J Inf Process Syst, Vol.13, No.5, pp.1183~1202, October 2017
https://doi.org/10.3745/JIPS.02.0031

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Image Restoration and Object Removal Using Prioritized Adaptive Patch-Based Inpainting in a Wavelet Domain

Rajesh P. Borole* and Sanjiv V. Bonde*

Abstract

Image restoration has been carried out by texture synthesis mostly for large regions and inpainting algorithms for small cracks in images. In this paper, we propose a new approach that allows for the simultaneous fill-in of different structures and textures by processing in a wavelet domain. A combination of structure inpainting and patch-based texture synthesis is carried out, which is known as patch-based inpainting, for filling and updating the target region. The wavelet transform is used for its very good multiresolution capabilities. The proposed algorithm uses the wavelet domain subbands to resolve the structure and texture components in smooth approximation and high frequency structural details. The subbands are processed separately by the prioritized patch-based inpainting with isophote energy driven texture synthesis at the core. The algorithm automatically estimates the wavelet coefficients of the target regions of various subbands using optimized patches from the surrounding DWT coefficients. The suggested performance improvement drastically improves execution speed over the existing algorithm. The proposed patch optimization strategy improves the quality of the fill. The fill-in is done with higher priority to structures and isophotes arriving at target boundaries. The effectiveness of the algorithm is demonstrated with natural and textured images with varying textural complexions.

Keywords

Image Inpainting, Object Removal, Region Filling, Texture and Structure Propagation, Wavelet Inpainting

1. Introduction

The two most commonly used graphical techniques for filling the gaps after object removals are image inpainting and texture synthesis. Digital image inpainting refers to the inpainting process performed on digitized images. Using an algorithm, the image with the mask is filled in such a way that resulting image looks natural and undistorted. The image manipulation technique is the process of removing objects from an image that starts with masking out the undesired object and making a gap in the area occupied by that object. Then, this gap is filled using digital inpainting techniques. The application of these methods for filling the gaps in an image is called constrained texture synthesis. It is used for filling images that have more textured areas. It is observed that neither structural inpainting nor texture synthesis alone can provide the ultimate solution. The combination of these two produces good results. Image inpainting considers the image as the collection of structures, shapes, and objects

Manuscript received August 18, 2014; accepted April 15, 2015; onlinefirst December 7, 2015.
Corresponding Author: R. P. Borole (rpborole@yahoo.com)
* Dept. of Electronics & Telecom, Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, MS, India ({rpborole, svbonde11}@yahoo.com)

that are separated from one another by sharp edges but where each one is smooth itself; whereas, the texture synthesis considers the low stochasticity features of the image.

In this paper, we propose a technique of image inpainting in a discrete wave transformed domain by using prioritized patch-based inpainting with performance improvement and patch optimization. Filling the damaged region is done by estimating coefficients from the surrounding wavelet coefficient in LL, LH, HL, and HH subbands.

2. Related Work

There are various approaches observed for image reconstruction in the literature based on image inpainting and texture synthesis with their own advantages and limitations. These techniques are listed as level lines, partial differential equation (PDE)-based inpainting, fluid interpolation, Euler's elastic, bounded variation, heat transfer, etc. [1-3].

(a) (b)

Fig. 1. Large object removal. (a) Original image. (b) Inpainted image

Texture replication and propagation is used to fill in the gaps in images. Different approaches have been developed for synthesizing textures, including statistical and image-based methods. Art restorators completes the missing part of the images by applying their knowledge and abilities to connect the unknown part and complete it, which is described as the connectivity principle in neuropsychology [4-6]. We used the same approach with minimum user interaction that is limited to the preparation of the mask of targeted damaged areas. Our automatic algorithm does the rest. The proposed patch-based inpainting approach overcomes many of the limitations of earlier approaches.

Automatic digital image inpainting was first brought into the field of image processing by [1]. This algorithm imitates the traditional inpainting processes, such as identifying the area to be corrected, marking the boundary of the region to be filled, and continuing the lines of similar color by using anisotropic diffusion-based on PDEs over the fill front. Another PDE approach-based on vector field regularization is used in [7]. These techniques are only effective for small scratches. These algorithms work well with the images that are relatively smooth and that have low noise or texture. The process is unable to fill in regions that are highly textured or contain a large amount of noise that leads to the non-continuation and blurring of the edges. Since the time, there have been a number of algorithms

proposed to solve the inpainting problem [2,3,8]. These algorithms are used to only fill narrow gaps in images and they fail to reconstruct large damaged regions in the images. Motion estimation and auto regressive models to interpolate losses in video stills from adjacent frames are used in [9]. The basic idea was to copy the right pixels from neighboring frames into the gap. The technique couldn't be applied to still images or film, where the regions to be inpainted span over many frames.

The work in [10] decomposes the original image into two components, which are processed separately by inpainting and texture synthesis. Finally, the outputs of the two are summed up as the resultant image. This approach still remains limited to the removal of small image cracks and the diffusion process used continues to blur the filled region. Drori et al. [11] describes an algorithm that interweaves a smooth approximation with exemplar-based detail synthesis for image completion. Both of these algorithms have been found to be extremely slow.

Harrison [12] was the first to propose the usage of exemplar-based synthesis specifically for object removal. He proposed that the level of 'texturedness' of the pixel's neighborhood determine the order in which a pixel in the target region was filled. Although his intuition is sound, strong linear structures are often overruled by nearby noise, reducing the value of the extra computation. A related technique in [13] drove the fill order by the local shape of the target region, but this method fails to explicitly propagate linear structures.

Several researchers have considered texture synthesis as a way to fill large image regions with 'pure' textures. Pure textures are defined as repetitive two-dimensional textural patterns with moderate stochasticity. This is based on a large amount of research that has been carried out on texture synthesis. This research has to replicate texture from the given small source sample of pure texture [14-16]. These sample-based techniques cheaply and effectively generate new texture by sampling and copying color values from the source [17-20]. It only contains the process to replicate textured areas in the images. In the case of regular textures, sample-based inpainting works well, but it fails to replicate structures in the still images.

In contrast with previous approaches, it was necessary to introduce the technique that does not require the user to specify where the fill-in information comes from. It can be automatically accomplished by the algorithm. The technique should allow to simultaneously fill-in numerous regions containing completely different structures and surrounding backgrounds. In addition, no limitations should be imposed on the topology of the region to be inpainted. Therefore, the patch-based approach of Criminisi et al. [21] seems to be more appropriate. In continuation, we are proposing wavelet domain processing driven by isophote energy and patch optimization. This results in an algorithm that has the efficiency and qualitative performance of exemplar-based texture synthesis, as well as linear structure inpainting, with the advantages of processing the edge details in various frequency bands at various resolution levels separately.

The methods in [22,23] propagate the image texture in a semiautomatic way, where the user needs to sketch the structure that must be preserved in the image. Afterwards, the algorithm just fills the region with the appropriate textures. Such methods present good results, but at the cost of user intervention.

Image inpainting based on wavelet techniques started with the work of Chan et al. [24] and Rane et al. [25], who used inpainting algorithms in order to recover lost wavelet coefficients by estimating information from surrounding wavelet coefficients. The algorithm in [25] repairs sparse blocks of JPEG image data lost in wireless communications. Their method fills 8×8 blocks of wavelet coefficients using a specific approach for each DWT level detail. It is more like a denoising algorithm for blocking salt and

pepper noise, rather than an actual inpainting algorithm. In fact, this type of technique is not exactly an inpainting technique.

