International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Texture Matching Based Image Inpainting Using Gaussian Filtering

[1]B. Srinisha, [2]Mrs. N. Krishnammal

[1]ME CSE, 2nd Year – Sri Shakthi Institute of Engineering and Technology.
[2]Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology.

ABSTRACT

Texture similarity and local intensity smoothness are both essential for solving most image inpainting problems. In this work, we propose a novel image inpainting algorithm that is capable of reproducing the underlying textural details using a texture measure and also smoothing pixel with the proposed Gaussian-weighted filtering. The proposed algorithm is compared with other image inpainting algorithms under different scenarios including object removal, texture synthesis, and error concealment. Experimental results show that the proposed algorithm outperforms the existing algorithms.

Keywords – Gaussian-Weighted filtering, Target Localization, Texture Similarity, Inpainting using Gaussian filtering, Image innpainting.

INTRODUCTION

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms a core research area within engineering and computer science disciplines too.

Digital Image Processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focus particularly on images. DIP focuses on developing a computer system that is able to perform processing on an image. The input of that system is a digital image and the system process that image using efficient algorithms, and gives an image as an output.

An image is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically. The value of f(x,y) at any point is gives the pixel value at that point of an image. Pixel is the smallest element of an image. Each pixel corresponds to any one value. In an **8**-bit gray scale image, The value of the pixel between 0 and 255. Each pixel stores a value proportional to the light intensity at that particular location.

Yeh, Raymond A.,pp 5845-5493 2017, propose a new method for semantic image inpainting that generates missing content by conditioning on available data. Yang, Chao, and Xin Lu., IEEE 2019, created high-resolution image inpainting using multi-scale neural patch synthesis. Deep learning advances have shown exciting promise in filling large gaps in natural images with semantically plausible and context aware details, influencing fundamental image manipulation tasks like object removal. Yu, Jiahui, and Xin Lu. (2018) generate image inpainting with contextual attention by inpainting large missing regions in an image using deep learning-based approaches. These methods can produce visually plausible image structures and textures, but they frequently produce distorted structures or blurry textures that do not match the surrounding areas. Image inpainting for irregular holes using partial convolutions was proposed by Liu, Andrew Tao, and Bryan Catanzaro in 2018. Existing deep learning-based image inpainting methods apply a standard convolutional network to the corrupted image, with convolutional filter responses conditioned on both valid pixels and substitute values in masked holes. They propose using partial convolutions, which are masked and renormalized to condition on only valid pixels.

In proposed system, a novel image inpainting algorithm that is capable of reproducing the underlying textural details using a texture measure and also smoothing pixel with the proposed Gaussian-weighted filtering. The proposed algorithm is compared with other image inpainting algorithms under different scenarios including object removal, texture synthesis, and error concealment.

- Inpainting operation is enhanced by Gaussian filter.
- Refined patches are invariantly not identified with edges.
- Gaussian distribution based filtering process enhances the inpainted region.

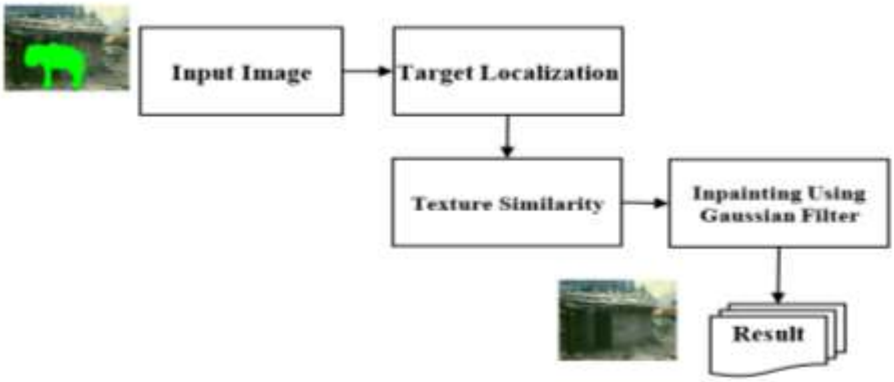
In this paper, the proposed method had been documented in upcoming section and it have been experimented with benchmark datasets to prove texture matching based image inpainting using Gaussian filtering. Finally, the results are concluded in the result analysis section.

