

A NOVEL IMAGE INPAINTING FRAMEWORK BASED ON MULTILEVEL IMAGE PYRAMIDS

Md. Salman Bombaywala^{1*} – Chirag Paunwala²

¹Uka Tarsadia University, Bardoli, Gujarat, 394350, India

²Department of Electronics and Comm. Engg., Sarvajani College of Engg. & Tech., Surat, Gujarat, 395001, India

ARTICLE INFO

Article history:

Received: 15.1.2019.

Received in revised form: 13.3.2019.

Accepted: 14.3.2019.

Keywords:

Exemplar-based inpainting

Image Inpainting

Multilevel

Pyramid

DOI: <https://doi.org/10.30765/er.1405>

Abstract:

Image inpainting is the art of manipulating an image so that it is visually unrecognizable way. A considerable amount of research has been done in this area over the last few years. However, the state of art techniques does suffer from computational complexities and plausible results. This paper proposes a multi-level image pyramid-based image inpainting algorithm. The image inpainting algorithm starts with the coarsest level of the image pyramid and overpainting information is transferred to the subsequent levels until the bottom level gets inpainted. The search strategy used in the algorithm is based on hashing the coherent information in an image which makes the search fast and accurate. Also, the search space is constrained based on the propagated information thereby reducing the complexity of the algorithm. Compared to other inpainting methods; the proposed algorithm inpaints the target region with better plausibility and human vision conformation. Experimental results show that the proposed algorithm achieves better results as compared to other inpainting techniques.

1 Introduction

Image inpainting manipulates an image in an undetectable manner. These manipulations include various applications ranging from object removal, scratch removal, text removal, red-eye removal, missing data reconstruction and so on. Inspired by the artwork of traditional restoration artists, Bertalmio et.al. [1] presented the digital implementation of image inpainting using partial differential equations (PDEs). They used anisotropic diffusion [2] to spread the information around the edge of the target region. The algorithm worked well for inpainting small regions while it failed to inpaint larger regions. Olivieria et.al. [3] presented an algorithm to inpaint an image using a convolution mask. The presented method was faster but it introduced blurring when inpainting larger regions. Many methods to inpaint textures in an image are inspired by the work of Efros and Leung [4]. Texture synthesis methods work well for restoring textures in an image. Bertalmio et.al. [5] used PDEs and texture synthesis methods for different image parts to develop an algorithm that simultaneously inpainted structures and textures in an image.

Criminisi et.al. [6] presented an exemplar-based approach to inpaint the target region by copying the information directly from the known part of the image. The best match was found using sum of squared differences (SSD). The method is explained in the next section. To improve the results of inpainting, researchers have proposed many variations of the above method. While most example based techniques use SSD to find the best matching patch, Xu and Sun [7] presented a sparse representation based approach. They used a linear combination of few best candidate patches to reconstruct the image.

In this paper, we propose a new multilevel pyramid-based image inpainting approach which utilizes coherency sensitive hashing [8] to find the best exemplar. The proposed method gives plausible results. The rest of the paper is organized as follows. In section 2 we discuss and review exemplar-based inpainting algorithms and relevant variations. In section 3 we present the proposed algorithm with implementation

* Corresponding author

E-mail address: srbombaywala@gmail.com

details. In section 4 we compare our algorithm with other state of art methods for different types of images. We conclude in section 5.

2 Exemplar-based image inpainting algorithms

Criminisi et.al. [6] have effectively presented the exemplar-based approach for image inpainting. The developed iterative algorithm inpainted structure as well as textures simultaneously. The algorithm required a mask representing the area to be inpainted; also known as the target region and the algorithm iterated over the boundaries of this mask. The basic mechanism of the algorithm is summarized in Figure 1. Each iteration consists of three main steps: Patch Priority Calculation, Best Match Search and Patch Propagation.