A technique focused on image inpainting in the wavelet domain was presented by Patwardhan and Sapiro [26]. They used the projection-based convex sets (POCS). Their method projects the image in a wavelet domain and estimates the target region pixel coefficients using surrounding wavelet coefficients and projects it onto the original image to fill the missing region pixels by moving back and forth from spatial to wavelet domains and vice versa. However, their method is not efficient due to the computational overhead for filling the regions with complex textural information and a larger target area.

The techniques proposed in [27] work with the DWT mask but leads to a diffusion blur in the case of a large target region. The authors in [28] proposed CWT processing while the method proposed in [29] works in wavelet transform with the fast marching method (FMM). The approaches proposed in [30-32] used wavelet transform techniques for reproducing edge details with various techniques like total variation (TV), divisional regularization, the fast optimization transfer algorithm, etc. mostly for processing and restoring noisy images. They work better for simple textures and small cracks in images. The method proposed in [33] uses structure propagation by level lines and Bezier curve approximations to reproduce the edges, but it only works for small regions. Most of the algorithms proposed in [27-32] reproduce edges and simple textures better for small damaged regions, but they lead to the existence of artifacts in large regions and big object removal.

In this paper, we are proposing a prioritized patch-based image inpainting method applied to wavelet transformed scaling and detailed subbands by providing an estimation of the patch's isophote energy. Since it was observed in our previous work on spatial domain inpainting [34,35] that the most effective patch size for inpainting depends on various types details exist in the image under consideration. The additional contribution of our current study is patch size optimization based on the best matching distance in all subbands and RGB planes with a variational energy of wavelet coefficients in the surrounding of the pixel to be filled.

3. Wavelet Domain Processing, A Brief Review

As a powerful mathematical tool, the wavelet transform (WT) finds several applications in computer graphics and image processing. The continuous wavelet transform of a function $f(x) \in L^2(\mathbb{R})$ is given by:

$$W_{(a,b)} = \langle f_{(x)}, \psi_{a,b}(x) \rangle \tag{1}$$

$$W_{(a,b)} = \int_{-\infty}^{+\infty} f_{(x)} \times \psi^*_{a,b}(x)\, dx \tag{2}$$

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \times \psi\left(\frac{x-b}{a}\right) \tag{3}$$

where a, b are scaling and translation constants respectively. DWT is a discretized version in which the spatial domain function $f(x)$ can be expressed as a series expansion of the scaling and wavelet functions as proposed in [36] and given below:

$$f(x) = \sum_k W_\varphi(j_0, k)\, \varphi_{j_0,k}(x) + \sum_{j=j0}^{\infty} \sum_k W_\psi(j, k)\, \psi_{j,k}(x) \tag{4}$$

where, $\varphi_{j_0,k}(x)$, $\psi_{j,k}(x)$ represent the scaling and wavelet functions, respectively, and satisfy:

$$\varphi_{j,k}(x) = \sum_n h_\varphi(n)\, 2^{(j+1)/2}\, \varphi(2^{j+1}x - n) \tag{5}$$

$$\psi_{j,k}(x) = \sum_n h_\psi(n) 2^{(j+1)/2}\, \varphi(2^{j+1}x - n) \tag{6}$$

for all $j, k \in \mathbb{Z}$. The $W_\varphi(j_0, k)$ & $W_\psi(j, k)$ are the scaling, wavelet coefficients, respectively, and are given by:

$$W_\varphi(j, k) = \sum_n h_\varphi(n - 2k)\, W_\varphi(j + 1, n) \tag{7}$$

$$W_\psi(j, k) = \sum_n h_\psi(n - 2k)\, W_\varphi(j + 1, n) \tag{8}$$

where, $h_\varphi(n - 2k)$ and $h_\psi(n - 2k)$ are the scaling and wavelet vector expansions, respectively, which are basically forms of LPF and HPF filters. Eqs. (7) and (8) reveal a remarkable relationship between the DWT coefficients of adjacent scales. The scale j approximates and details coefficients that can be computed by convolving the scale j+1 approximation coefficients $W_\varphi(j + 1, n)$ with the time reversed scaling and wavelet vectors $h_\varphi(-n)$ and $h_\psi(-n)$ and dyadic subsampling of the result. The low-pass filtered approximation is resubmitted to this process to get the next level of decomposition. The process for Level 1 decomposition is shown in Fig. 2 for a 2D image.

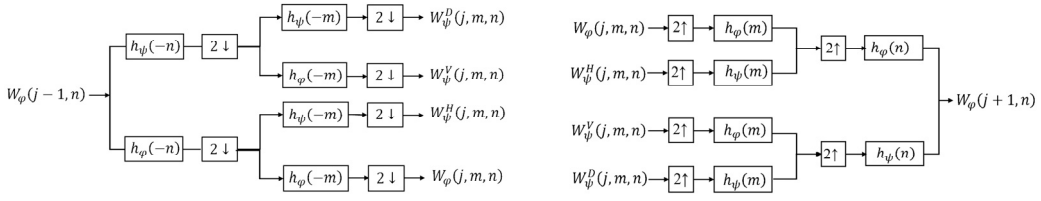


Fig. 2. (a) 2D image wavelet decomposition. (b) 2D image wavelet reconstruction.

The wavelet analysis is done by applying the dilation and translation of function $\psi_{0,0}(x)$, which is also called the *mother wavelet*. The notion of scale is very important in the wavelet theory in that the coarser the scale of analysis, the fewer details of the signal there are that are resolved. On the other hand, a finer scale resolves more details of the processed signal or image. The implementation of DWT in Fig. 2 is based on the pyramidal algorithm proposed by Mallat [37]. It applies 1D transform column-wise and then row-wise to the image. The resultant coefficients are the scaling (LL), horizontal (HL), vertical (LH), and diagonal (HH), respectively, which are detailed as:

$$\{W_\varphi(j, m, n),\ W_\psi^H(j, m, n), W_\psi^V(j, m, n), W_\psi^D(j, m, n)\ \} \tag{9}$$

An example of decomposition in two dyadic scales using DWT is illustrated in Fig. 3. It is important to note that two properties of the DWT, on which this work is strongly based, are: that the

approximation image is smooth, and that the detail images provide information on image edges that are high frequency details, which are separated over various bands by the band pass filtering effect of DWT. They can be separately tackled and superposed to obtain the final image.

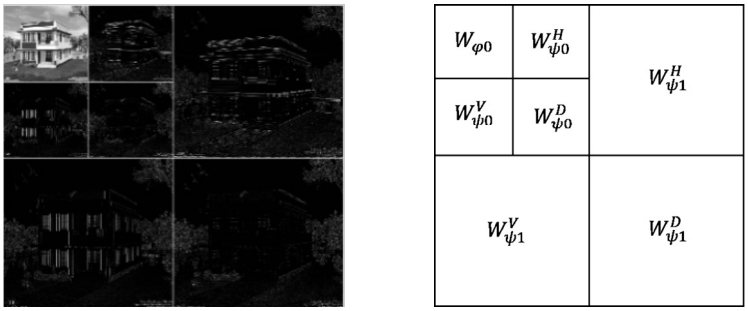


Fig. 3. Two level wavelet decomposition (a) and the wavelet coefficients (b).

