ISSN 2321 3361 © 2020 IJESC

Research Article Volume 10 Issue No.5

Image Reconstruction with the Help of Missing Patch Generation using Machine Learning

Pinakee Kaushik[1], Dr. Neha Agarwal[2]
Student[1], Assistant Professor[2]
Maharaja Agrasen Institute of Technology, GGSIPU, Delhi, India

Abstract:
Image reconstruction and copy image for gery are two of the most popular research fields in image processing nowadays. Various techniques and methodologies have been developed to reconstruct an image. Patch match algorithm is used for finding and replacing the nearest matching patch selected by user. Over the years, many techniques have been developed and used in this application. In this work, we have proposed an approach which is ahybrid of the kNN classification algorithm, k-D tree algorithm and random search. The results suggest that they perform optimally when used together, by reducing the overall time complexity to O(mklog M), where mis the number of pixels in the selected patch and k is the number of nearest pixels that are being searched in the image for every pixel present in the patch. This approach finds the globally dense matches for the patch selected in an efficient and optimal way. The patch match algorithm is especially useful in computer vision applications like object detection and symmetry detection.

Keywords: k-Nearest Neighbors, k-dimensional trees, Patch Match, Random search, copy image Forgery.

I. INTRODUCTION

Today, images are playing a crucial role in our lives. From medical applications to electronic imaging applications, images are everywhere. Digital image processing or digital image reconstruction find application in many fields - Image sharpening and restoration, medical fields, remote sensing, encoding, color processing, etc [1]. One such application is digital image forgery and reconstruction of the image. Image forgery can be employed to hide or manipulate important information present in the image or document. Numerous image forgery techniques[2][3] and algorithms are being used today, such as SVM [4], DCT [5], Generic algorithms and Expanding algorithms[6] to name a few. Many other methods such as DAISY descriptors[7], block matching and residual descriptors[8] are specialized methods for image forgery. The patch match algorithm [9] is an efficient method which works on the idea of finding correspondences between small regions by finding the nearest matching pixels present in the image. Patch match algorithm today is being applied to many fields and areas such as optical flow estimation, stereo correspondence, reshuffling contents of images, etc. Conventionally, the algorithm[9] scans the whole image for similar patches to fill in, which increases the time for running the algorithm with a time complexity of m*n, where m and n represent the number of pixels present in the picture. The proposed work employs a novel approach which completes the work in a time complexity of O(m log M), where m is the number of pixels in the selected patch and M is the number of pixels in the original reference image. A hybrid of the k-NN classification.

The rest of the paper is organized as:-
- Section 2 gives a brief introduction of the related work to the article.
- Section 3 explains the methodology of the work.
- Section 4 will discuss the conclusion and results of the work.
- References are listed in the end.

2. RELATED WORK

Digital Image Forgery algorithms are generally slow. Patch Match algorithms when run conventionally has very less computational efficiency. In our approach, we have reduced the searching time by using a hybrid of k-D tree and k-NN. Earlier works have approached the problem differently. In[10], the authors have developed a framework called patch match filter (PMF). The approach is a convergence of two techniques-patch match-based randomized search and efficient edge a waring image filtering. The paper has also proposed a super- pixel based search mechanism that helps in improving the computational efficiency. A modified version of the patch-match algorithm is used in [11] for handling large displacement motions, it uses edge preserving nearest neighbors for propagation patterns in addition to offsets. Patch Match algorithm is extensively used fro application in super resolution. A lot of work has been done in this field with patch match [12][13][14].[12] introduces a new framework which uses in paint ment to reduce the complexity of the algorithm.[13] uses multi-alias patch match,[14]has used edge prior synthesis to increase the efficiency.

3. METHODOLOGY

After researching several different approaches used in the path, we tried to design a new approach in order to improve upon the metrics that define the smooth functioning and execution of the application. The proposed methodology applies some of the intuitions generated from the s, in contrast to the changes made by us to optimize the previous approaches in order to save time. The approach present in this paper can be summarized into 3 stesps. In the first step, each pixel in the selected patch is assigned a random pixel's value apart from the values of the pixels present in the selected patch. This step is basically the

