IJST, Transactions of Electrical Engineering, Vol. 35, No. E2, pp 95-108
Printed in The Islamic Republic of Iran, 2011

A BILATERAL IMAGE INPAINTING*

H. NOORI,** S. SARYAZDI AND H. NEZAMABADI-POUR

S-building, Shahid Bahonar University of Kerman, Kerman, I. R. of Iran
Email: hossein.noori2010@yahoo.com

Abstract– In this paper a new image inpainting scheme using bilateral filters is proposed. Since in the digital image inpainting pixels' values in a damaged region are unknown and are needed in calculation of weights according to the range filter of bilateral filters, in the proposed scheme we substitute the difference between two gray level values in the range filter by multiplication of two vectors: direction between two pixels and gradient direction of known pixel in the neighborhood of damaged pixels. The algorithm is iterative, fast and simple to implement. In addition, to achieve a better performance, the number of iterations is adaptively determined according to the region type (structural or textural) using the local variance. To evaluate the performance of the proposed method, several comparative experiments are performed. Experimental results confirm the effectiveness of the proposed algorithm.

Keywords– Image inpainting, image interpolating, image reconstruction, bilateral filtering

1. INTRODUCTION

Image inpainting is the process of filling-in missing regions in an image in such a way that the inpainted region cannot be detected by an observer who does not know the original image. "*Inpainting*" is an artistic synonym for image interpolation, and has been used among museum restoration artists for a long time. The notion of digital inpainting was first introduced by Bertalmio et al. [1]. The object of digital inpainting is to reconstitute the missing or damaged areas of the image in order to make it more legible and to respect its unity. Inpainting techniques have broad applications such as photo restoration, zooming, special effects and super-resolution; primal-sketch based perceptual image compression and coding, and the error concealment.

To recover the structural and textural contents in a damaged area, inpainted (output) pixels must be calculated using the surrounding undamaged areas data. Most image areas consist of both texture (regions with high variation like Barbara's trousers) and structure regions (smooth areas with boundaries like a cartoon image). Usually, inpainting techniques focus only on texture or structure restoration. A textural inpainting technique produces good results in textural regions, while failing to reconstruct missing parts in structural regions. In contrast, structural inpainting techniques achieve good results in structural regions but weak results in textural regions.

Bertalmio et al. [1] proposed a digital image inpainting algorithm based on partial differential equations (PDEs). They fill in the damaged areas to be inpainted by propagating information from the surrounding undamaged region along level lines (isophotes). Au and Takei [2] described an inpainting scheme by providing a numerical solution to the Navier-Stokes equations. In [3] the author presented a new approach based on neural networks to recover degraded images. In that paper two learning features have been presented which are very effective in implementation and result in good speed. Ballester et al.

*Received by the editors November 28, 2010; Accepted December 13, 2011.
**Corresponding author

investigated the use of a formal variational formulation for the simultaneous propagation of edges and gray-scale values [4]. Another important PDE based algorithm is the total variation (TV) algorithm of Chan and Shen [5]. The total variation (TV) inpainting model has used an Euler-Lagrange equation and inside the inpainting domain the model simply employs anisotropic diffusion based on the contrast of isophotes. The total variation inpainting model is to find a function *u* such that the following functional is minimized.

$$TV[u] = \int_{\Omega} |\nabla u| dxdy \tag{1}$$

Where Ω is damaged domain and ∇u denotes the gradient of image u. By using the Euler-Lagrange equation the energy functional is as follows:

$$\frac{\partial u}{\partial t} = \nabla \cdot (\frac{\nabla u}{|\nabla u|}) + \lambda_e (u^0 - u) \tag{2}$$

Where λ_e is the extended Lagrange multiplier, u^0 is the original damaged image, t shows time step and $|\nabla u|$ denotes norm2 of the gradient of image u. In this approach only range filter -a filter that operates on the gray level of pixels- is considered. It considers only 8-neighborhood of damaged pixels, and needs a large number of iterations, while the method presented in this article considers both range filter and domain filter –a filter that operates on the distance between pixels; it also considers a w×w (w is user defined number) neighborhood around damaged pixels, and needs few iterations, for example, only 2 or 3 iterations. This model was designed for inpainting small regions and has good result in image reconstruction and noise reduction as well as being able to connect broken edges.