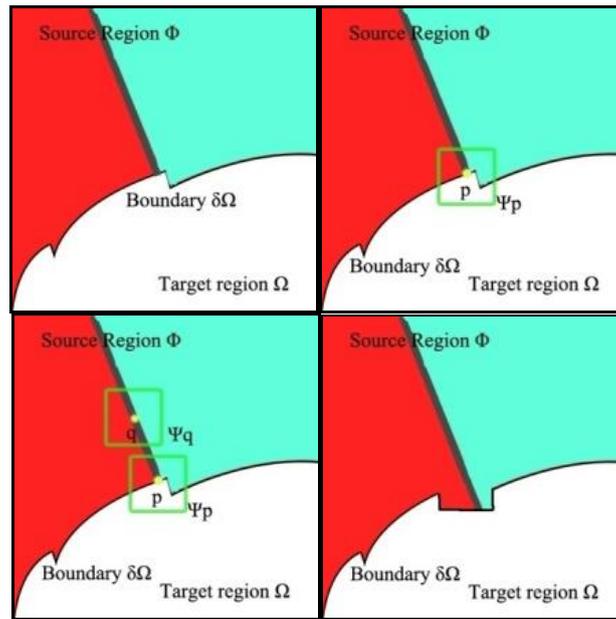


Figure 1. The basic mechanism of exemplar-based image processing.

2.1 Patch Priority Calculation

The patch priority determines the sequence of inpainting over the target region boundary $\delta\Omega$. Calculation of the patch priority involves determining data term and a confidence term for all the pixels on $\delta\Omega$. Every patch on the boundary contains some information about the source region. The confidence term gives priority to the patch which has the most information from the source region. The data term prioritizes the patches located on the structures. This ensures that the algorithm inpaints the structures first. The human visual process first notices the structures in a scene and then focuses on the details. To be consistent with the above fact, the calculation of data term is important. Many variations in the calculation of data term have been proposed over the time. Criminisi et.al. [6] used gradients on $\delta\Omega$ to calculate the data term. Other variants include the use of cross isophotes [9], tensors [10], structure sparsity [7], total variation [11], ISEF [12] etc. A comparative analysis of example-based methods is presented by Bombywala et. al. [13]. The final priority is calculated as a combination of the confidence term and the data term. We have used the ISEF based data term for our algorithm.

2.2 Best Match Search

For a patch with the maximum priority on $\delta\Omega$, the search for the best match is carried out on the source region. Many approaches have been presented to do find the best match. Criminisi et.al. [6] used the sum of squared differences to exhaustively find the best match from the entire source region. The match score is calculated only with respect to the known part of the target patch on $\delta\Omega$. As SSD tends to prefer pixels of a uniform region, a modified Bhattacharya distance was used to find the best match [14]. Other distance metrics like hamming distance and normalized cross correlation can also be used to find the best matching pair of patches [15].

However, these methods are computationally costly and have a high probability of error since only one patch was used. To overcome this, Wong and Orchard [11] used multiple patches by assigning corresponding weights to generate the best match. However, this method resulted in unnecessary smoothing. Xu and Sun [7] proposed the reconstruction using sparse representations. Although the results obtained were better, it introduced computational complexity in form of dictionary learning and updates. DCT coefficient based best match search was used to speed up the process [16]. Randomized patch search was also utilized to find the best match [17]. In our algorithm, we have used Coherency Sensitive Hashing (CSH) [8] for finding the best patch match. CSH is fast and accurate as the information is propagated much faster because it uses the similarity of appearance in the image plane.

2.3 Patch Propagation

Once the best match is obtained, the information must be propagated in the target region. As shown in Figure 2, the patch at the boundary $\delta\Omega$ contains an unknown part and some known part. Only the pixels belonging to the unknown parts are substituted in patch propagation. It should be noted here that the best match was found by only considering the pixels belonging to the source region. Most algorithms rely on direct substitution of patch pixels the target region. However, the sparsity based inpainting methods use linear combination of more than one patch to fill the target region. This leads to unnecessary smoothing which is more plausible in larger target regions. Xu and Sun [7] used an additional component to find the neighborhood compatibility of the best match patch before propagating the information. This resulted in improved quality of inpainting at the cost of increased complexity. Our algorithm utilizes pyramid based approach. We propagate the inpainted information from one level to other. We also define a neighborhood around the best match and limit our search space to the corresponding neighborhood at other level. This eliminates the need to find the compatibility and hence speeds up the process and decreases the complexity of the algorithm. We utilize downsampling process to generate the pyramid and hence the information is propagated to the next level using reverse logic.