4. Proposed Algorithm (PBI_DWT)

Image inpainting and texture synthesis appear to be different, but they might actually complement each other. If we could combine the advantages of both approaches, we would get a clear gap filling that is the natural extension from the surrounding area. Criminisi et al. proposed the algorithm that does exactly this [21]. They used the sampling concept from Efros and Leung's approach [38]. The improvement over Efros and Leung's method was that Criminisi et al.'s approach took isophotes into consideration and gave higher priority to the points that were on the boundary of a gap, as well as on structures, isophotes, and corners, which led to them being naturally extended into the gap. To identify these points, Criminisi et al. [21] assigned a priority value to all the pixels on the boundary of the gap. The important points on structure and corners were then assigned higher priorities according to the algorithm, and thus, the linear structures were extended first.

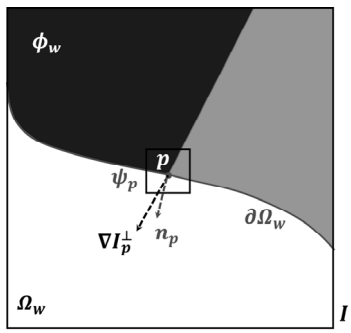


Fig. 4. Notation diagram.

The proposed model in this paper for digital image inpainting in the wavelet domain explores wavelet coefficients to ensure the proper preservation of the image structure and the correct filling of the

inpainting region with our prioritized patch based texture synthesis and patch optimization technique. We are proposing the additive form of priorities, which is more linear than [21], and additionally uses the variational isophote energy of the fill-front patches to decide the priorities along the fill-. The notation used in the algorithm is shown in Fig. 4, where I is an entire subband and ϕ_w, Ω_w, $\partial\Omega_w$ are the source region (undamaged portion of the image), the target region, and the target region boundary (fill-front), respectively. For each pixel on the boundary, the patch ψ_p is constructed with the pixel at the center, as shown in Fig. 4. The n_p is a normal to the contour $\partial\Omega_w$ of the target region Ω_w and $I_p^{\perp}$ is the isophote (direction and intensity) at point p. The source region is denoted by ϕ_w and the entire subband is denoted with I.

The algorithm is supplied with the damaged image I_u and the user defined mask image I. The mask image is marked with the target area, Ω_w with any suitable color. The decomposition of both images is done by the decimated DWT. The resulting LL approximation and HL, LH, and HH details the subband's target regions are then fill with our prioritized patch based algorithm with patch optimization at every decomposition levels. The corresponding mask specifications are maintained using the subbands of mask images. The approximation and detail subbands are processed separately. The final image is reconstructed from these filled subbands using inverse DWT. The algorithm is explained in the following section.

4.1 Prioritized Patch-Based Inpainting Algorithm with Patch Size Optimization in a Wavelet Domain

To keep track of filling the DWT coefficients in the target region, the binary mask is prepared from the mask image, with 1 at the positions of the pixels to be filled. The target region is cleared with 0 in the input image $\hat{I}_u$ and subjected to wavelet decomposition. The border of this region creates an artificial edge, which leads to generating nonzero coefficients in details wavelet decomposition. The extension of this border edge is a function of the support of the chosen wavelet basis where compact support wavelets present less of a spread and wider support wavelets produce more of spread of the unwanted edge information in the target region. It is minimized by restricting to the compact support db1 wavelet and by keeping the decomposition level in the range of 1-3.

As the DWT coefficients are position dependent on the subbands of the same and all other higher-level decompositions, any shift in filling leads to a target region border visualization effect. This can be tackled by the wavelet domain inpainting mask subband of the mask image I.

The algorithm identifies the fill-front pixels and stores their indices for subsequent processing. It assigns the priorities to each pixel on the fill-front. The details of the priority computation are discussed in the following sections. In our proposed algorithm, the patch's priority is the weighted sum of three elements: the confidence term $C_{\psi_p}^{w}(p)$, the data term $D_{\psi_p}^{w}(p)$, and the variational isophote energy $E_{\psi_p}^{w}(p)$. Their contribution is adjusted depending on the structural and textural complexity of the image. The algorithm then finds the highest priority indexed patch $\psi_{p\in\partial\Omega w}$ and searches the matching index patch $\psi_{q\in\phi_w}$ to fill the pixel positions in $\psi_{p\in\partial\Omega w,\forall p\in\Omega_w}$ as:

$$I_{p\in\partial\Omega w,\forall p\in\Omega_w} = I_{q,\{\forall q\in\psi_{q\cap\phi_w}\leftrightarrow p\in\psi_{p\cap\Omega w}\}} \tag{10}$$

The one iteration of the filling process by the proposed algorithm is pictorially shown in Fig. 5, which indicates (a) as original image I with Ω_w as the region to fill, $\partial\Omega_w$ as the target region boundary (fill-front), $\phi_w = I - \Omega_w$ as the source region; (b) as ψ_p, the index patch to be filled centered @ $p \in \partial\Omega_w$; (c) as the probable match of ψ_p along the isophote (e.g., ψ'_q and ψ''_q); and (d) the unfilled portion of $\psi_{p\in\Omega_w}$ is filled from the best corresponding matching patch portion of $\psi_{q\in\phi_w\cap\psi_p}$.

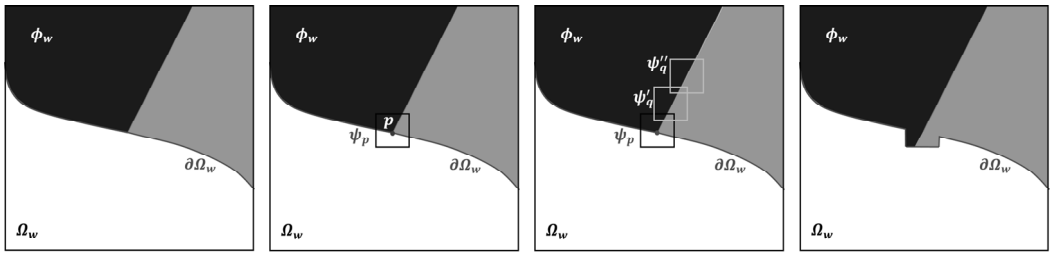


Fig. 5. Structure propagation by patch based synthesis. (a) Fill-front, (b) patch ψ_p, (c) patch search, and (d) fill-region filled.

4.2 Computation of Priorities

As pointed out by other authors in [1] and [21], the random or sequential fill order gives the onion pill effect, which does not produce good results in general [35]. For plausible results we propose selecting a patch with high variational isophote energy and high confidence as the highest priority candidate to fill. In this work the priority is decided by three factors:

1) Variational isophote energy $E^w_{\psi_p}(p)$: The area of higher variational energy is mostly due to more structure and texture. Hence, it is of prime importance in the fill order.
2) Isophote strength and direction $D^w_{\psi_p(p)}$: This term describes how strongly the isophote is hitting the boundary. It boosts the priority of a patch that an isophote 'flows' into. $D^w_{\psi_p}$ is especially important since it encourages linear structures to be synthesized first and is propagated securely into the target region. High strength isophotes need to be processed on priority. For better restoration and continuity of image texture and structure, the strength and the direction of the isophote in the neighborhood of the target region needs to be estimated so that the edges "penetrate" the target region in the same direction as in the source area. This can be ensured by the gradient and normal vector to the boundary at point p, as shown in Fig. 4.
3) Patch confidence $C^w_{\psi_p}(p)$: This term indicates how many pixels are already filled in the patch. The pixels on sharp corners and tendrils are assigned high confidence via this term and are filled on priority, which leads to the simultaneous propagation of the neighboring structure and texture in the gap.