Chan and Shen have two other similar works in this topic, the curvature-driving diffusion (CDD) algorithm and the Euler elastica and curvature based inpainting model [6, 7]. The CDD model enhances the TV method by driving diffusion process to correct the isophotes' directions and thus allow inpainting of thicker regions. In [6], the authors introduced a new variational image inpainting model that dealt with the problems of total variation based approach, in which the authors tried to minimize the following energy equation.

$$E_2[\gamma] = \int_{\gamma} (a + bk^2) ds \tag{3}$$

where γ shows an elastic curve, a and b are two constants, ds is arc length element and k is the scalar curvature that is defined as follows:

$$k = \nabla \cdot (\frac{\nabla u}{\nabla u}) \tag{4}$$

where $\nabla \cdot$ denotes divergence operator.

Their approach includes a regularization term that penalizes not only the length of edge in an image, but the integral of the square of curvature along the edge contours. This allows both for isophotes to be connected across large distances, and their directions to be kept continuous across the edge of the inpainting region. The major drawback of this algorithm is that it suffers from over-smoothing effect, especially when the regions to be filled are thick.

PDE methods require a difficult implementation process- because of complex equations and their discretion- and nontrivial iterative numerical methods such as anisotropic diffusion schemes. On the other hand, PDE based algorithms are designed to connect edges or to extend level lines into the damaged area [8-13]. Most of them produce disturbing artifacts if the damaged area is surrounded by a textured region.

Telea[14] proposed a fast marching algorithm that can be seen as the PDE based approach without the computational overhead. It is considerably faster and simpler to implement than other PDE based methods and produces very similar results comparable to other PDE methods. Oliveira et al. [15] proposed a fast inpainting method which depends on the convolution operation. Convolving an image with a Gaussian kernel is equivalent to isotropic diffusion (linear heat equation). They used a weighted average kernel that only considers contributions from the neighboring pixels, and fills the damaged area by convolving repeatedly the region to be inpainted with a diffusion template. The number of convolution iterations is independently controlled by a certain threshold on the change of pixel value from the previous iteration or by the user. A modification to the Oliveira inpainting model is introduced in [16]. This modification reduces the time of inpainting and increases the quality of the results based on the modification of convolution stage. In [17] Bornemann and Marz proposed a fast inpainting algorithm based on coherence transport. They tried to combine the celebrated method of Bertalmio [1] and linear, fast, non-iterative inpainting method of Telea [14]. In fact, they propose modifying the weight function of Telea's algorithm to obtain the formal stationary state of Bertalmio, and the edge-oriented transport direction of Bertalmio by coherence direction is substituted to increase the robustness of their approach. In this way, the authors follow Weickert's [18] approach of using the structure tensor to the robust determination of coherence directions in images. In [17], the authors proposed the following weight function.

$$w(p,q) = \sqrt{\frac{\pi}{2}} \frac{\mu}{|p-q|} \cdot \exp\left(-\frac{\mu^2}{2\varepsilon^2} \left| \overrightarrow{c^{\perp}}(p) \cdot (p-q) \right|^2\right) \tag{5}$$

In the above equation, *p,q* are two vectors that show the location of the damaged pixel and a known pixel in its neighborhood, respectively. ε is a user defined parameter and *c* is coherence direction and is computed as follows:

$$\vec{c} = \text{normalized eigenvector to the minimal eigenvalue of } J_{\rho}(\nabla u_{\sigma}) \tag{6}$$

$$J_{\rho}(\nabla u_{\sigma}) = K_{\rho} * (\nabla u_{\sigma} \otimes \nabla u_{\sigma}), \qquad u_{\sigma} = K_{\sigma} * u \tag{7}$$

Where K_{σ} and K_{ρ} is the Gaussian kernel with variance σ and ρ, respectively, and * shows convolution operation. μ in equation (5) can be calculated as follows:

$$\mu(p) = \begin{cases} 1 & if\ \lambda_1(p) = \lambda_2(p) \\ 1 + \kappa \cdot \exp\left(\dfrac{-\delta_{quant}^4}{(\lambda_2(p) - \lambda_1(p))^2}\right) & otherwise \end{cases} \tag{8}$$

