Implementing and Analysis of the Exemplar-Based Image Inpainting Method

J. Serdoncillo
EE 5561: Image Processing and Applications
University of Minnesota - Twin Cities
Minneapolis, United States
serdo004@umn.edu

***Abstract*—The first mini-project of the semester composes the implementation of a research paper made by Perez, and Toyama (2004) titled "Region Filling and Object Removal by Exemplar-Based Image Inpainting" which is a continuation of an earlier project. Various different object removal techniques have been used in the past but most of them result in blurred patches or obvious seams. This breaks the illusion and usually makes the image look worse. During the present time, most workers will manually do the object removal and region filling through the use of photoshop or complex models of artificial intelligence. This takes a considerable amount of time and effort. In the mentioned paper, their method highlights order of patching and confidence of pixel values algorithms. This creates an effective and fast image inpainting process which is superior to different methods. Or as they claim. The method was re-implemented using python and through using different real and synthetic images. It was seen that the success the method claims is proven to be a fact.**

***Index Terms*—Object Removal, Image Inpainting, Implementation, Patch-based method**

I. Introduction

A. Inpainting Techniques

Image inpainting is traditionally used for removing large objects from an image and to fill up that area with a texture similar to the background. Ideally, the replaced image should unseen to the human eye for the modified target region. This paper is an implementation of an image inpainting technique using a patch-based method while also utilizing a priority system in order to propagate edges properly. There are a variety of algorithms and papers to implement image inpainting. Some use diffusion based models which can introduce a blur [9]. Others split the image into structure and structure which can take a long computation time [7]. The closest other technique is through the use of comparison of a target pixel to surrounding pixel which is promising but is susceptible to noise [13]. In the other aspect, some also tried to do a "priority system" which changes the filling order of the target image. Overall, it can be seen that other algorithms and techniques have lots of flaws compared to the Criminisi algorithm. There are similar techniques which uses patch-based methods for quick time [8]. The Criminisi algorithm has been shown to provide substantial results for a fraction of computational time and cost.

(a) (b)

Fig. 1. Removing unwanted objects from images. (a) Original photograph containing a butterfly and a random stick. (b) A mask was created to highlight the area of the stick and then removed using Exemplar inpainting.

B. Current Techniques

One thing to note is that this paper was published almost 2 decades ago. In that span of time, there are now new methods in order to fill up a target image or even expand a current image and hallucinate new details. A big notable change is through the use of human labor in order to manually and more specifically modify an image to produce an output that would be good enough for the human eye. Most of the time, these outputs are usually judged subjectively based on an individual's perspective. Quantifying this parameter can be hard or have some drawbacks to some factors which makes it hard for a computer to optimize an objective function. Usually the best method is to then judge it with pure human eyes.

In addition to photoshop, a famous trend has been popping up as well which are large diffusion models such as Dall-E 2 [14]. Aside from just filling up target regions worthy to most people, these are also now used in order to produce new images based on a text prompt. Their success is due to an outstanding amount of training data it has been given which is trained using machine learning and artificial intelligence methods. Although pretty great for entertainment, most people don't have other big uses for it.

Going back to the academic world, research related to image inpainting are still thriving. Image inpainting is great for digital reconstruction of damaged old images [4], image completion [3], and digital restoration of famous art [5]. The fundamental concepts for their implementation are still grounded on statistics and math. New iconic methods like large diffusion models are now mainly made through artificial intelligence.

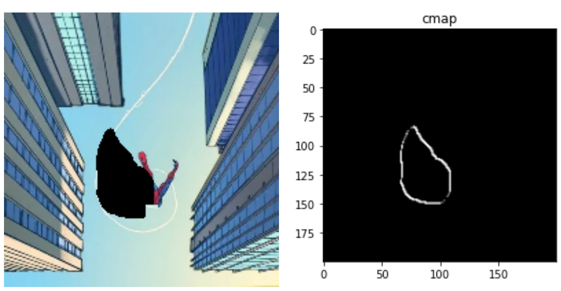
II. Methods

The implementation of the Criminsi algorithm can be summarized into three steps, namely: *Computing Patch Priorities*, *Propagating Texture and Structure Information*, and *Updating confidence values*. How it was implemented can be described more in the following sections.