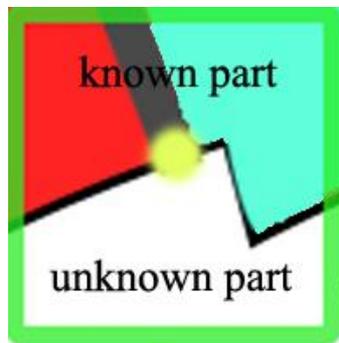


Figure 2. The patch on boundary $\delta\Omega$ containing some part of the source region and some part of the target region.

3 Multilevel Pyramid based Image Inpainting

In this section, we describe the proposed image inpainting algorithm. The main steps on overall framework of the proposed method are shown in algorithm 1.

Algorithm 1: Multilevel Pyramid-based Image Inpainting framework

1. Build the pyramid of the input image and its mask by using downsampling.
 2. Choose the lowest resolution image and inpaint it.
 3. Propagate the information to the next level and inpaint the new level.
 4. Continue until the last level (highest resolution) is reached.
-

We downsample the image by a factor of 2 at every level to generate the image pyramid. This results in subsequent levels of half size in both the dimensions. Multi-resolution property of wavelet is also utilized in

literature for inpainting [18, 19]. However, these methods intended to separate out structure and texture and inpaint both individually. Our algorithm inpaints the textures and structures simultaneously and hence is computationally less complex. In our algorithm, we have restricted the number of level to $R=3$. Figure 3 shows the basic structure of image pyramid and one such image pyramid used. Multilevel pyramid of the mask is also calculated before the start of inpainting.



Figure 3. Image Pyramid.

After the construction of multilevel image pyramid, image inpainting is done on the top level (lowest resolution). First, the boundary of the target region is calculated and for each pixel ‘p’ on the boundary $\delta\Omega$, priorities are calculated as the product of the confidence term and the data term.

$$P(p) = C(p) * D(p) \quad (1)$$

where $C(p)$ is the confidence term calculated as

$$C(p) = \frac{\sum_{q \in \mathcal{V}_p \cap (I-\phi)} C(q)}{|\mathcal{V}_p|} \quad (2)$$

and $D(p)$ is the data term calculated as

$$D(p) = \frac{|\nabla I_{isef}|}{\alpha} \quad (3)$$

∇I_{isef} is calculated as suggested by Goswami and Paunwala [12] and α is the normalizing factor. In our experiments, we kept the value of α to be 255 for 8 bit images.

For the pixel with the highest priority on $\delta\Omega$, best match patch is calculated from the source region. We use CSH algorithm [8] to find the best matching patch. CSH utilizes hashing to initialize the matching and image coherence to find the best match. CSH speeds up the process by relying on the fact that the images are coherent in nature. That is, if a pair of similar patches is found, then their neighbors are also likely to be similar.

CSH mainly consists of two stages: Indexing stage and Search stage. In the indexing stage, primitive hash functions based on Walsh-Hadamard kernels are used to create hash tables. Similar patches irrespective of their location in the image are likely to be hashed to the same bin. In the search stage, CSH generates a rich set of candidate patches by considering both appearance and spatial coherence. Once the best matching patch is found, the match point coordinates are stored and the target patch is inpainted using the best match patch. The match point coordinates are further used to define a neighbourhood for search space in the next level. Only the pixels belonging to target region are replaced and the mask image is updated. The above process is repeated till all the pixels in the masked image are processed.

The next step of the algorithm propagates the information calculated in the low-resolution level ‘R’ to the next level ‘R-1’. Since the levels are generated by downsampling the above level by 2, every pixel in the

above level will map to 4 pixels in the next level (See Figure 4.). Therefore, the best match coordinates are also propagated. The inpainting is done on the (R-1)th level.