METHODOLOGY

In this paper, we examined the texture matching based image inpainting problem by using Target Localization, Texture Similarity Matching and Inpainting Using Gaussian Filter. In this paper, the Gaussian Filtering method is proposed in detail.

In this method, the texture matching image inpainting model has been developed based on the Gaussian filtering. Input images are given into the model and the target localization selects the target patch from the source image where the unknown part from the missing region. A Gaussian filter is a linear filter. It's usually used to blur the image or to reduce noise. If you use two of them and subtract, you can use them for "unsharp masking" (edge detection). The Gaussian filter alone will blur edges and reduce contrast. After picking the target patch, we utilise a new texture similarity metric to choose the candidate patches from the source region patches.

Then selected target patch get generated by texture similarity which measure texture similarity between source region and target patch and the measure is computed using weighted pixel values.



Target Localization:

First apply a priority function to choose the next target patch. The target patch has a known part from the source region and an unknown part from the missing region. The center of the target patch lies on the contour of the outer border (δφ) of the missing region. The priority function contains two terms, a confidence term and a data term. The confidence term C(ψp) is the ratio of known pixels within the patch. The data term D(ψp) computes the dot product of the isophote vector $\overrightarrow{\nabla I_p}^{\perp}$ (along the direction of equal intensity lines as shown) andthe normal vector $\overrightarrow{n_p}$ at the center pixel p.

The priority function is defined as,

$$P(\Psi p) = C(\Psi p)D(\Psi p), P\epsilon\delta\phi$$

Where,

$$\boldsymbol{C}(\boldsymbol{\Psi}_p) = \frac{|\boldsymbol{\Psi}_p \cap \boldsymbol{\Psi}_c|}{|\boldsymbol{\Psi}_p|}, \boldsymbol{D}(\boldsymbol{\Psi}_p) = \frac{\left|\overrightarrow{\nabla I_p}^{\perp} \cap . \overrightarrow{n_p}\right|}{\boldsymbol{I}_{max}}$$

Where $\overrightarrow{\nabla I_p}^{\perp}$ is the isophote vector orthogonal to the gradient $\overrightarrow{\nabla I_p}$ (computed using the central difference operator) at centerpixel p, Imax is the maximum possible gray-level value, whichis 255 in our case, and $\overrightarrow{n_p}$ is the unit vector orthogonal to theouter border. Here, $\overrightarrow{\nabla I_p}^{\perp}$ and $\overrightarrow{n_p}$ are computed at centerpixel p.

Texture Similarity Matching:

After choosing the target patch, we use a new texture similarity measure for selecting the candidate patches from the source region patches ψq centered at pixel such that ψq^Ω=φ . The proposed similarity measure satisfies the following objectives. First, the measure is able to compute the textural similarity between a chosen candidate patch the source region and the target patch. Second, the measure is computed using weighted pixel values, where the weights are given by a Gaussian kernel, whose center is adaptively chosen for different target patches with varying spatial distributions of known and unknown pixels.

Inpainting Using Gaussian Filter

After choosing the candidate patches, we apply the α-trimmed mean filter [48], [49] to the λ candidate patches using the procedure. For each missing pixel from the target patch at the index location l, we group the corresponding pixels at the same index location within the candidate patches into a set S = Iˆ qi(l);i = {1,2,···,λ}, and order the intensities in the set S in ascending order to obtain

S_o ={X_j; j ={ 1,2,···,λ}}, satisfying

$X_1 \le X_2 \le \cdots \le X_\lambda$.

