. In this study, iterative methods (Jacobi, Gauss-Seidel and SOR) have been taken into consideration [2]. According to the method of Jacobi, the following expression can be written for the solution of Eq. 1:

$$f(x,y)^{(k+1)} = f(x\text{-}1,y)^{(k)} + f(x,y\text{-}1)^{(k)} - 4f(x,y)^{(k)} + f(x+1,y)^{(k)} + f(x,y+1)^{(k)} - \Delta g(x,y) \tag{3}$$

In Eq. 3, k denotes the number of iterations, Δ is the Laplace operator. The following expression can be used for Δg:

$$\Delta g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \tag{4}$$

Assuming that the edited picture is scanned from top to bottom and from left to right, Eq. 1 can be rearranged as follows considering the previous iteration and the current iteration:

$$f(x,y)^{(k+1)} = f(x\text{-}1,y)^{(k+1)} + f(x,y\text{-}1)^{(k+1)} - 4f(x,y)^{(k)} + f(x+1,y)^{(k)} + f(x,y+1)^{(k)} - \Delta g(x,y) \tag{5}$$

Eq. 5 is called the Gauss-Seidel method. Note that since the values in that iteration are also used in the equation, compared to the Jacobi method, the result is faster. On the other hand, in the SOR method, a weighting factor ω is added as shown in the following equation to produce the result in less time than in the first two iterative methods:

$$f(x,y)^{(k+1)} = f(x,y)^{(k+1)} + \frac{\omega}{4}[f(x\text{-}1,y)^{(k+1)} + f(x,y\text{-}1)^{(k+1)} - 4f(x,y)^{(k)} + f(x+1,y)^{(k)} + f(x,y+1)^{(k)} - \Delta g(x,y)] \tag{5}$$

$$\omega = \frac{2}{1+sin\left(\frac{2\pi}{x_{max}-x_{min}+1}\right)} \tag{6}$$

Android Device Only Based Application

In this application, the process of photomontage is done according to the Poisson equation approach which was presented in the previous section, where the iterative methods are taken into account during its solving. The process steps of the application can be summarized as follows:

i. Load the source image and roughly select the section to be photomontaged,

ii. Load the target image, paste the selected section in the target image, then drag and drop it in the desired position,
iii. Select an iterative method to solve the Poisson equation and run it,
iv. Terminate the iterations based on a specified error threshold value,
v. Show and save the resulted image.

Each of the steps i, ii, iii and v are show in Fig. 1. The used images as source [4] and target [5] images are licensed under CC0 License.