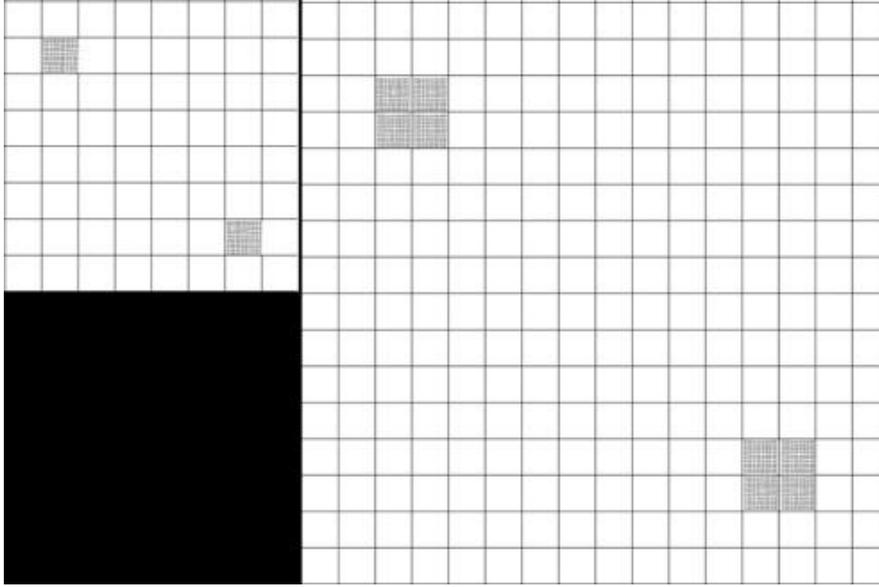


Figure 4. Pixel mapping between pyramid levels.

Since the best match points are available from Rth level, the search for the best match is restricted to the neighbourhood of the propagated match points only. This reduces the computational complexity and hence reduces the time of inpainting. The above procedure is repeated for the next subsequent levels till the last level (Original image) is inpainted. Algorithm 2 shows out multilevel image pyramid inpainting algorithm.

Algorithm 2: Proposed Image Inpainting

Data: Input image I , mask Ω , resolution number R .

Result: Inpainted Image

$\{I^r\}_{r=1}^K \leftarrow \text{ImagePyramid}(I)$

$\{\Omega^r\}_{r=1}^K \leftarrow \text{MaskPyramid}(\Omega)$

for $r = R:1$ do

 while $\Omega^r \neq \{\}$ do

$\partial\Omega \leftarrow \text{edge } \Omega$

$\vec{P} \leftarrow \text{Priority vector over } \partial\Omega$

$\{M^r\} \leftarrow \text{Matchpoint } s(\max(\vec{P}))$

$I^r(\psi_p^r) \leftarrow \psi_m$

 update Ω^r

 end

$\{M^{r-1}\} \leftarrow \text{propagate}\{M^r\}$

$I^{r-1}\{\psi_p^{r-1}\} \leftarrow \text{propagate}(I^r\{\psi_p^r\})$

if $r = 1$, then inpainted image $\leftarrow I$

end

Our algorithm works with a low-resolution image of a small size which makes the execution faster. Even for subsequent levels, the best match is found from a defined neighbourhood rather than from the entire image which makes the algorithm speedy. The algorithm is also consistent with the human visual system. When presented with any scene, human vision focuses on the large structures in the image and then the details. Our algorithm inpaints structures first and then as the level becomes finer, the details are inpainted gradually.

4 Results and Discussion

We have tested the implemented algorithm over varieties of images and compared the results with PDE based inpainting approach [1], convolution based inpainting approach [3], example-based inpainting approach [6] and ISEF based exemplar inpainting approach [12]. Objective analysis comparing the peak signal to noise ratio (PSNR), structural similarity index (SSIM) and feature similarity index (FSIM) shows the effectiveness of our algorithm.

We have tested the algorithm for various cases of object removal, text removal and scratch removal.

Figure 5 shows the original images discussed in this paper along with the percentage of damaged region.



Figure 5. Original set of images with the percentage of are to be inpainted used to show the effectiveness of our algorithm in this paper.

Figure 6 and Figure 7 show the example of object removal. As can be seen, PDE and TV-based method fails to remove large object plausibly. Our result looks more plausible when compared to example-based or ISEF-based inpainting results. The linearity of the structural segments is maintained by our algorithm. Figure 8 and 9 show the examples of scratch removal. As can be seen, the PDE-based, TV-based and convolution-based approaches introduce blurring. The exemplar-based techniques are tested for patch sizes ranging from 2 to 14 and are not able to remove the scratches effectively (see Figure 9). Our algorithm reconstructs textures accurately. However, some blurring effect is visible in texture part (Figure 9). Figure 10 shows the example of text removal. Although some blocking effect is visible in our results, still the proposed algorithm performs better than the existing counterparts. TV based algorithm produces results which are comparably plausible for text removal cases. However, the PSNR values show the effective reconstruction using proposed algorithm.



