2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) | 978-1-6654-3521-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSES52305.2021.9633858

Comparative Study of Gradient Domain Based Image Blending Approaches

Maturi Tanuj[1], Aishwarya Virigineni[1], Apoorva Mani[1] and Subramani R[2]

[1]Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India
[2]Department of Mathematics, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India
charantanujma63@gmail.com1, aishwaryavirigineni@gmail.com1, apoorva.blue73@gmail.com1, r_subramani@blr.amrita.edu

Abstract :- The paper describes a comparative study of three different approaches that are used for image blending. The main focus will remain on the approaches where single source image and target is composed in the gradient domain. The main aim of the study is to portray the importance of considering gradients in image blending and why it makes the blending more realistic and effective. The study of all the approaches that are taken under consideration have been executed using MATLAB. The three different approaches are Naive blending, poisson image blending and Mixed gradient approach of image blending.

Keywords—Gradient Domain, PoissonBlending, Mixed Gradient, Sparse, Naïve Blending, Seamless Cloning,

I. INTRODUCTION

With many upcoming technologies and tools, Image Editing and Manipulation of image gradients has taken a turn to become more and more efficient. Image editing can be broadly classified based on global changes and local changes. Here Global changes concentrates on correcting the intensity, adding filters, reshaping of the image whereas Local changes concentrates more on achieving goals in a particular area of the image in a seamless method [1,2]. In this project we'll be concentrating on the local changes. Our Human eyes are very sensitive to lighting differences and colours within an image that's why image gradients play a very important role in manipulation of the image[3, 4].We should also keenly note that playing with these image gradients is what makes the project all the more interesting to achieve efficient results. Some examples of the basic operations that we'll come across while discussing Local changes are local illumination changes, seamless tiling, seamless cloning, texture flattening etc [5, 6].

Image blending has proved to be a very useful application when it comes to enhancing the quality of an image by removing unwanted objects in the background, healing damaged places if the image and also Panorama stitching.Several upcoming image blending techniques have come to use these days. The Adobe Photoshop uses alpha compositing for blending 2 images. We also have other approaches that follow a multi-scale portrayal for blending certain regions of one image into another. Patch based synthesis is an approach where similar patches in images are searched in order to generate a natural composited image[2]. Whereas in Gradient domain is used in most of the techniques for blending images. This paper presents a comparative study of 3 different techniques used for image blending. The 3 techniques that will be taken into study are: (i)Naive Blending , (ii)Poisson Blending, (iii)Mixed Gradient approach. Multiple test cases have been used in order to analyse the 3 techniques effectively.

II.POISSON BLEND

Let's look at the given illustration and try to understand the basic working of Image blending. Here S represents a subset(closed) of R2 which intern is the image domain. We have a source image, target image and a mask region from which I want to take the source pixels of region of interest through mask and put them into the target image[7, 8]. Here we specify a region called Omega. Omega is basically the pixels that we want to take from the source. Omega further is a closed subset of S. And then we are going to pay a particular attention to the boundary of the mask region called D Omega or del Omega. Here we'll be considering 2 functions f and f*. function f denotes an unknown scalar function that is defined over the closed subset of S i.e.; Omega. whereas function f* denotes a known function over S minus, in other words the interior of Omega. Figure1 illustrates the notations [9, 10].

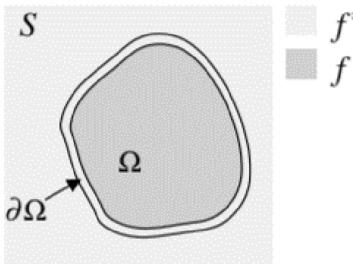

Fig. 1 Guided Interpolation Notation

So, the idea here is in order to reduce the colour mismatch between the source and the target, we create the composite in the gradient domain[11]. We want the gradient of the composite inside the transition region Omega to look asclose as possible to the source image gradients. Also, the composite $I_{(x,y)}$ must match the target image $T_{(x,y)}$on the boundary of the region. So mathematically, we want to minimize all of the pixels of the composite inside the compositing region. We are going to integrate over the area i.e.; the difference between the gradient of the composite $\nabla I_{(x,y)}$ which is what we are generating and the gradient of the source$\nabla S_{(x,y)}$. We will need to make it as small as possible subject to the constraint thatthe image that we create has to agree with the target colours right around the boundary of the region[12].