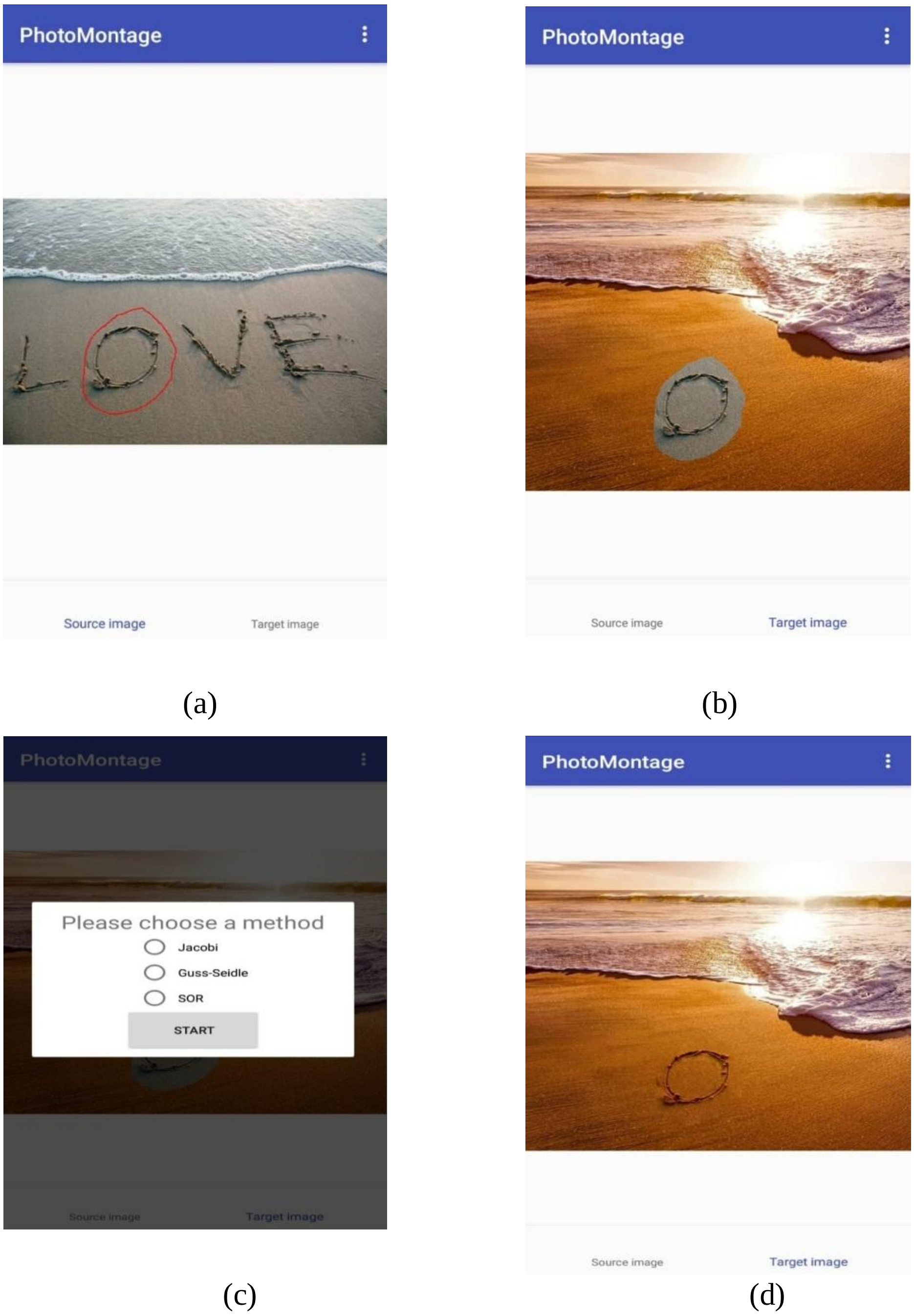


Fig. 1. Photomontage process using Android device only based application: (a) step i, (b) step ii, (c) step iii, (d) step v.

The evaluation of the application, both in terms of the quality of the resulted image and the execution time has been made. In the iterative methods, $\varepsilon = 1$ is taken to terminate the iteration and zero values are assigned to the unknowns as elementary values.

A SAMSUNG Galaxy J7 SM-J700H smart phone which works based on Android 6.0.1 operating system was used during the test. Furthermore, the used smart phone contains an Exynos 7580 1.5 GHz CPU and 1.5GB of RAM memory. Moreover, The total number of pixels of the selected region from the source image is 18456 pixels. As can be seen from Fig. 1 (d), the photomontage process has been successfully accomplished. As a result, the processing times for the iterative methods are given in Table 1 in seconds. As shown in Table 1, the SOR method produces approximately 90 times faster than the Jacobi method and 32 times faster than the Gauss-Seidel method.

Table 1. The necessary time to get the result in Fig. 1 (d) using the Android device only based application.

Method	Execution Time (s)
Jacobi	6866
Gauss-Seidel	2430
SOR	76

Despite the effectiveness of SOR method compared to the other methods, it is still not fast enough especially when the number of pixels of the selected area increases. In order to optimize the execution time of the application, we propose that the process related to the calculations will not be done locally by the Android device, but instead, the device will be connected to a higher performance server which takes care of the calculations part.

Client-Server Based Application

The main innovation in this case is to take into account a client-server based approach to achieve the photomontage process. As in the case of distributed system approaches, in order to remove problems such memory saturation, CPU warming or long time processing, the client sends the process related data to a server with higher performances which achieves the process and sends the final results to the client. In order to realize the client-server architecture, we adopted the Java socket client-server model [6].

The connection between the client and the server is done through a socket which is a logical communication channel that opens between two parties in a network in order to exchange data between them. Opening that channel requires a port definition for the server and the client [7]. Fig. 2 shows how the communication is done between our proposed application and the server.

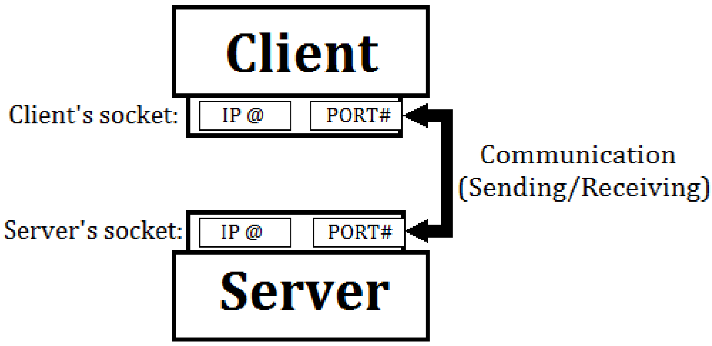


Fig. 2. The client-server communication through sockets.

As in the previous case, the process of making the photomontage is done according to the Poisson equation approach. The process steps of the application are as follows:

i. Load the source image and roughly select the section to be photomontaged (client side),
ii. Load the target image, paste the selected section in the target image, then drag and drop it in the desired position (client side),
iii. Select an iterative method to solve the Poisson equation, (client side),
iv. Send the related data of the solving process to the server (client side),