4.2.1 The variational isophote energy, $E^w_{\psi_p}(p)$

The DWT detail coefficients separate the high frequency variations from a smooth background. We find the variational energy to be:

$$E^{w}(p) = \sum_{\forall i,j} {W_{\psi(i,j)}}^{2}$$

$$E_{\psi_p}^{w}(p) = \arg \max_{W_{\psi}^{d}, d \in LH,HL,HH} \left[\sum_{d} \sum_{\substack{\forall i,j \\ \in \psi_p \cap \phi_w}} \left(W_{\psi_{p(i,j)}}^{d} \right)^{2} \right] \tag{11}$$

where $W_{\psi}^{d}, d \in LH, HL, HH$ denotes the DWT detail coefficient at a particular decomposition level and $\forall i,j \in \psi_p \cap \phi_w$ denotes the pixels of patch ψ_p that lies in the source region containing valid information. For a color image we consider the variational energy of three RGB plane patches.

4.2.2 The strength and direction, $D_{\psi_p}^{w}(p)$

The strength in the direction of the isophote is given by data term as:

$$D_{\psi_p}^{w}(p) = \frac{\nabla I_p^{\perp}(x,y) \times n_p}{\alpha}, \qquad 0 \le D_{\psi_p}^{w} \le 1 \tag{12}$$

where $\nabla I_p^{\perp}(x,y)$ is the magnitude along the normal to the fill-front at pixel p and α is a normalization factor (e.g., $\alpha = 255$ for gray level images and $\alpha = 3 \times 255$ for RGB images).

As we know that the LH and HL details capture the edges in horizontal and vertical directions respectively, we propose using them as the gradient components in horizontal and vertical directions. Thus:

$$\nabla I_p^{\perp}(x,y) = \begin{bmatrix} W_{\psi}^{H}(x,y) \\ W_{\psi}^{V}(x,y) \end{bmatrix} \tag{13}$$

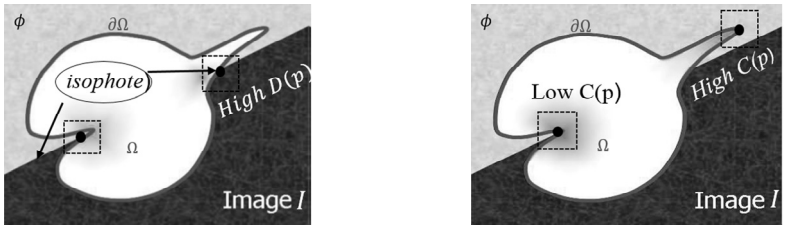
with Eq. (13) and the unit normal vector, n_p, which are found from ∇ operation on the logical mask image, the proposed algorithm decides the isophote strength in the direction of the propagation for processing the LH, HL, and HH coefficients. The estimation of the data term $D_{\psi_p}^{w}(p)$ for the LL subband is done by gradient operating on that subband. The data terms are estimated to be high for the pixel patches on the isophote (indicated in green), as shown in Fig. 6(a).

4.2.3 The confidence

The proposed algorithm assigns high priorities to the pixels on corners and thin tendrils via a confidence term similar to [21]. The confidence of the pixel is estimated by the status of the filled neighborhood of the pixel. In Fig. 6(b), the pixels marked in the green area have a higher confidence to that of the pixel in red colored area. The confidence of the pixel is estimated as:

$$C_{\psi_p}^{w}(p) = \frac{\sum_{q \in \psi_p \cap \phi_w} C_{\psi_p}^{w}(q)}{|\psi_p|}, \qquad 0 \le C_{\psi_p}^{w} \le 1 \tag{14}$$

where $C_{\psi_p}^{w}(p)$ is the confidence of the already filled pixel q.



(a) (b)

Fig. 6. Effects of confidence and data terms, modified from [21]. (a) Data terms and (b) confidence.

In the proposed algorithm, the confidence of the source region (ϕ_w) pixels is initially set to 1 and the gap region (Ω_w) to 0. Once the boundary pixel patch ψ_p is filled, the newly filled pixels of the target region of $\psi_{p\{\forall p\in\Omega_w\}}$ will be assigned the same confidence $C^w_{\psi_p}(p)$ of the parent pixel.

4.2.4 The priorities

In the proposed algorithm the priority estimation is based on geometric aspects in the patch and the isophote energy of the wavelet coefficients in the patch. It was observed in our implementation of the algorithm by Criminisi et al. [21] that as the algorithm propagates deeper into the target region, the confidence values drop exponentially. This makes the computed priorities undistinguishable and leads to incorrect filling order as filling progresses deeper into the target region. Even if the data values increase, the shape of the priority curve is dominated by exponential falling confidence values, which leads to artifacts. The effect of this is more noticeable while filling a large target region. This intense dominance is due to the effect of multiplicative priorities proposed in [21]. Thus, we proposed the additive form of priority estimation as Eq. (15), which is more linear and stable to unexpected changes. Eq. (15) also adds the patch variational energy to strengthen the structure propagation. The pixel priorities are estimated as:

$$P(p) = C^w_{\psi_p}(p) + D^w_{\psi_p}(p) + E^w_{\psi_p}(p) \tag{15}$$

The direct combination of $C^w_{\psi_p}(p)$ and $D^w_{\psi_p}(p)$ is still unreasonable due to a significant difference in their values. As the algorithm propagates, the falling rate of $C^w_{\psi_p}(p)$ is significant due its exponential nature, as shown in Fig. 7. Therefore, it needs to be regularized as:

$$RC^w_{\psi_p}(p) = (1-\omega) \times C^w_{\psi_p}(p) + \omega, \quad 0 \le \omega \le 1. \tag{16}$$

where, ω is a regularsing factor.

The additional weights are incorporated to control the contribution of $C^w_{\psi_p}(p)$, $D^w_{\psi_p}(p)$ and $E^w_{\psi_p}(p)$ in the estimation of priorities for various types of images as shown below:

$$P(p) = \alpha \times RC^w_{\psi_p}(p) + \beta \times D^w_{\psi_p}(p) + \gamma E^w_{\psi_p}(p) \tag{17}$$

where,

$$0 \le \alpha, \beta, \gamma \le 1 \ \ \& \ \ \alpha + \beta = 1$$

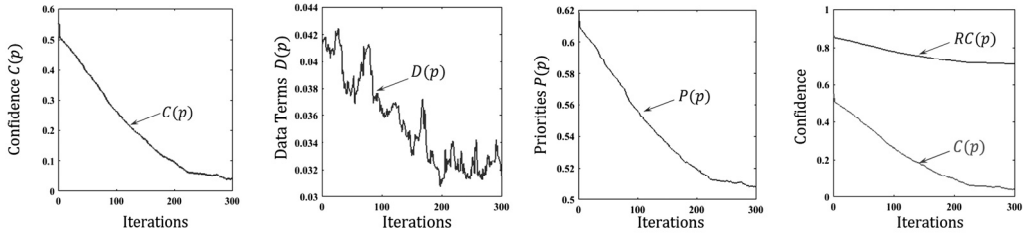
We typically used $\alpha = 0.7, \beta = 0.3$ & $\gamma = 0.5$ in Eq. (16) for test results.
Finally, the first highest priority block is decided by:

$$p \Leftrightarrow \arg\max_{p \in \partial\Omega_w}\{P(p)\} \tag{18}$$

where, $p \in \partial\Omega_w$ indicates the index of the highest priority patch @ p in the image.