Figure 6. Object Removal. (a) PDE [1] (b) TV [20] (c) Convolution [3] (d) Exemplar [6] (e) ISEF [12] (f) Proposed.



Figure 7. Object Removal. (a) PDE [1] (b) TV [20] (c) Convolution [3] (d) Exemplar [6] (e) ISEF [12] (f) Proposed.

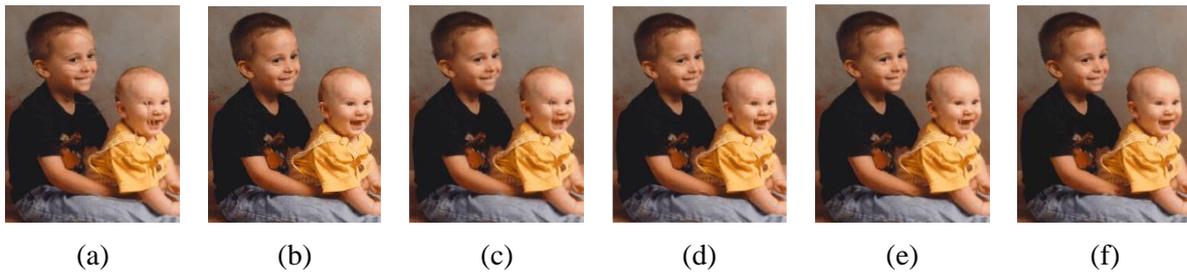


Figure 8. Scratch Removal. (a) PDE [1] (b) TV [20] (c) Convolution [3] (d) Exemplar [6] (e) ISEF [12] (f) Proposed.

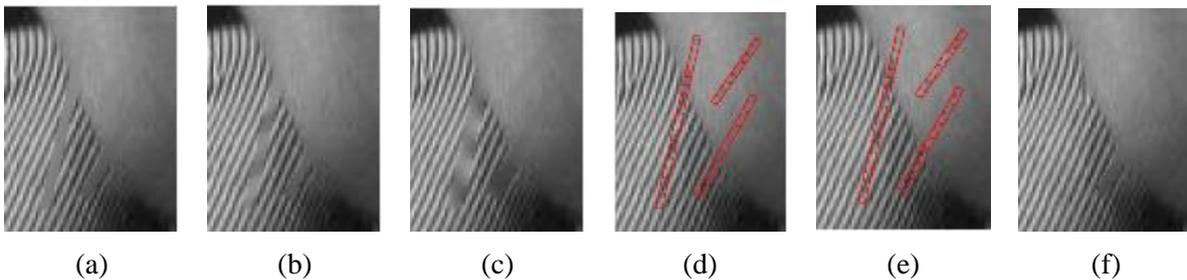


Figure 9. Scratch Removal. (a) PDE [1] (b) TV [20] (c) Convolution [3] (d) Exemplar [6] (e) ISEF [12] (f) Proposed.



Figure 10. Text Removal. (a) PDE [1] (b) TV [20] (c) Convolution [3] (d) Exemplar [6] (e) ISEF [12] (f) Proposed.

Table 1 shows the PSNR values of reconstructed results for various techniques. Since ground truth for object removal is not present, the PSNR values are calculated for scratch and text removal cases. As can be seen, our algorithm performs better than the other in reconstructing the damaged image.

Figure 11 shows the comparative performance of our algorithm for different objective analysis indices. We can see that the proposed inpainting algorithm produces one of the best perceptual quality images. It can be seen that our algorithm excels other algorithms in PSNR, SSIM and FSIM metrics.

Table 1. PSNR values of inpainted images using different techniques.