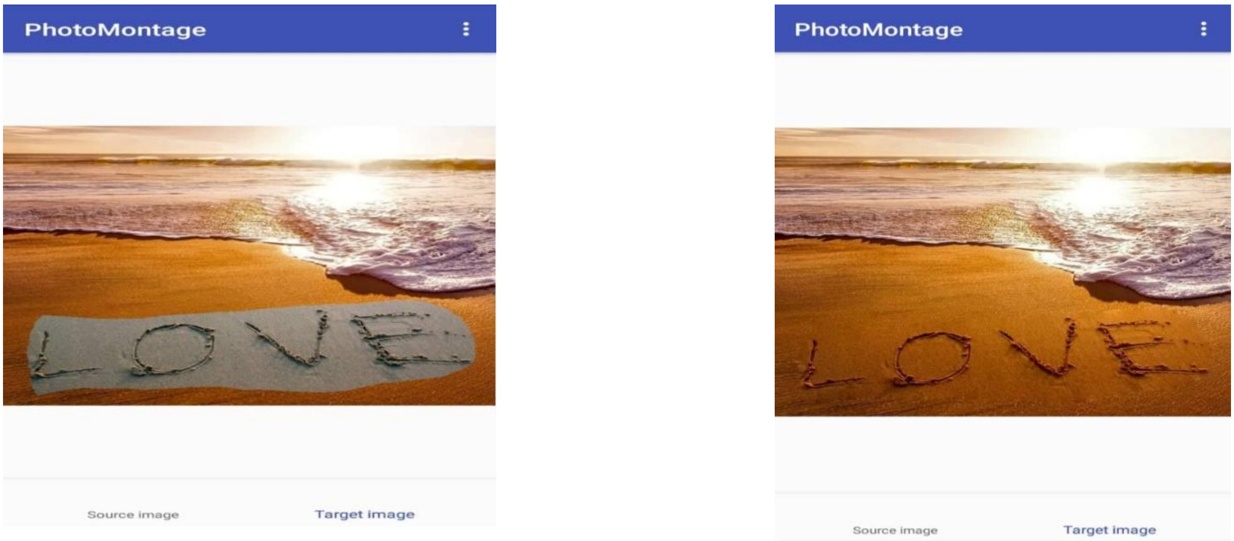
v. Receive the related data of the solving process from the client (server side),
vi. Run the selected iterative method to solve the Poisson equation (server side),
vii. Terminate the iterations based on a specified error threshold value (server side),
viii. Send the resulted image to the client (server side),
ix. Show and save the resulted image (client side).

The same example of the previous case is tested, where the value $\varepsilon = 1$ is taken into account to terminate the iterations of the selected iterative method and also zero values are assigned to the unknowns as elementary values. In addtion to the same smart phone that has been used previously, a machine with 8 GB of RAM and an Intel Core i7-2670QM 2.20GHz CPU and works based on Microsoft Windows 7 Professional operating system is used to run the server. The processing times for the iterative methods in this case are given in Table 2 in seconds. As shown in Table 2, the execution time is decreased by about 97% to 99%.

Table 2. The necessary time to get the result in Fig. 1 (d) using the client-server based application.

Method	Execution Time (s)
Jacobi	106
Gauss-Seidel	17
SOR	2

In order to check the effectiveness of the client-server application, a bigger area with 113059 pixels is selected from the source image and pasted in the target. The result of this test is shown in Fig. 3. The processing times for the iterative methods are given in Table 3 in seconds. As shown in Table 3, the SOR method produces approximately 120 times faster than the Jacobi method and 12 times faster than the Gauss-Seidel method. Besides, the execution time of the SOR method proved that it is still can be useful when the number of pixels gets higher dispute the other methods.



(a) (b)

Fig. 3. Photomontage process using client-server based application: (a) the selected section is pasted in the target image, (b) the resulted image.

Table 3. The necessary time to get the result in Fig. 3 (b) using the client-server based application.

Method	Execution Time (s)
Jacobi	2874
Gauss-Seidel	305
SOR	24

Conclusion

In this paper, a study has been made on how to make photomontage effectively using android based applications that use the Poisson equation. Two applications have been proposed where the first one achieves the necessary calculations for the photomontage process locally, while the second connects to a server which takes care of the calculations. In addition, each of the developed applications solves the Poisson equation iteratively using either the method of Jacobi, Gauss-Seidel or SOR. Also, both of the developed applications have been tested while the client-server based application using the SOR method proved its effectiveness. As future works, other image processing approaches such as images segmentation and images coloring can be realized to work based on Android operating system (Android device only and client-server applications).

References

[1] P. Perez, M. Gangnet and A. Blake, "Poisson Image Editing", ACM Transactions on Graphics, vol. 22(3), pp. 313-318, 2003.

[2] C. H. Rycroft, "Iterative Methods for Linear Systems", School of Engineering and Applied Sciences, Harvard University, Cambridge, 2007.

[3] S. Salleh, A. Y. Zomaya, S. Abu Bakar, Computing For Numerical Methods Using Visual C++, 1st ed, New Jersey, John Wiley & Sons, Inc., 2007

[4] (2018) PEXELS website. [Online]. Available: https://www.pexels.com/photo/beach-coast-island-landscape-348520/

[5] (2018) The PEXELS website. [Online]. Available: https://www.pexels.com/photo/seashore-under-white-and-blue-sky-during-sunset-210205/

[6] The ORACLE website. [Online]. Available: https://docs.oracle.com/javase/tutorial/networking/sockets/index.html

[7] G. Coulouris, J. Dollimore, T. Kindberg and G. Blair, Distributed Systems: Concepts and Design, 5th ed., Boston, Addison-Wesley, 2012.