4.3 Performance Improvement

It is also obvious that the best matching patch will be mostly found in the close vicinity of the target patch. The complete source area search method in [21] has been found to have a large performance overhead. Thus, we are proposing that the search area A_s is bound around the target patch ψ_p either by the n × m pixel or by empirical equations proposed in our earlier work [34] to improve the performance of the algorithm. In our current work we are using $A_s = K \times N \times N$, where K is a search area factor to limit the search area and $N \times N$ is the size of the patch.



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

Fig. 7. Average fall of (a) $C^w_{\psi_p}$, (b) $D^w_{\psi_p}$, (c) priorities, and (d) regularized $RC^w_{\psi_p}$.

4.4 Patch Optimization

It has been observed that different kinds of images have varying texture and structure complexions Therefore, the fixed patch size may not provide equally good results. In complex images, the patch size needs to be varied in accordance to the structural and textural complexities around the target area. Manually changing the patch size on every iteration is a difficult task. Thus, we are proposing the automatic patch size optimization in the algorithm. The image values of the highest priority patch corresponding to index patch ψ_p are given as:

$$\hat{\boldsymbol{I}}_{\boldsymbol{p}_{(i,j)\{\boldsymbol{p}\in\partial\Omega_w,\forall i,\forall j\in\widetilde{\psi}_p\cap\phi_w\}}} \tag{19}$$

The algorithm automatically decides the optimum patch size by testing the quality of the filling source patch $\hat{\boldsymbol{I}}_{\boldsymbol{q}}$ with $\hat{\boldsymbol{I}}_{\boldsymbol{p}}$, which is done by keeping the minimum sum squared error (SSE) patch. The SSE is estimated only for the undamaged portion of the target patch by varying the patch size from 3×3 to 15×15 pixels. At each iteration, after the highest priority index patch ψ_p with a minimum size is selected as the filling target, the best matching patch ψ_q is searched in the bounded area in the neighborhood of ψ_p as the source patch. The highest priority index patch, $\hat{\psi}_{p\{p\in\partial\Omega_w\}}$ is constructed with the initial patch size of $\underline{sz}$.

The image patch $\hat{I}_{q\{\boldsymbol{q}\in\phi_w\cap A_s\}}$ corresponding to index patch ψ_q is searched in the bounded area such that:

$$\hat{I}_q = \arg\min_{sz\in[3,15]}\left[d(\hat{\psi}_{p\{p\in\partial\Omega_w\}}, \hat{\psi}_{q\{\boldsymbol{q}\in\phi_w\cap A_s\}})\right] \tag{20}$$

where, $\hat{I}_p$, $\hat{I}_q$ are the image values corresponding to the index patches $\hat{\psi}_p$ and $\hat{\psi}_q$.

The estimation of distance $d(\hat{\psi}_{p\{p\in\partial\Omega_w\}}, \hat{\psi}_{q\{\boldsymbol{q}\in\phi_w\cap A_s\}})$ is given by:

$$d(,) = \left[\frac{\sum_{\forall i,j}\left(\hat{I}_{p(i,j)\{\forall i,j\in\phi_w\cap A_S\}} - \hat{I}_{q(i,j)\{\forall i,j\in\phi_w\cap A_S\}}\right)^2}{\#\psi_{p\{\forall p\in\phi_w\cap\psi_p\}}}\right]_{\underline{sz}}^{\overline{sz}} \tag{21}$$

For patch optimization, the distance metric $d(,)$ is estimated over the LH, LL, and HH subband portion of every best matching patch $\hat{\psi}_{q\in\phi_w} \Leftrightarrow \hat{\psi}_p$ in the bounded area by computing the SSE with the automatic variation of a patch size from $\underline{sz} = 3 \times 3$ to $\overline{sz} = 15 \times 15$) by Eqs. (20) and (21). Since a bigger patch size leads to higher SSE, the normalization is done using number of patch pixels $\#\psi_{p\{p\in\phi\cap\psi_p\}}$ involved in calculating Eq. (21). It is possible that multiple best matching patches $\hat{I}_q$ are found with the same distance $d(,)$. In that case we computed the variance with respect to the mean of $\hat{I}_{p\in\hat{\psi}_p\cap\phi_w}$ as:

$$V = \frac{\sum_{i,j}\left(\hat{I}_{q(i,j)\{\boldsymbol{q}\in\phi_w-\psi_p,\forall i,\forall j\in\hat{\psi}_q\cap\phi_w\}} - M\right)^2}{\#\psi_{p\{p\in\phi_w\cap\psi_p\}}} \tag{22}$$

$$M = \frac{\sum_{i,j}\left(\hat{I}_{p(i,j)\{\boldsymbol{p}\in\partial\Omega_w,\forall i,\forall j\in\hat{\psi}_p\cap\phi_w\}}\right)}{\#\psi_{p\{p\in\phi_w\cap\psi_p\}}} \tag{23}$$

where, $\#\psi_p$ is the cardinality of the set.

4.5 Proposed Algorithm Pseudo Code

- Extract programmatically the initial fill front $\boldsymbol{\partial\Omega_w}$.
- Repeat{

  For each level j {For (LH, HL, HH){

  1) Identify the fill front, $\boldsymbol{\partial\Omega_w^t}$. If $\boldsymbol{\Omega_w^t} = \boldsymbol{0}$, exit.
  2) Compute priorities using Eq. (17).

     For ($\underline{sz}$ to $\overline{sz}$){
  3) Find patch, $\boldsymbol{\psi_{p\in\partial\Omega_w}}$, with the maximum priority.
  4) Search the bounded area, $A_s \in \phi_w$, for matching the patch, $\hat{\boldsymbol{I}}_{\boldsymbol{q}} \Leftrightarrow \boldsymbol{\psi_p}$
  5) Apply patch automation optimization, Eq. (20,21).

     }// Repeat steps (3-5) iteratively to find best sized patch, $\hat{\boldsymbol{I}}_{\boldsymbol{q}}$.
  6) Copy image data from optimized best matching source patch, $\hat{I}_q$ corresponds to $\psi_{p\in\Omega}$.
  7) Update Confidence of newly filled pixels
  8) Update the fill status of newly filled pixels.

     }// Repeat steps 1 to 7.

}}}// Repeat step (1-8) until all pixels are filled

9) Repeat steps (1-8) for LL subband.

} until all the pixels are filled.

10) Reconstruct final image using IDWT

where 't' indicates the iterator.

4.6 Implementation Details

The program will start with masked and degraded images. The mask image is prepared by the user with any photo editing tool (e.g., MS Paint). It contains the object to be inpainted, marked with some color. The isophote computation is done by the image gradients with a 90° gradient rotation and it is normalized with α as:

$$\nabla I_p^{\perp}(x,y) = \frac{\nabla I_p \; rotatedby \; 90^0}{\alpha} \tag{24}$$

α is a normalization factor (e.g., 255 for a typical grey level image and 255 × 3 for an RGB image). The contour of the target region is found by convolving the target region with a Laplacian of Gaussian (LoG) edge detector. The fill-front normals are calculated as normalized gradients at target boundary pixels by using a binary mask image. The computation of the data term for each pixel on the boundary is carried out by using Eq. (12).