A. Contour Detection

In order to choose what patch to focus on, the contours of the target image is needed first and their corresponding pixel values. Usually, this can be done by using a Sobel filter in order to get sharp gradients in the x and y-direction and by using a convolution on an image. To make the implementation more uniquely different, a kernel is designed using a central differencing kernel for x and y which when used should produce roughly the same result.

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \tag{1}$$



(a) (b)

Fig. 2. Contour detection is needed to do first before computing patch priorities. (a) shows the original image with an erased area which to be inpainted. By using the kernel designed above and using a convolution on a boolean confidence mask we get the following contour (b). This gives out all of the possible pixel values to calculate their priorities.

B. Priority Calculation

Based on the paper, it was observed that the filling order is critical in patch-based methods. The quality of the output image synthesis is improved if some patches are done first based on the confidence of what it should be and if there are any strong structures to propagate. The priority is calculated with the equation below as a product of a confidence value and a data value.

$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p}) \tag{2}$$

The confidence value of a patch is based on the the mean confidence of the given patch. The data value is the based on the projection of an image gradient with the patch normal vector. More details about the implementation can be referred to succeeding sections or the original paper.

$$C(\mathbf{p}) = \frac{\sum C(\mathbf{q})}{n * n} \quad \text{and} \quad D(\mathbf{p}) = \frac{|I_p \cdot \mathbf{n}_p|}{255} \tag{3}$$

C. Source Patch Search

After a target patch has been chosen, a source patch will now be found which will be used as a source copy. This will be based on the similarity of a source patch with known pixel values in the target patch. The parameter to be compared is the sum of squared differences while the pixel values are represented in CIE Lab color space.

$$dist(\mathbf{p}, \mathbf{q}) = \sum_{pixel} (\text{LabCIE } \mathbf{p} - \text{LabCIE } \mathbf{q})^2 \tag{4}$$

III. Results

A. Filling Process

After implementing the methods above, it was seen that now the image is patched not in a top-bottom left-right matter but instead is based on the priority value of the patch at each iteration. More relevant representations can be seen in the original paper and would actually be best seen as a form of a video during the patching process.

B. Basic Images

In order to test the success of the implementation of the exemplar method, it is first tested on basic images. The most basic image possible is through the use of singular edge greyscaled image as seen in the figure below. A basic target mask is a square in a region of the discontinuity.

(a) (b) (c)

Fig. 3. Perfect recreation from (a) to (c) using a (b) simple square mask.

It can be seen in figure 3 that the technique is able to fully reconstruct the original image. In figure 4, a more complicated mask is applied with areas on the edge and also just around a homogenous area. Despite the complex mask, the technique was also able to get close to the original image. It seems like there is a small bump but is barely noticeable.

C. Images from Paper

To truly test the capabilities of this technique and prove its effectiveness as seen in the paper. Images from the paper are used with figure 6 showing a triangle edge which originally showed overshooting and figure 5 showing the possibility of using inpainting over an inpainting output.

In figure 4, the sign was masked but it can be seen that the second image made the dog look different. However, upon choosing the dog as a mask, the quality of the output looks

(a) (b)

Fig. 4. Complex mask (a) produces a nice (b) output which can be inpainted again to recreate original image.