Then, we apply the α-trimmed mean filter to the set So to obtain the intensity of that missing pixel, given by,

$$I_p(l) = mean_\alpha(S_0) = \frac{1}{\lambda - 2\alpha\lambda} \sum_{j=\alpha\lambda+1}^{\lambda-\alpha\lambda} X_j$$

For a given ratio α, the α-trimmed mean of the λ elements in So is obtained by ignoring the αλ smallest elements and the αλ largest elements, and then computing the mean of the remaining elements. In our experiment, we use α =0.2. The λ candidate patches are chosen based on the similarity between the known part of the target patch and the corresponding part of the candidate patches; thus, the other part of each candidate patch (corresponding to the unknown portion of the target patch) could be very different from that of other candidate patches. For this reason, we choose the α-trimmed mean instead of a full sample mean or a fully truncated mean (median).

When computing the intensity of a missing pixel using the corresponding pixels from the candidate patches, the α-trimmedmean filter is less sensitiveto outliers (extremely large or small intensity values) than the full sample mean. Also, the α-trimmed mean filter makes use of more than one pixel, which provides a better estimate than the fully truncated mean.

The proposed inpainting algorithm can be applied to inpaint small or large missing regions in grayscale or color images. When inpainting color images, the input image is defined as I : D → R3, and the isophote vector at each pixel is computed as the average of the isophote vectors of the three color channels and is given by,

$$\vec{\nabla} I_p{}^{\perp} = \frac{1}{3} \sum_{t \in \{R,G,B\}} \nabla \vec{I}_{p,t}^{\perp}$$

where t represents the red (R), green (G), or blue (B) channel, and ∇Ip,t⊥ is the isophote vector at center pixel p in channelt . The NLTS measure for inpainting color images is given by,

$$NLTS(I_{\hat{p}}, I_q) = exp\{-\sum_{t\in(R,G,B\}} \left\| (I_{p,t} - I_{q,t})^{o2} o G_p \right\|_1\}$$

where Iˆ p,t and Iq,t are the intensity values of the patch p and q, respectively, in channel t. When inpainting a missing pixel at the index location l, we fill in each channel of the missing pixel separately by computing the α-trimmed mean of the set

$S_{o,t}$= {X j,t; j ={ 1,2,⋯,λ}}, which consists of the intensities of the corresponding pixels within the candidate patches, in channel t.

$$I_{p,t}(l) = mean(S_{o,t}) = \frac{1}{\lambda - 2\alpha\lambda} \sum_{j=\alpha\lambda+1}^{\lambda-\alpha\lambda} X_{j,t}$$

EXPERIMENTAL RESULTS AND ANALYSIS

In this section, We evaluate the proposed algorithm along with current inpainting algorithms for three different image inpainting applications: object removal, texture synthesis, and error concealment. In the applications of object removal and texture synthesis, the missing region to be inpainted may have an irregular shape, while in the application of error concealment, the missing region is usually a square or a rectangular block.

Parameter Setting

There are five variable parameters in the proposed algorithm: the patch size m, the parameter h in the NLTS measure, the number of candidate patches λ, the value ofα in the α-trimmed mean filter, and the standard deviation of the Gaussian kernel σ. In all of our experiments, the values of h, λ, andα are fixed. The patch size depends on the size of the image texture pattern. A small patch size relative to the size of the texture pattern may fail to reconstruct the texture of the image. We use a patch size in the range of [3, 17] for our proposed algorithm. The value of σ depends on the patch size (σ ∝m) and the busyness [54] of the image texture. If σ is very small, the pixels on the edge of a patch would have a Gaussian weight close to zero, so that when computing the similarity between patches, a large patch size may act the same as a smaller patch size. Moreover, when propagating a narrow structure (e.g., a single straight line), the value of σ should be smaller than that used for inpainting a textural image, because pixels near the center of the Gaussian kernel, (xc, yc), may be sufficient to match the structure. In our experiments, we tried the value of σ in the range of [0.5,4], and we found that it gives good results in the range of [1.5,2.5].