The confidence of each pixel is calculated by constructing a patch ψ_p around that pixel on the fill-front. Then, the array of pixel indices $\psi_p | p \in \partial\Omega_w \cap \phi_w$ is found. The confidence of the pixel is determined as:

$$C_{\psi_p}^{w} = \frac{Number\ of\ Pixels\ already\ filled\ in\ \psi_p}{Total\ pixels\ in\ \psi_p} \tag{25}$$

Using $C_{\psi_p}^{w}(p), D_{\psi_p}^{w}(p)$ and $E_{\psi_p}^{w}(p)$, the patch priority $P(p)$ for boundary pixel is computed using Eq. (17). The typical values of $\alpha = 0.2$, $\beta = 0.8$ & $\omega = 0.7$ $\gamma = 0.5$ are used as initial values.

The maximum priority pixel on the fill-front is found and the index patch is obtained around it as ψ_p, as shown in Fig. 5(b). Part of $\psi_{p\in\Omega_w}$ in Ω_w is to be filled at the first iteration of the filling process. The global search is performed in $A_s \in \phi_w$ for the best matching patch ψ_q, as shown in Fig. 5(c). Only the pixels of ψ_p corresponding to the target region will be filled from $\psi_{q\in\phi_w}$ in Fig. 5(d). The fill region status of newly filled pixels is then updated. The confidence values of newly filled pixels are updated to their parent pixels. If the newly filled region contains the isophote, this is to be considered in the next iteration of the proposed algorithm. The algorithm is iterated until the entire fill region is covered in all DWT subbands. The LL approximation is processed separately with the same algorithm without variational patch energy. The final image is reconstructed from all {j} level results using an inverse DWT. All experiments were run on a 3.2 GHz Intel I7 processor with 8 GB of RAM.

5. The Metric Used for Estimating Comparative Quality

Peak signal to noise ratio (PSNR):

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I_1(i,j) - I_2(i,j)]^2 \tag{26}$$

$$MAX_I = \max\left[\max_{\forall i,j}(|I_1|), \max_{\forall i,j}(|I_2|)\right] \tag{27}$$

$$PSNR = 10.\log_{10}\left[\frac{MAX_I^2}{MSE}\right] \tag{28}$$

where, I_1 is the original image and I_2 is either the degraded or inpainted image.

6. Results

Fig. 1 shows the large object removal process. Figs. 8 and 9 shows comparative image results of our proposed algorithm over the algorithm proposed in [21] and [39] respectively. The test results of the proposed algorithm for a variety of natural are shown in Fig. 10. The corresponding quality measures are listed in Table 1, which also lists the PSNR-D of the degraded image and the PSNR Cri of our implementation of Criminisi et al. [21] algorithm. The last column of Table 1 lists the results of our proposed algorithm. The comparative trend of our proposed algorithm and the results of the algorithm in [21] are shown in Figs. 11 and 12 shows the visual results of our proposed algorithm for various textured images. All of the textured images were taken from the STex Salzburg Texture Image Database.

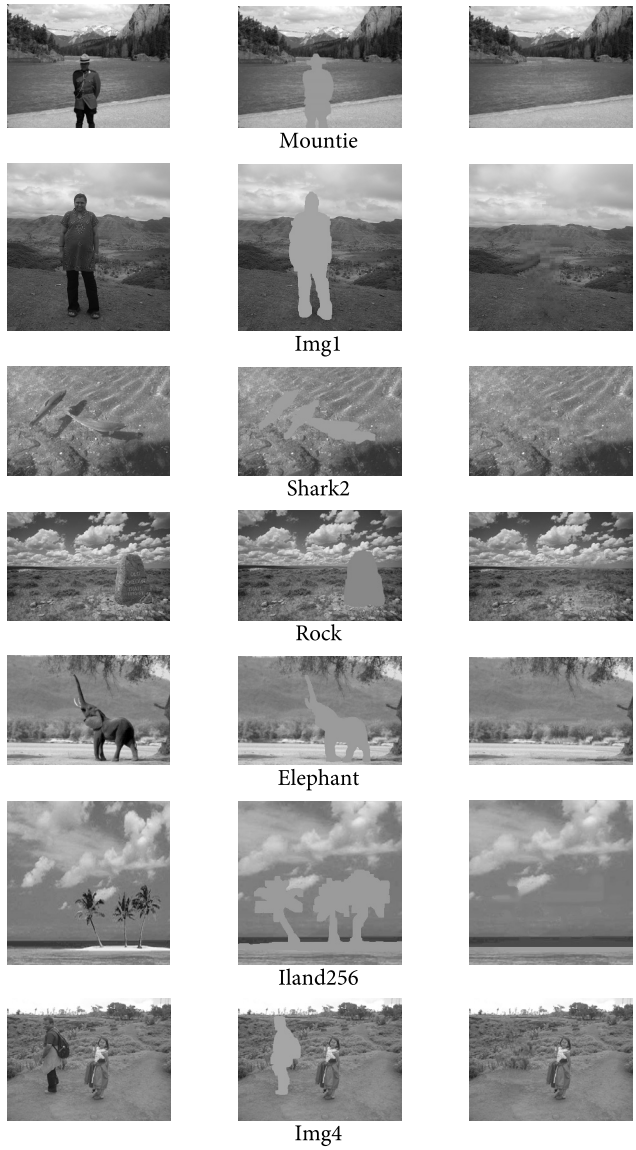
Table 1 lists the quality measures for the textured images in Fig. 12. It also lists the PSNR-D for the degraded image, the PSNR Cri for the algorithm in [21], and the PSNR PBI_DWT for our proposed algorithm. The graphical comparison is shown in Fig. 13 for the textured images in Table 2. It was observed that the proposed algorithm achieves a quality improvement of 5.16 dB to 12.1 dB in relation to degraded images in the case of natural images and from 7.41 dB to 16.25 dB in the case of textured images. The proposed algorithm achieved a comparative betterment of up to 3.28 dB and up to 3.81 dB over the algorithm in [21] in the cases of natural and textured images, respectively.

(a) (b) (c)

Fig. 8. (a) Original images, (b) images by method [21], and (c) present result

(a) (b) (c)

Fig. 9. (a) original images, (b) images by method [39], and (c) present result.



(a) (b) (c)

Fig. 10. Natural images. (a) original, (b) mask, and (c) inpainted.

Table 1. PSNR results in (dB) of, proposed algorithm & [21], for natural images

Name	PSNR-D	PSNR Cri	PSNR PBI_DWT
Mountie	4.13	8.59	9.29
Elephant	3.18	9.28	10.01
Img1	3.49	12.11	12.93
ILand256	2.86	11.59	14.88
Shark2	4.21	12.78	13.49
Img4	3.85	12.38	13.46
Rock	4.73	15.19	16.46

PSNR=peak signal to noise ratio.