initialization step. Secondly, propagation is done on the region that is outside of the selected patch region in order to find out the best alternative for every pixel present in the target patch. Secondly, propagation is done on each pixel, and each pixel checks for a better matching pixel from the rest of the pixels present in the image. If a better pixel is found, then the respected pixel is replaced with the value of that pixel. This process is repeated for each pixel present in the selected patch. Lastly the search radius is increased in order to find out better candidates for the replacement of every pixel in the selected patch. The main goal behind the approach used here is that we need to find a corresponding patch for the patch (or region) selected by the user, from the same image or some reference image. This idea of computing correspondences between various regions of the image is one of the main issues of many computer vision problems. The search for correspondences of a region can be classified as either local, or global. The local search is carried out in a limited spatial window, whereas in global search all possible windows are considered, except for the selected patch window. Firstly, an image is taken as input and is then converted into a grayscale image. Then a patch is chosen in the form of 4 points which is converted into a rectangular patch cloud for the algorithm requirement. This can be customized to different shapes of patches other than rectangle too. Lastly, the main task is to find the corresponding pixels to be filled in selected patch, from the other regions of the image that includes everything except the selected patch region. This task is done with the use of k-D-trees, ordered map or dictionary and a max heap. Initially a 3-D-tree is formed using all the pixels present in the image, such that the root is the first pixel, and then any next pixel that comes while iterating the image over each and every pixel except the patch region pixels is kept either on the left or right side of root on the basis of alternate checks for x, y and z at each level. Simultaneously, the address of the node of the pixel in the k-D-tree is stored in a map or dictionary so that the reference of the pixel can be accessed in O(1) time in the future. Once the k-D-tree is formed, then an array is made of the size n x k, which will store the k nearest for each pixel present in the patch, and n is the total number of pixels present in the patch. Then, the patch is iterated over each pixel and its corresponding reference is accessed from the map in O(1) time, and then the k pixels in the 3-D-tree which lie above and below the node in the 3-D-tree are kept in a max Heap with the Euclidean distance of each pixel from the original respective pixel that lies in the selected patch. Also, the pixel coordinates are stored in each node of a max Heap, but the Heap property is based on the Euclidean distance from the original pixel. Then, next k-nearest neighbours from the k-D-tree are traversed to check if they can replace the pixels in the current max heap. A pixel can replace the top element of the heap, if its Euclidean distance from the original selected pixel is less then the Euclidean distance of the top element pixel of the heap. Finally, all k-element pixels in the heap are kept into the n x k array, by popping from the heap. The first pixel popped from the heap is kept at the last column in the n x k array and the last element popped from the heap is kept at the the first column of the n x k array, because the last element popped will be the most nearly similar pixel for the original selected pixel. This process is repeated for every pixel in the patch to fill its k-nearest neighbours in the n x k array. Once the n x k nearest neighbor array is filled with nearest similar pixels for every pixel in the selected patch, the value of every pixel in the selected patch is replaced with the value at the first column of the n x k array for the respected pixel in the array. Finally, the image obtained is the output image, which has the old patch replaced with new similar pixels from the neighbouring regions of the selected patch from the rest of the image. The intuition behind the patch_match is to return an array (of the size of all pixels in the image) with k values per location in the array, which is done to attach k number of nearest matching pixels for each pixel in the image. Once this array containing k values for every pixel is returned in the form of offsets and weights, we extract out the 1st nearest matching pixel for every pixel present in the patch and replace it with the actual pixel value, which is then the final output image.

Figure.1. Approach Pipeline

Algorithm:

1. Initialize each pixel in the selected patch with random values
2. Propagate over every pixel in the selected patch
3. While propagation, check for the nearest k pixels in the k-D-tree for nearest pixels and push them in a max heap on the basis of their Euclidean distance
4. Increasing the radius of searching in the k-D-tree by checking for the next k pixels, and push any pixel in max heap if its

Euclidean distance is smaller than the top element in max heap, and pop the top element of heap.