$$\min_{I_{(x,y)} \in \Omega} \iint_{\Omega} ||\nabla I_{(x,y)} - \nabla S_{(x,y)}||^2 \, dxdy \qquad (1)$$

$$\text{such that} \quad I_{(x,y)} = T_{(x,y)} \text{ on } \partial\Omega \qquad (2)$$

The solution to this problem can be given making the Laplacian of the composite equal to the Laplacian of the source inside this region and they need to make the composite equal to the target on the outside of the region[3]. This is basically our Poisson equation.

$$\nabla^2 I_{(x,y)} = \nabla^2 S_{(x,y)} \text{in} \Omega \qquad (3)$$

$$\text{such that} I_{(x,y)} = T_{(x,y)} \text{on } \partial\Omega \qquad (4)$$

Matrix Generation

The image blending problem is expressed as a least-squares problem, wherein we need to solve AX = B for each and every pixel beneath the mask. We have a mask image of size equal to the size of the source and the target. The ideal values of mask are either 0 or 1. In our ultimate image, if 0 is our mask value, we will be simply taking the pixel value of the target in that moment. If it is 1, we would like to put in place a linear equation in such a manner that the gradient for a specific pixel is identical to the one in both the source and target images.

We get through the entire image and generate a vector along with an entry for each pixel that takes into account the real values of either one the target image, provided that the mask was 0 at that point, or either in the gradient value from the source provided that the mask turned out to be 1 at that point[5]. We will be able to find out our output image when we find the coefficient matrices A and B.B is a (Nx1) matrix which shows the required output gradient and values, and A is a huge, sparse matrix which is described as (NxN), represents unidentified gradient values. A is identical in all the three coloured channels R, G, B, hence we create only once, but we calculate three different B column vectors: one for every colour channel. X comprises the values of all the output pixels. The value of N represents the number of pixels that remain under the mask as well as those maintain 1 value.

Mixed Gradient

Pixel resolution that was not in value 1, like some spots that were in accordance with the mask, then We had given the accurate value (4 if not the border). But the problem occurs when the pixel remains on the borders which could not coincide with the gradients that have existed in a source image. To deal with this problem, the Laplacian is being used on every pixel in a source image: 4(Sourcepixel) – (Top pixel) - (Bottom pixel) - (Right Pixel) - (Left Pixel)

Track the same step just like the Poisson blending but then again utilize the gradient into the source or target along with the larger magnitude, sooner than the source gradient. Single possibility of blending when you use the mixed gradient is to mix an image of writing on a simple background to another picture.

III. Implementation

Input has 3 components: source image, target image and a mask image will be created according to the user preference. Here, mask gives information about the regionin the target image where the cropped area of source image has to be pasted.Mask image represents the binary image where 0 refers to the aim background should be preserved, while a 1 refers to the source object need to be replicated over.

Source, target and mask images are kept in same size during the blending process. source image will be cropped and mask can will be created and that will be fitted on the target image. Matrix manipulations algorithm requires the source ,target and mask images in same size to implement algorithm[7].

Matrix manipulations algorithm involves resolving the structure of matrix equations.as each pixel has an unknown in the result image, leading to large matrices[6].

We take sparse matrix A,B here A is height*width × height*width. That makes sure each and every pixel will have a separate column in the system of equations. Matrix, B, is a column vector of height height*width.

The rule followed in order to fill the matrix: By Iterating through each pixel of the mask, it is determined whether that pixel should be placed at as the target pixel or not. If the value of the pixel is 0 then no change will be made in that particular position in the target image whereas is the pixel value is 1 then , the gradient of the source image must be calculated and replaced in the target image.

The gradient is determined by using discrete Laplacian by considering at most of 4 neighbours. Whenever we have to compute the gradient of the source, a matrix called A for that pixel get a value of 4 and all the neighbouring pixels such as left ,right, above and below of the source image get a value of -1 in the matrix[10]. The value b at that particular pixel is calculated according to the given formula:

$$4 * s(i,j) - s(i-1,j) - s(i+1,j) - s(i,j-1) - s(i,j+1)$$

here, s(i,j) signifies source image, and (i,j) refers to the columns and rows of the image.

When A and b will be filled, we will start solving the system of equations. To put up an equation, we take Ax = b here x is considered as output column vector. When the system has been resolved, x will be easily reformed to the normal dimensions of the given target image[12].