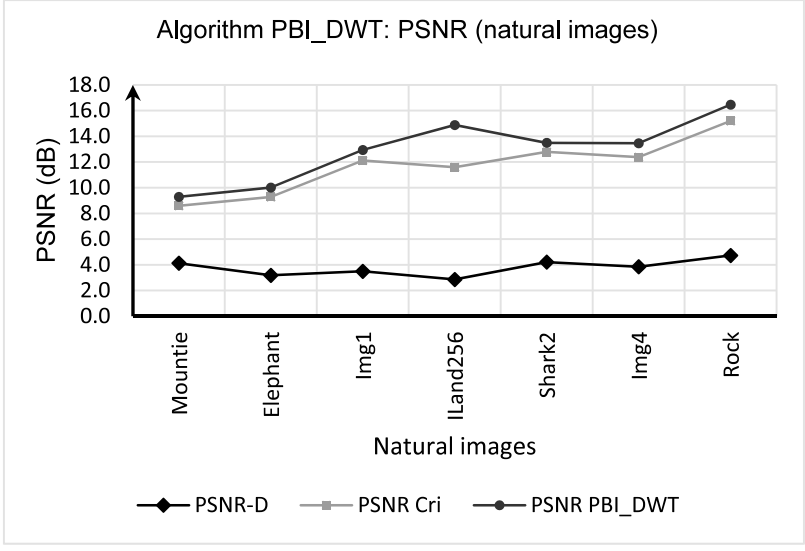


Fig. 11. Comparative peak signal to noise ratio (PSNR) results natural images, of PBI_DWT over algorithm of [21].

Table 2. PSNR results in (dB) of, proposed algorithm & [21], for textured images

Sr	Name	PSNR-D	PSNR Cri [21]	PSNR PBI_DWT
1	Bush2	4.66	11.44	13.31
2	Bush5	4.30	11.62	14.12
3	Bush0	3.55	12.43	14.62
4	Bark	6.50	13.46	13.91
5	Ratten2	7.30	16.76	18.33
6	Wood40	8.50	18.88	20.58
7	Fabric15	10.65	20.62	24.43
8	Fabric2	6.78	21.71	23.03

PSNR=peak signal to noise ratio.

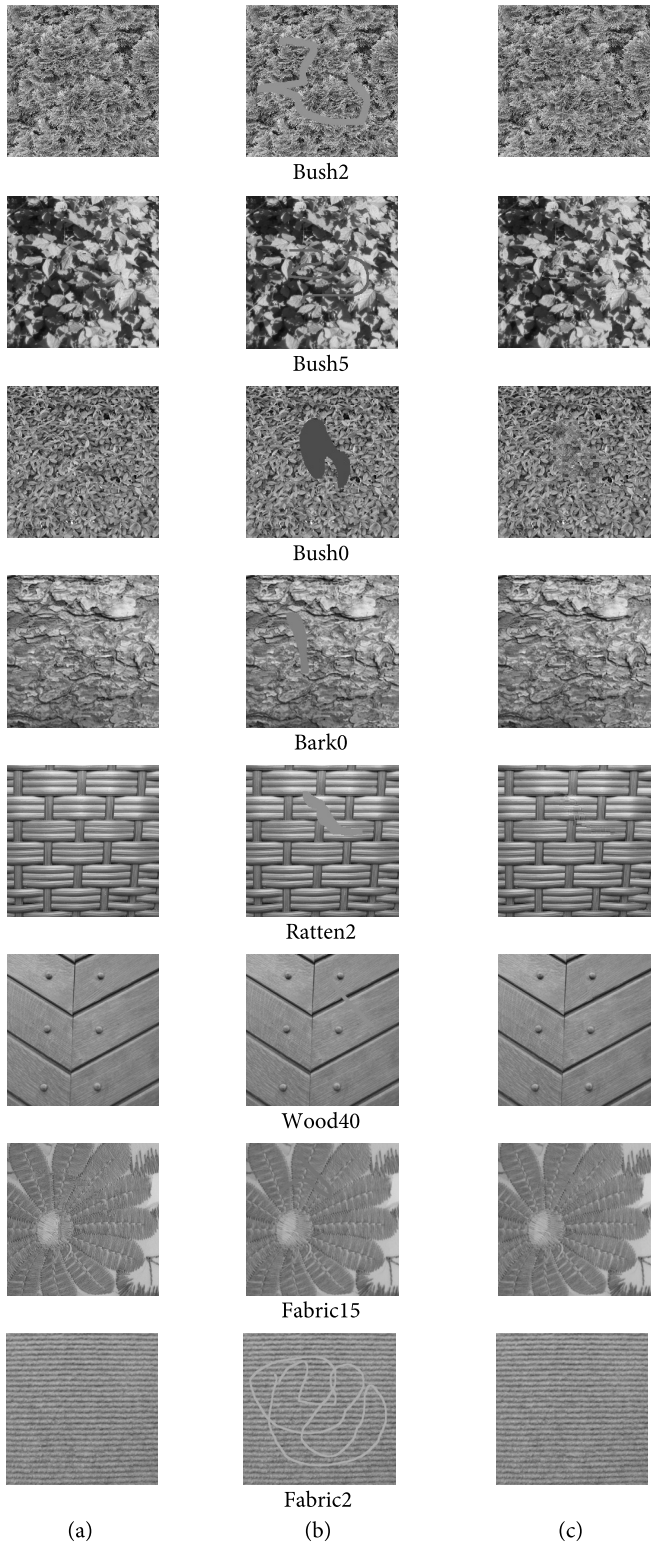


Fig. 12. Texture reconstruction (Salzburg Texture Image Database). (a) Original, (b) mask, and (c) inpainted images.

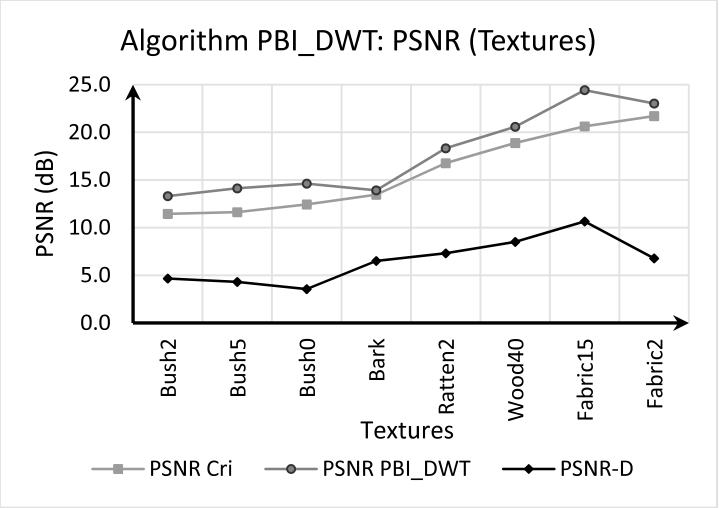


Fig. 13. Comparative peak signal to noise ratio (PSNR) results textures, of PBI_DWT over algorithm of [21].

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'00)*, New Orleans, LA, 2000, pp. 417-424.

[2] C. Ballester, V. Caselles, J. Verdera, M. Bertalmio, and G. Sapiro, "A variational model for filling-in gray level and color images," in *Proceedings of 8th IEEE International Conference on Computer Vision (ICCV2001)*, Vancouver, Canada, 2001, pp. 10-16.

[3] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation,* vol. 12, no. 4, pp. 436-449, 2001.

[4] G. Kanizsa, *Organization in Vision: Essays on Gestalt Perception*. New York, NY: Praeger, 1979.

[5] L. Pessoa, E. Thompson, and A. Noe, "Finding out about filling-in: a guide to perceptual completion for visual science and the philosophy of perception," *Behavioral and Brain Sciences*, vol. 24, no. 6, pp. 723-802, 1998.