(a)

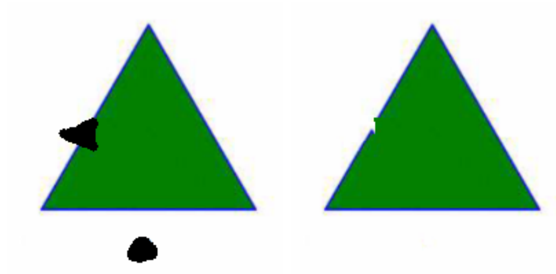
(b) (c)

Fig. 5. Removing the sign from (a) to produce (b). Then removing the dogs from (b) to produce (c).

much better and it doesn't even seem like the sign or the dog was originally there. In figure 5, an edge of the triangle is chosen to test how it would do for a diagonal edge. Although not perfect, the triangle looks almost perfect. One big thing to note as well is the speed of the technique only taking a few seconds. It is important to note that all of the images used here are 200 by 200 pixels but the method extends to much bigger images.

D. Personal Images

Some extra test cases were done using images that are new to the technique. These are personal images taken using a phone camera or from a screenshot of a digital drawing. Figure 1 is a personal photo taken which shows a lot of texture in the



(a) (b)

Fig. 6. Triangle photo from original paper (a) is masked to produce an (b) almost perfect triangle.

background image and a butterfly and a stick in the foreground. To highlight the butterfly, the stick is chosen to be a mask and the resulting output propagates the texture of the ground. In figure 7, spiderman and his web is targeted to be removed. It can be seen that the resulting image has some minor issues. The web at the top wasn't selected due to problems of the method close to the edges, a part of the building is copied, and lastly a random speck is made.

(a) (b)

Fig. 7. Spiderman photo (a) for image inpainting with a complex target mask produces a (b) decent but contains weird artifacts.

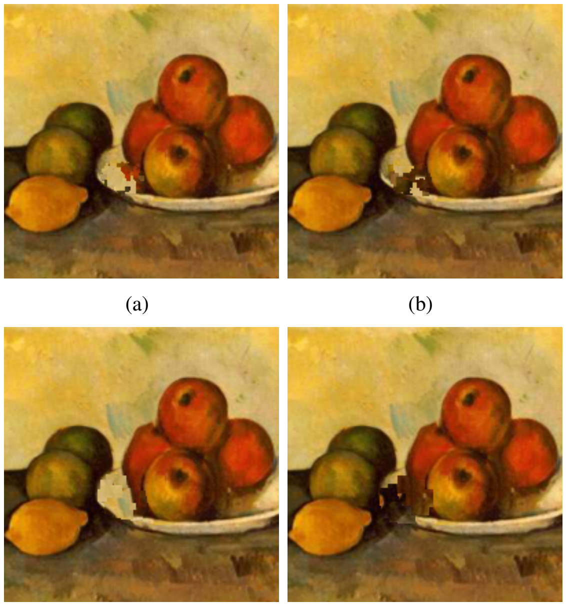
IV. Discussion

A. Patch-Size

Although not mentioned eaerlier, the main hyper-parameter that can be tweaked using this technique is the patchsize. By adjusting the patchsize, the amount of information known inside a certain patch changed. Additionally, this can hinder the resolution of the pattern which produces visible patches or seams. A big thing it affects as well is computational cost and also the number of iterations it needs in order to produce the output. Figure 8 shows the effect of tweaking the patch size. It can be seen that the quality does not have a direct correlation to the patch size. It can also be noted that the speed of the computation is roughly the same for all the patch sizes.

B. Limitations

With all of the success of this method in terms of quality and speed as comparison to different methods as seen in the original paper, there are still lots of limitations to this method. There is a huge limitation if there are no similar



(a) (b) (c) (d)

Fig. 8. Image inpainting using different patch-sizes to cover a small fruit using patch-size (a) 3 (b) 5 (c) 9 and (d) 13. The best result is with (b) which is a smaller patch size than what was used in the rest of this paper.

patches, prediction of what a target mask image may contain (not present), and some images that have curves. Having an abstract image may produce patterns not desired. Background texture to a foreground object. Multiple edge lines on a same contour won't be able to propagate both properly as well. Based on the comments of fellow students, it also seems like the implementation of this technique is non-trivial which might hinder people from using it.