According to our implementation, we'll be taking two images that is our source and target images. A mask is created for region of interest in the source image[9]. We'll be having another window popup which represents our target image . We'll have to adjust the position of bounding box of the mask on target image and double click in the desired location. This will be giving us 3 kinds of outputs. One being that of Naive Blending and the another one is Mixed Gradient Blending. Naive Blending output is expected to portray a mere copy paste of source image on target image. Mixed Gradient blending is expected to give a better seamless blend than Possion blend.

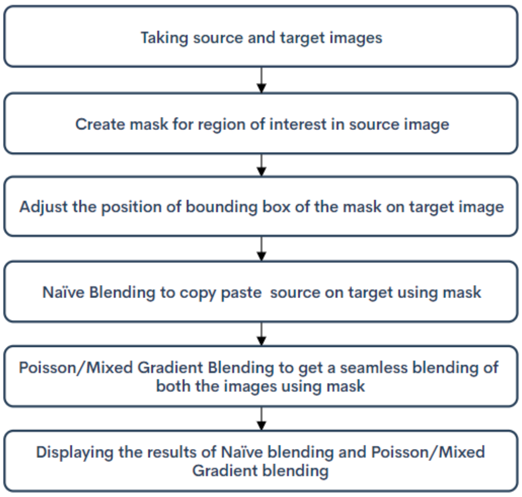

Fig. 2 Working Model

IV. Results

The main aim of this study was to compare and contrast among the 3 approaches of image blending that have been taken into consideration i.e.; Naive Blending, Poisson blending and mixed gradient approach. In order to make it more effective we have taken 2 types of test cases: (i) Target image and Source image having background of different gradient . (ii) Target image and source image having background of similar gradient.

Naïve Blended Output

As we can see in Figures 5 and 10 irrespective of the source and target images belonging to similar or different gradient background, the resultant image looks like a mere copy pasting of source image on target image.

Poisson Blending Output

From Poisson blending results Figure 6 and Figure 11 we can observe that the algorithm performs well for images part of backgrounds with similar gradients. We can see how seamlessly the images have blended in case(ii). It can be inferred that Poisson blending performs better than Naive blending, but it should also be kept in mind that it has a limitation of working under only similar gradient background condition.

Mixed Gradient Output

This algorithm seems to have outperformed the previous 2 approaches as it can be seen in Figure 7 and Figure 12. This approach has provided desirable results for both the cases.

CASE I:

In this case, we have considered two images which have different gradient backgrounds.

Here, Figure 3 is the source image and Figure 4 is the target image.

Fig. 3 Source Image of Case1

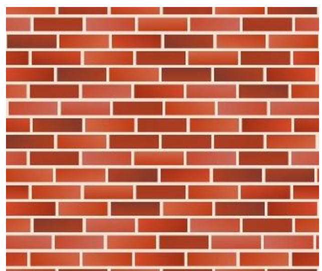

Fig. 4 Target Image of Case1

Figure 5 is the result ofNaive Image Blending

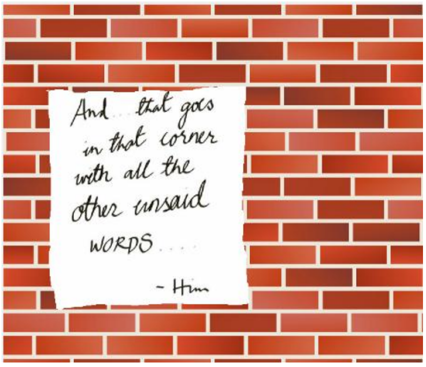

Fig. 5 Naïve Blended Result of Case1

Figure 6 is the result of Poisson Image Blending

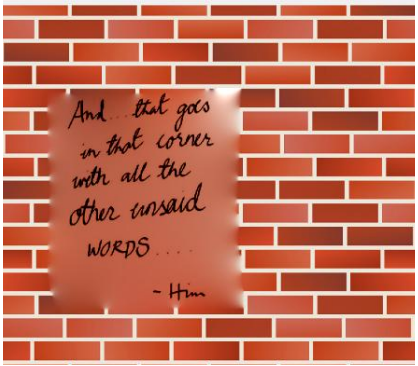

Fig. 6 Poisson Blended Result of Case1

Figure 7 is the result of Mixed Gradient Image Blending

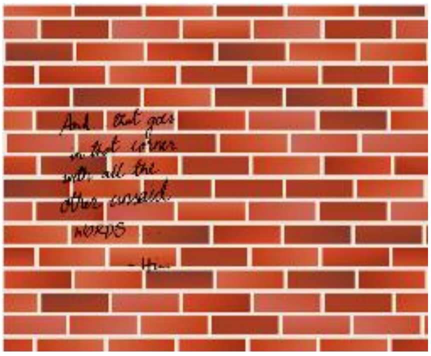
Fig. 7 Mixed Gradient Result of Case1

CASE II:

In this case, we have considered two images which have similar gradient backgrounds. Here, Figure 8 is the source image and Figure 9 is the target image.