	PDE [1]	TV [20]	Convolution [3]	Exemplar [6]	ISEF based [12]	Proposed
Image1	48.2419	51.8818	46.0715	49.6660	50.1011	52.4670
Image2	43.3915	48.2105	42.1415	45.7225	45.8654	50.8057
Image3	34.7942	39.8682	35.7721	46.3419	36.3175	45.3317
Image4	34.7238	39.3405	34.6648	28.4525	28.5034	46.8827
Image5	25.3376	31.9140	24.9292	18.0645	18.1039	38.7182
Image6	33.2160	33.9966	29.3138	39.4422	39.7502	45.1334
Image7	31.8985	31.8993	29.1736	23.0824	23.0603	41.8260

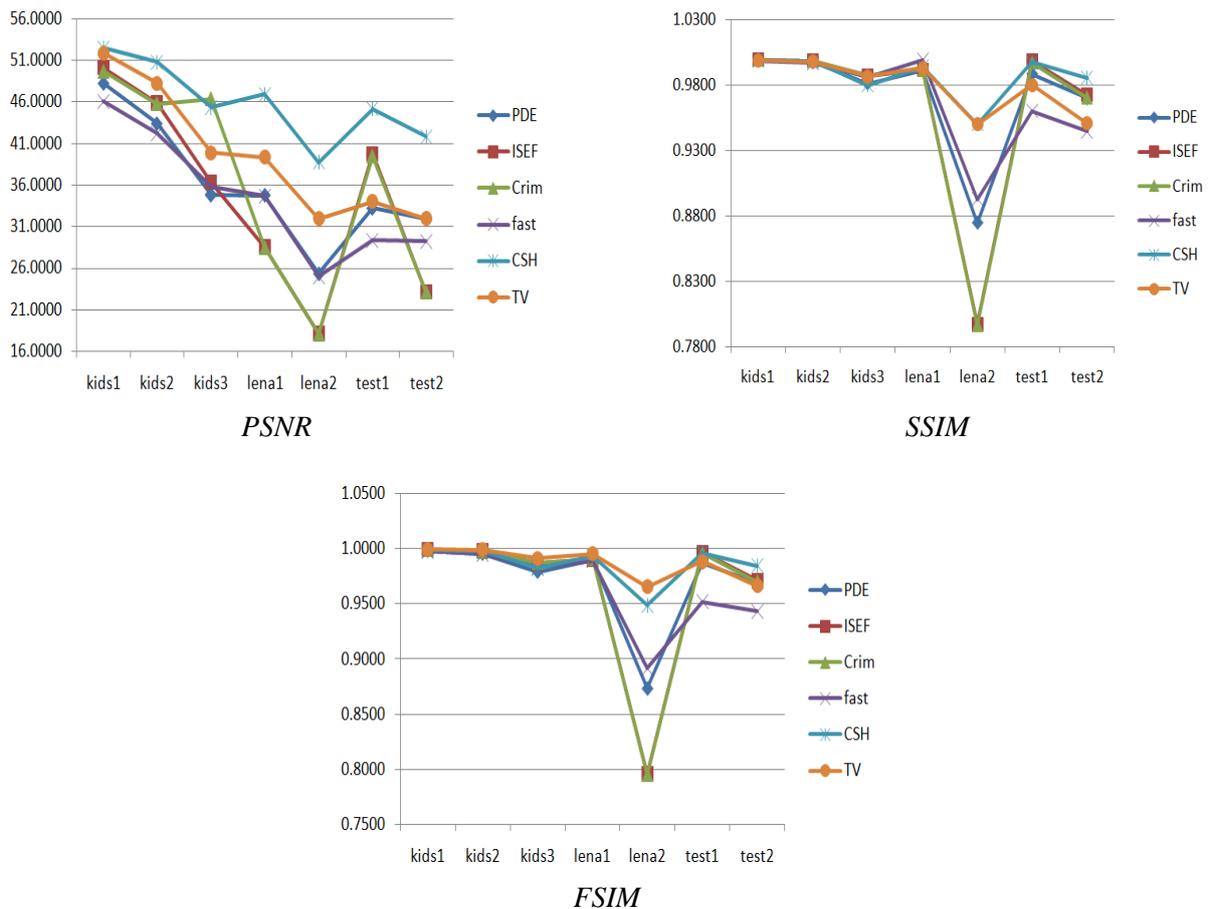


Figure 11. Comparative objective analysis of inpainting methods.