C. Comparison with Current Techniques

This is a paper from almost 2 decades ago. Currently people will need to use photoshop or AI in order to do this inpainting. Using either takes either lots of resources to train or human skill which takes a substantial amount of time to learn. Although this method has its limitations, it can still be seen how good this was compared to any other technique used during that time. For simple and/or smaller images, this technique might be more useful to use than modern techniques. There are a lot of current academic research rooted on this paper and was used as inspiration to other current studies with the same goal which is region filling.

D. Future Work

In terms of future work, I think that this method is great with respect to its compactness and quality output. There are still various region filling methods that can be done that may produce better outputs. A suggestion is to use the successful concepts of filling order and image patching to additional techniques such as adaptive patch-size which can improve the speed and even the quality based on the amount of texture information. A current paper [6] uses the Criminisi technique but adds a factor of entropy for the data term which improves the filling order. However, more work can be done with region filling in terms of properly recreating structure information which might use data information across different points in the contour and connecting them together based on the gradient data across the whole image.

Acknowledgment

The author would like to thank my Image Processing and Applications Mentors, Professor Akcakaya and Teaching Assistant Merve Gulle for inspiring me to persevere and helping me understand some topics in this paper. Also would like to thank my labmates Raphael Riberio and Ricardo Linhares for being with me in this class.

References

[1] A. Criminisi, P. Perez and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," in IEEE Transactions on Image Processing, vol. 13, no. 9, pp. 1200-1212, Sept. 2004, doi: 10.1109/TIP.2004.833105.

[2] A. Criminisi, P. Pérez and K. Toyama, "Object removal by exemplar-based inpainting", Proc. Conf. Computer Vision and Pattern Recognition, 2003-June.

[3] I. N. Sari, K. Masaoka, J. Takarabe and W. Du, "High-Resolution Art Painting Completion using Multi-Region Laplacian Fusion," 2023 Sixth International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 2023, pp. 28-31, doi: 10.1109/IS3C57901.2023.00016.

[4] T. Shukla, P. Maheshwari, R. Singh, A. Shukla, K. Kulkarni and P. Turaga, "Scene Graph Driven Text-Prompt Generation for Image Inpainting," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 2023, pp. 759-768, doi: 10.1109/CVPRW59228.2023.00083.

[5] A. Basu et al., "Digital Restoration of Cultural Heritage With Data-Driven Computing: A Survey," in IEEE Access, vol. 11, pp. 53939-53977, 2023, doi: 10.1109/ACCESS.2023.3280639.

[6] Y. Chen, W. Qi, Q. Jia, H. Peng, Y. Li and C. Zhang, "Image Inpainting Based on Improved Criminisi Algorithm," 2023 8th International Conference on Computer and Communication Systems (ICCCS), Guangzhou, China, 2023, pp. 938-943, doi: 10.1109/ICCCS57501.2023.10150830.

[7] M. Bertalmio, L. Vese, G. Sapiro and S. Osher, "Simultaneous structure and texture image inpainting", Proc. Conf. Comp. Vision Pattern Rec., 2003.

[8] M. Ashikhmin, "Synthesizing natural textures", Proc. ACM Symp. Interactive 3D Graphics, pp. 217-226, 2001-Mar.

[9] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image inpainting", Proc. ACM Conf. Comp. Graphics (SIGGRAPH), pp. 417-424, 2000-July.

[10] A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer", Proc. ACM Conf. Computer Graphics (SIGGRAPH), pp. 341-346, 2001-Aug.

[11] L. Liang, C. Liu, Y.-Q. Xu, B. Guo and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling", ACM Trans. Graphics, 2001.

[12] A. Efros and T. Leung, "Texture synthesis by nonparametric sampling", Proc. Int. Conf. Computer Vision, pp. 1033-1038, 1999-Sept.

[13] P. Harrison, "A nonhierarchical procedure for re-synthesis of complex texture", Proc. Int. Conf. Central Europe Computer Graphics Visualization and Computer Vision, 2001-Feb.

[14] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. . "Hierarchical Text-Conditional Image Generation with CLIP Latents" 2022. ArXiv, abs/2204.06125.