5. Finally, empty the heap for a pixel by popping and storing the values in an n x k array.

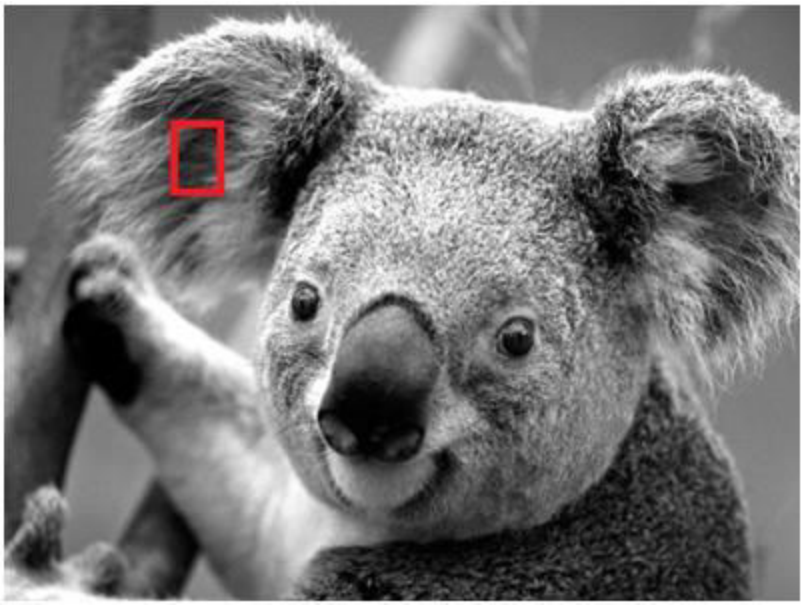

Figure.2. Input Image

4. CONCLUSION AND FUTURE WORK

This algorithm can be extended for use in various other important Computer Vision applications like Image Retargeting, Image Recompletion, Image Reshuffling, Object detection and Symmetry detection. This algorithm can be extended for use in various other important Computer Vision applications like Image Retargeting, Image Recompletion, Image Reshuffling, Object detection and Symmetry detection. Although our algorithm is faster than the O(m^2) approach (m is the number of pixels in the pixels), but still there is a room for growth. There can be a better approach using some different search algorithm or a combination of various search algorithms in the future. Also since in this approach we have worked on the technique to replace a patch with the corresponding patch from the same image, we can also try replacing the patch selected with the most similar patch found in some other reference image.

Figure.3. Output Image

5. REFERENCES

[1]. G. A. Baxes, Digital Image Processing: Principles and Applications, New York:Wiley, 1994

[2]. Gajanan K. Birajdar, Vijay H. Mankar, Digital image forgery detection using passive techniques: A survey, Digi- tal Investigation, Volume 10, Issue 3, 2013, Pages226-245,ISSN1742-2876, https://doi.org/10.1016/j .diin. 2013. 0 4.007. (http://www. science direct. Com /science/ article/ pii/ S174 2287 613000364)

[3] Muhammad Ali Qureshi, Mohamed Deriche, A bibliography of pixel-based blind image forgery detection techniques, Signal Pro- cessing: Image Communication, Volume 39, Part A, 2015,

Pages 46-74, ISSN 0923-5965, https:/ /doi .org/ 10.1 016/ j.ima ge. 2015.08.008. (http: //www. science direct. com/ science/ article/ pii/ S09 26515 00 13 93)

[4]. Ryu SJ., Lee HY., Cho IW., Lee HK. (2008) Document Forgery De- tection with SVM Classifier and Image Quality Measures. In: Huang YM.R. et al. (eds) Advances in Multimedia Information Processing - PCM 2008. PCM 2008. Lecture Notes in Computer Science, vol 5353. Springer, Berlin, Heidelberg

[5]. Alahmadi, A., Hussain, M., Aboalsamh, H. et al. SIViP (2017) 11: 81. https://doi.org/10.1007/s11760-016-0899-0

[6]. Gavin Lynch, Frank Y. Shih, Hong-Yuan Mark Liao, An efficient expanding block algorithm for image copy-move forgery detection, Information Sciences, Volume 239, 2013, Pages 253- 265, ISSN 0020-0255, https:// doi. org/ 10. 1016/ j.ins.2013.03.028. (http: //www. science direct .com/ science/ article/ pii/ S0020025513002338)

[7]. Jing-Ming Guo, Yun-Fu Liu, Zong-Jhe Wu, Duplication forgery detection using improved DAISY descriptor, Expert Systems with Applications, Volume 40, Issue 2, 2013, Pages 707-714,ISSN0957-4174, https:// doi.org/ 10.1016/ j.eswa. 2012. 0.0 02 (http://www.sciencedirect.com/science/ article/ pii/ S09 57417 41 20 09 633)

[8]. D.Cozzolino, D.Gragnaniello and L.Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 5297-5301. doi:10. 1109/ IC IP. 2014.7026072

[9]. Barnes C., Shechtman E., Goldman D.B., Finkelstein A. (2010) The Generalized Patch Match Correspondence Algorithm. In: Daniilidis K., Maragos P., Paragios N. (eds) Computer Vision – ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6313. Springer,Berlin, Heidelberg

[10]. Jiangbo Lu, Hongsheng Yang, Dongbo Min, Minh N. Do; The IEEE Conferenceon Computer Visionand Pattern Recog nition (CVPR), 2013, pp.1854-1861

[11]. Linchao Bao, Qingxiong Yang, Hailin Jin; The IEEE Conference on Computer Visionand Pattern Recognition (CVPR),2014,pp.3534-3541

[12].Le Meur O., Guillemot C. (20 12) Super- Resolution- Based Inpaint ing. In: Fitzgibbon A., Lazebnik S., Perona P., Sato Y., Schmid C. (eds) Com- puter Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7577. Springer, Berlin, Heidelberg

[13].Shi W.etal.(2013)Cardiac Image Super-Resolution with Global Correspondence Using Multi-Atlas Patch Match. In: Mori K., Sakuma I., Sato Y., Barillot C., Navab N. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013. MICCAI 2013.Lec- ture Notes in Computer Science, vol8151.Springer, Berlin, Heidelberg

[14]. Y. Tai, S. Liu, M. S. Brown and S. Lin, "Super resolution using edge priorand single image detail synthesis," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Fran- cisco, CA, 2010, pp. 2400-2407.doi: 10.1109/CVPR.2010.5539933