5 Conclusion

In this paper, we propose a multilevel pyramid based image inpainting algorithm. The image is represented as a pyramid and the inpainting starts at the coarsest level which speedups the process. Next, the information is passed to the next level which reduces the search space and brings other benefits. The utilization of coherency sensitive hashing method to find the best match provides accurate and quick results. The proposed algorithm is compared with existing methods and the results show improvement in the resulting image. Moreover, as discussed, the proposed algorithm is consistent with the human visual process and it is confirmed by the obtained results.

References

- [1] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting, The 27th annual conference on Computer graphics and interactive techniques, New Orleans, LA, USA, 2000, 417-424.
- [2] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on pattern analysis and machine intelligence, 12 (1990), 7, 629-639.
- [3] Olivieria, B., Richard, M., Chang, M.Y.S.: Fast digital image inpainting, International Conference on Visualization, Imaging and Image Processing, Marbella, Spain, 2001, 106-107.
- [4] Efros, A. A., Leung, T. K.: Texture synthesis by non-parametric sampling, The Seventh IEEE International Conference In Computer Vision, Kerkyra, Greece, 1999, 1033-1038.
- [5] Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting, IEEE transactions on image processing, 12 (2003), 8, 882-889.
- [6] Criminisi, A., Pérez, P., & Toyama, K.: Region filling and object removal by exemplar-based image inpainting, IEEE Transactions on image processing, 13(2004), 9, 1200-1212.

-
- [7] Xu, Z., & Sun, J.: Image inpainting by patch propagation using patch sparsity, *IEEE transactions on image processing*, 19(2010), 5, 1153-1165.
 - [8] Korman, S., Avidan, S.: Coherency sensitive hashing, *The 13th International Conference on Computer Vision*, Barcelona, Spain, 2011, 1607-1614.
 - [9] Wu, J., Ruan, Q.: Object removal by cross isophotes exemplar-based inpainting, *The 18th IEEE international conference on Pattern Recognition*, Hong Kong, China, 2006, 810-813.
 - [10] Le Meur, O., Gautier, J., Guillemot, C.: Exemplar-based inpainting based on local geometry, *The 18th IEEE International Conference on Image Processing*, Brussels, Belgium, 2011, 3401-3404.
 - [11] Wong, A., & Orchard, J.: A nonlocal-means approach to exemplar-based inpainting, *The 15th IEEE International Conference on Image Processing*, San Diego, California, 2008, 2600-2603.
 - [12] Goswami, P., & Paunwala, C.: Exemplar-based image inpainting using ISEF for priority computation, *International Conference on Circuits, Systems, Communication and Information Technology Applications*, Mumbai, India, 2014, 75-80.
 - [13] Bombaywala, M.S., Goswami, P., Paunwala, C.: Comparative analysis of exemplar based image inpainting techniques, *The 2nd IEEE conference on Emerging Technology Trends in Electronics, Communication and Networking*, Surat, India, 2014, 1-6.
 - [14] Bugeau, A., Bertalmío, M., Caselles, V., Sapiro, G.: A comprehensive framework for image inpainting, *IEEE Transactions on Image Processing*, 19 (2010), 10, 2634-2645.
 - [15] Ralekar, C., Dhondse, S., Mushrif, M. M.: Image reconstruction by modified exemplar based inpainting, *The 2nd IEEE International Conference on Communications and Signal Processing*, Chengdu, China, 2015, 1005-1010.
 - [16] Kwok, T. H., Sheung, H., Wang, C. C.: Fast query for exemplar-based image completion, *IEEE Transactions on Image Processing*, 19 (2010), 12, 3106-3115.
 - [17] Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D. B.: PatchMatch: A randomized correspondence algorithm for structural image editing, *ACM Transactions on Graphics*, 28 (2009), 3, 24.
 - [18] Cho, D., Bui, T. D.: Image inpainting using wavelet-based inter-and intra-scale dependency, *The 19th IEEE International Conference on Pattern Recognition*, Tampa, Florida , 2008, 1-4.
 - [19] Zhang, H., Dai, S.: Image inpainting based on wavelet decomposition, *Procedia Engineering*, 29 (2012), 3674-3678.
 - [20] Getreuer, P.: Total variation inpainting using split Bregman, *Image Processing On Line*, 2 (2012), 147-157.