Fig. 8 Source Image of Case 2


Fig. 9 Target Image of case2

Figure 10is the result of Naive Image Blending


Fig. 10. Naïve Blended Result of Case2

Figure 11 is the result of Poisson Image Blending


Fig. 11. Poisson Blended Result of Case2

Figure 12 is the result of Mixed Gradient Image Blending

Fig.12 Mixed Gradient Result of case2

Tabular Representation of Results

NAME OF THE APPROACH	TEST CASE 1	TEST CASE 2
NAIVE BLEND	FAILED	FAILED
POISSON BLEND	SATISFACTORY	NOT SATISFACTORY
MIXED GRADIENT	SATISFACTORY	SATISFATORY

V. CONCLUSION

This project has given us a clearcut idea about both the strengths and the shortcomings of each approach that was taken under study. Naive blending has come out to display a mere copy paste method of combining two images without actually blending them together, so it can explicitly be made out that the output image is not a unique image but is a combination of multiple images overlapped. Poisson blending works excellent on blending images that belong to having background of similar gradients. Mixed Gradient approach has been seen to have outperformed both Naive and poisson blending when it comes to all the test cases that we took under consideration be it images that belong to similar gradients background or not. One keen observation that can be made while differentiating mixed gradient approach from poisson blending is that in mixed gradient the gradient of the image with larger magnitude is used. Whereas in poisson blending , the gradient of the source image is used as default for blending.

REFERENCES

[1] Tanaka, Masayuki, Ryo Kamio, and Masatoshi Okutomi. ”Seamless image cloning by a closed form solution of a modified poisson problem.” SIGGRAPH Asia 2012 Posters. ACM, 2012.

[2] Bhaswitha Maddula, Manikantha Vallabhaneni, Sarada Jayan and Subramani R (2020) Chaotic Evolutionary Algorithm, 2020 IEEE International Conference for Innovation in Technology (INOCON), 1 – 4, DOI: 10.1109/INOCON50539.2020.9298233

[3] Burt, Peter, and Edward Adelson. ”The Laplacian pyramid as a compact image code.” IEEE Transactions on communications 31.4 (1983): 532-540.

[4] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In Conf. on Computer Vision and Pattern Recognition, pages II:509516, 2001.

[5] S. Tamuly, Jyotsna C, and Amudha, J., “Deep Learning Model for Image Classification”, Advances in Intelligent Systems and Computing, vol. 1108. Springer International Publishing, Cham, 2019.

[6] Prez, Patrick, Michel Gangnet, and Andrew Blake. ”Poisson image editing.” ACM Transactions on Graphics (TOG). Vol. 22. No. 3. ACM, 2003.

[7] Raguvir S, Radha D(2019) Analysis of Explicit Parallelism of Image Preprocessing Algorithms - A Case Study. Proceedings of the International Conference on ISMAC in Computational Vision and Bio- Engineering 2018 (ISMAC-CVB). Lecture Notes in Computational Vision and Biomechanics, vol. 30: 1499 – 1508.

[8] Manikantha Vallabhaneni, Bhaswitha Maddula, Sarada Jayan, Subramani R (2020) Chaotic Hooke-Jeeves Algorithm using Cubic map with MATLAB code, 2020 IEEE International Conference for Innovation in Technology(INOCON), 1 – 4, DOI: 10.1109/INOCON50539.2020.9298244

[9] Frankot, Robert T., and Rama Chellappa. ”A method for enforcing integrability in shape from shading algorithms.” IEEE Transactions on pattern analysis and machine intelligence 10.4 (1988): 439-451.

[10] Mohammed Afifi and Khaled F. Hussian,(2015), “A modified poisson image blending”, Computational visual media, vol. 1, No. 4: 331 – 341.

[11] O. K. Sikha, Sachin Kumar S., and Soman K. P., “Salient Region Detection and Object Segmentation in Color Images using Dynamic Mode Decomposition”, Journal of Computational Science, 2017.

[12] O. K. Sikha, Soman, K. P., and Sachin Kumar S., “VMD-DMD coupled data-driven approach for visual saliency in noisy images”, 2019.