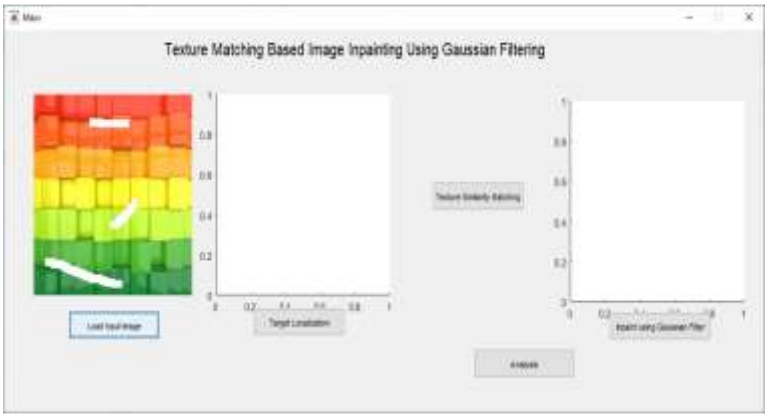
Inpainting Textured Images

The proposed image inpainting algorithm is able to reconstruct different types of texture due to the NLTS measure, in contrast to other image inpainting methods.

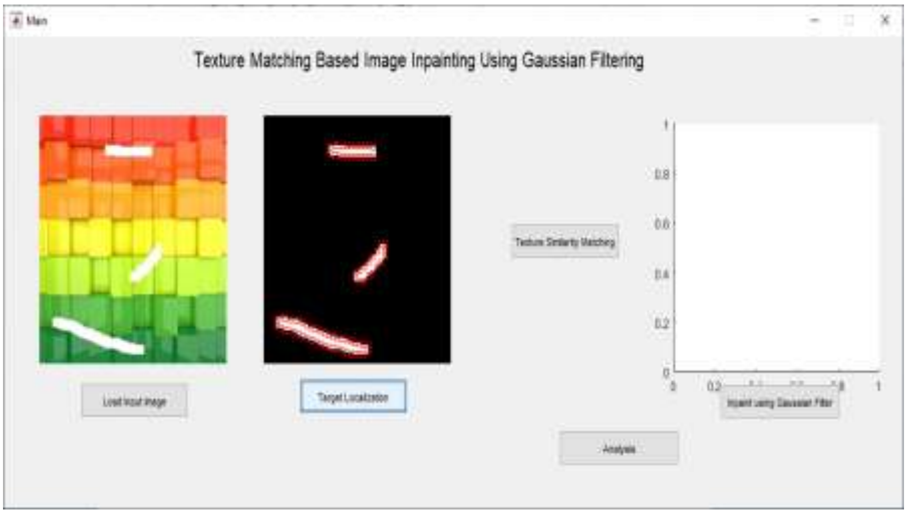
Error Concealment

In this section, we apply the proposed image inpainting algorithm for the application of recovering missing data from digital images after wireless transmission. If the transmitted images are divided into blocks of size 8 × 8 when transmitted (e.g., JPEG), the received images may lose an entire block or consecutive blocks due to noise. Our proposed algorithm is able to recover texture and geometric structure within the missing blocks.

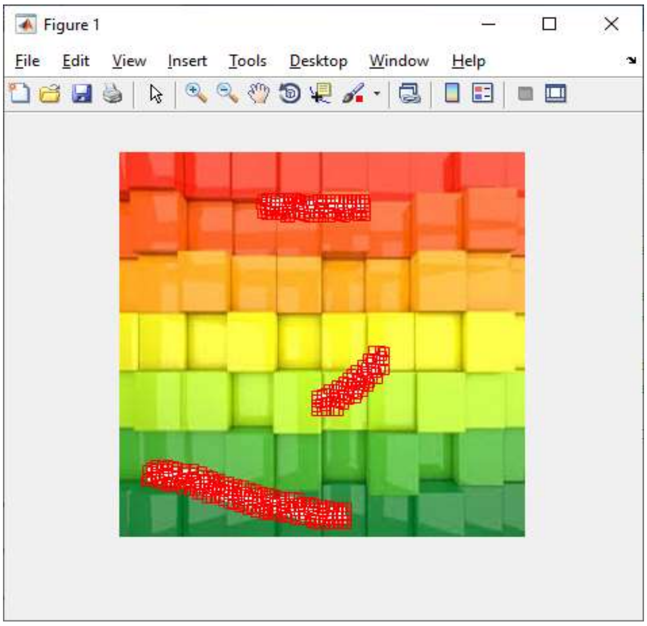
Load Input Image:



Target Location:



Texture Similarity Matching:



Output:

CONCLUSION

In this work, we have developed an efficient and robust image inpainting method that uses a new nonlocal texture similarity measure to search for several candidate examplers for each target patch, which are then fused together using the Gaussian filter to fill in each pixel within the target patch. For the application of object removal, our experimental results demonstrate that the proposed algorithm performs better than the other inpainting algorithms w.r.t. both the qualitative analysis and the observer studies. For the application of inpainting texture images, the proposed algorithm outperforms the other inpainting algorithms in terms of the qualitative appearance, even though the quantitative metrics do not always agree. For the application of error concealment, our experimental results show that the proposed algorithm is capable of recovering different textures and structures within the missing blocks of the received image. In addition, our proposed algorithm is also very fast in inpainting images as compared to other method.

FUTURE ENHANCEMENT

As a future work, we plan to extend the method to very high-resolution inpainting applications using ideas similar to progressive growing of GANs. The proposed future inpainting framework and contextual attention module can also be applied on conditional image generation, image editing and computational photography tasks including image-based rendering, image super-resolution, guided editing and many others.

References

1) M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in Proc. ACM Conf. Comput. Graph., 2000, pp. 417–424.

2) M. Bertalmio, A. N. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., Jun. 2001, pp. I355–I362.

3) F. Li, L. Pi, and T. Zeng, "Explicit coherence enhancing filter with spatial adaptive elliptical kernel," IEEE Signal Process. Lett., vol. 19, no. 9, pp. 555–558, Sep. 2012.

4) T. Chan and J. Shen, "Mathematical models for local nontexture inpaintings," SIAM J. Appl. Math., vol. 62, no. 3, pp. 1019–1043, Jul. 2006.

5) T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," J. Vis. Commun. Image Represent., vol. 4, no. 12, pp. 436–449, Dec. 2001.

6) C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," IEEE Trans. Image Process., vol. 10, no. 8, pp. 1200–1211, Aug. 2001.

7) J. Shen, S. H. Kang, and T. F. Chan, "Euler's elastica and curvaturebased inpainting," SIAM J. Appl. Math., vol. 63, no. 2, pp. 564–592, Jul. 2002.

8) H. Grossauer and O. Scherzer, "Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D," in Proc. 4th Int. Conf. Scale Space Methods Comput. Vis., 2003, pp. 225–236.

9) J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," Numer. Algorithms, vol. 53, no. 1, pp. 67–92, 2010.

10) P. Li, S.-J. Li, Z.-A. Yao, and Z.-J. Zhang, "Two anisotropic fourth-order partial differential equations for image inpainting," IET Image Process., vol. 7, no. 3, pp. 260–269, Jun. 2013.

11) H. Grossauer, "A combined PDE and texture synthesis approach to inpainting," in Proc. Eur. Conf. Comput. Vis., vol. 3022, T. Pajdla and J. Matas, Eds. Berlin, Germany: Springer, May 2004, pp. 214–224.

12) A. Bugeau and M. Bertalmio, “Combining texture synthesis and diffusion for image inpainting,” in Proc. Int. Conf. Comput. Vis. Theory Appl., 2009, pp. 26–33.

13) A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” IEEE Trans. Image Process., vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

14) J. Wu and Q. Ruan, “Object removal by cross isophotes exemplarbased inpainting,” in Proc. IEEE Int. Conf. Pattern Recognit., Jun. 2006, pp. 810–813.

15) L. Cai and T. Kim, “Context-driven hybrid image inpainting,” IET Image Process., vol. 9, no. 10, pp. 866–873, Oct. 2015.