[6] S. Sarkar and K. L. Boyer, "Integration, inference, and management of spatial information using Bayesian networks: perceptual organization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 3, pp. 256-274, 1993.

[7] D. Tschumperle and R. Deriche, "Vector valued image regularization with PDEs: a common framework for different applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 506-517, 2005.

[8] S. Masnou and J. M. Morel, "Level lines based disocclusion," in *Proceeding of International Conference on Image Processing (ICIP 98)*, Chicago, IL, 1998, pp. 259-263.

[9] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. Rayner, "Interpolation of missing data in image sequences," *IEEE Transactions on Image Processing,* vol. 4, no. 11, pp. 1509-1519, 1995.

[10] M. Bertalmio, L. A. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2003)*, Madison WI, 2003, pp. 707-712.

[11] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment based image completion," *ACM Transactions on Graphics,* vol. 22, no. 3, pp. 303-312, 2003.

[12] P. Harrison, "A non-hierarchical procedure for re-synthesis of complex texture," in *Proceeding of the 9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG2001)*, Plzen, Czech Republic, 2001, pp. 190-197.

[13] R. Bornard, E. Lecan, L. Laborelli, and J. H. Chenot, "Missing data correction in still images and image sequences," in *Proceedings of the 10th ACM International Conference on Multimedia*, Juan les Pins, France, 2002, pp. 355-361.

[14] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proceedings of 24th annual ACM Conference Computer Graphics and Interactive Techniques (SIGGRAPH)*, Los Angeles, CA, 1997, pp. 361-368.

[15] D. Garber, "Computational models for texture analysis and texture synthesis," Ph.D. dissertation, University of Southern California, Los Angeles, CA, 1981.

[16] D. J. Heeger and J. R. Bergen, "Pyramid based texture analysis/synthesis," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, 1995, pp. 229-238.

[17] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceeding of the 28th ACM Annual Conference on Computer Graphics*, Los Angeles, CA, 2001, pp. 341-346.

[18] L. Liang, C. Liu, Y. Q. Xu, B. Guo, and H. Y. Shum, "Real-time texture synthesis by patch based sampling," *ACM Transactions on Graphics,* vol. 20, no. 3, pp. 127-150, 2001 .

[19] M. Ashikhmin, "Synthesizing natural textures," in *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, Research Triangle Park, NC, 2001, pp. 217-226.

[20] J. Wu and Q. Ruan, "Object removal by cross isophotes exemplar based inpainting," in *Proceeding of 18th International Conference on Pattern Recognition (ICPR2006)*, Hong Kong, 2006, pp. 810-813.

[21] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar based image inpainting," *IEEE Transactions on Image Processing,* vol. 13, no. 9, pp. 1200-1212, 2004.

[22] Y. Chen, Q. Luan, H. Li, and O. Au, "Sketch guided texture based image inpainting," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Atlanta, GA, 2006, pp. 1997-2000.

[23] J. Sun, L. Yuan, J. Jia, and H. Y. Shum, "Image compilation with structure propagation," *ACM Transactions on Graphics,* vol. 24, no. 3, pp. 861-868, 2005.

[24] T. F. Chan, J. Shen, and H. M. Zhou, "Total variation wavelet inpainting," *Journal of Mathematical Imaging and Vision,* vol. 25, no. 1, pp. 107-125, 2006.

[25] S. D. Rane, J. Remus, and G. Sapiro, "Wavelet domain reconstruction of lost blocks in wireless image transmission and packet switched networks," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Rochester, NY, 2002, pp. 309-312.

[26] K. Patwardhan and G. Sapiro, "Projection based image and video inpainting using wavelets," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, 2003, pp. 857-860.

[27] D. Cho and T. D. Bui, "Image inpainting using wavelet based inter- and intra-scale dependency," in *Proceedings of 19th International Conference on Pattern Recognition (ICPR)*, Tampa, FL, 2008, pp. 1-4.

[28] G. Hua and M. Orchard, "Image inpainting based on geometrical modeling of complexwavelet coefficients," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, San Antonio, TX, 2007, pp. 553-556.

[29] F. Wang, D. Liang, N. Wang, Z. Cheng, and J. Tang, "An new method for image inpainting using wavelets," in *Proceeding of International Conference on Multimedia Technology (ICMT)*, Hangzhou, China, 2011, pp. 201-204.

[30] Y. W. Wen, R. H. Chan, and A. M. Yip, "A primal–dual method for total-variation-based wavelet domain inpainting," *IEEE Transactions on Image Processing,* vol. 21, no. 1, pp. 106-114, 2012.

[31] L. Chen, X. Chen, and P. Guo, "Blind image restoration using divisional regularization and wavelet technique," in *Proceeding of 4th International Conference on Natural Computation (ICNC'08)*, Jinan, China, 2008, pp. 476-480.

[32] R. H. Chan, Y. W. Wen, and A. M. Yip, "A fast optimization transfer algorithm for image inpainting in wavelet domains," *IEEE Transactions on Image Processing,* vol. 18, no. 7, pp. 1467-1476, 2009.

[33] J. C. Hung, C. H. Hwang, Y. C. Liao, N. C. Tang, and T. J. Chen, "Exemplar-based image inpainting base on structure construction," *Journal of Software,* vol. 3, no. 8, pp. 57-64, 2008.

[34] R. P. Borole and S. V. Bonde, "Image analysis & restoration by exemplar inpainting," in *Proceedings of National Conference on Communication, Computing and Networking (NCCCNT'13)*, Nanded, India, 2013.

[35] R. P. Borole and S. V. Bonde, "Patch-based inpainting for object removal and region filling in images," *Journal of Intelligent Systems,* vol. 22, no. 3, pp. 335-350, 2013.

[36] R. C. Gonzalez and R. E. Wood, *Digital Image Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[37] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 674-693, 1989.

[38] A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceeding of the 7th IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1033-1038.

[39] D. Sun, L. Yuan, Y. Zhang, J. Zhang, and G. Pan, "Structure-aware image completion with texture propagation," in *Proceeding of 6th International Conference on Image and Graphics (ICIG)*, Hefei, China, 2011, pp. 199-204.

Rajesh P. Borole http://orcid.org/0000-0001-9812-2505

He received B.E. in Electronics and Telecomm Engineering in 1989, M.Tech in Electronics Design Technology in 1995 from CEDT, Aurangabad, Department of Electronics and Telecom Govt. of India. He received Ph.D. from SGGSI&T, Nanded, SRTM University. After worked in Project and Development in Process Automation and Control in various industries for 9 Years, he shifted to Education and Research for 15 years in various engineering Institutes and worked in the position of Head of Dept. E&TC. Currently he is working with Dr. D. Y. Patil College of Engineering, Pune in the Department of E&TC as Head and research scholar. He is working as technical consultant in Industrial Automation and System networking. He is a life member of ISTE, India.

Sanjiv V. Bonde

He received the B.E. and M.E. degrees in electronics engineering from Marathwada University, Aurangabad, India, in 1988 and 1994, respectively. Since 1988, He has been a faculty member of electronics engineering department at S.G.G.S. Institute of Engineering and Technology, Nanded, India. He obtained Ph.D. degree from School of Biosciences and Bioengineering, Indian Institute of Technology-Bombay, Mumbai, India. He is having 16 publications in conferences and journals. His research interests include signal detection, estimation, and image processing.