In that κ is an arbitrary parameter, δ_{quant} is taken as resolution of the quantization and λs are the eigenvalues of J_{ρ}. In the above equations u shows the missed image and exp(.) denotes the exponential function. This algorithm is very similar to the proposed algorithm. But the algorithm in [17] uses inverse of distance as the domain filter, in the range filter it adopted coherence transport as the edge-oriented transport of Bertalmio, and it uses the fast marching method of Telea[14] to propagate information in the missed areas as well. However, in the proposed algorithm a Gaussian function is considered as domain filter, orthogonal gradient of the image is used as the edge-oriented transport of Bertalmio, beginning from the boundary, and after restoring the damaged boundary the missed regions are shrunk.

All previously mentioned methods produce disturbing artifacts if the damaged area is surrounded by a textured region because they cannot model the textural regions in an image and they only try to connect damaged edges and structures. A very simple and highly effective algorithm for textured region inpainting

was presented by Efros and Leung [19]. They modeled the image as a Markov Random Field and synthesized texture in a pixel by pixel way, by picking existing pixels with similar neighborhoods. This algorithm performs well, but is very slow because of one pixel by pixel inpainting. In [20], Efros improved the speed by using a similar and simpler algorithm, which filled in the image in a block by block of pixels way.

An extension of this algorithm was presented by Criminisi et al. [21]. In this algorithm the pixels that are placed along the edges are filled in with high priority.

Since most images contain both structure and texture regions, some methods combine structure and texture inpainting techniques to obtain more effective results. Bertalmio et al. [9] decompose the image into structure and texture sub-images. Then inpainting techniques are separately applied into both structure and texture sub-images. For structure-texture decomposition a variational method may be used. Wong and Orchard [22] proposed an exemplar-based inpainting technique inspired from nonlocal-means image denoising technique [23]. They used nonlocal image information from multiple samples within the image.

In brief, PDE based inpainting algorithms preserve structures, but they are slow and inadequate for texture inpainting. Texture based inpainting techniques are faster and are suitable for texture inpainting, but in structured regions they do not provide adequate results. Combined techniques, however, achieve very satisfactory results because of the decomposing process, and applying both structure and texture inpainting algorithms is time consuming. Convolution based inpainting methods are the most rapid inpainting techniques but suffer from low quality of the inpainted image. To overcome this drawback, in this paper a new inpainting algorithm is proposed by convolving the damaged image with a space variant kernel. At each pixel, the kernel is calculated using its neighbors in both space and range domain. This idea uses bilateral filter for digital image inpainting as this filter utilizes both range and domain filters. Since the bilateral filters are very efficient in image denoising, and both image denoising and inpainting use the same principle (estimating the damaged (lost) pixel value by its neighbors), a high performance can be expected.

The organization of the paper is as follows. The next section is devoted to a brief review of bilateral filtering. The proposed inpainting algorithm is described in section 3. To evaluate the performance of the proposed method, a comparative study with some well known methods is given in section 4. Finally, in section 5 a conclusion is given.

2. BILATERAL FILTERING

The bilateral filter was originally proposed by Tomasi and Manduchi in 1998 [24] as a heuristic tool for noise removal. A similar filter (Digital-TV) was proposed by Chan, Osher and Shen in 2001[25]. A bilateral filter is an edge-preserving smoothing filter. Whereas many filters are convolutions in the image domain, a bilateral filter also operates in the image's range (pixel values or gray levels domain). The bilateral filter replaces a pixel's value by a weighted average of its neighbors in both space and range. This is applied as two Gaussian filters at a localized pixel neighborhood, one in the spatial domain, named the *domain filter*, and the other in the intensity domain, named the *range filter*. This approach permits preserving sharp edges. Every sample is replaced by a weighted average of its neighbors:

$$I_f(p) = \frac{\sum_q w(p,q) I(q)}{\sum_q w(p,q)} \tag{9}$$

Where, I and I_f represent the input and filtered signal, respectively. p is current pixel (position) and q is a neighbor pixel and $w(p,q)$ is a weight devoted to pixel q. Furthermore,

$$w(p,q) = w_1(p,q) \times w_2(p,q) \tag{10}$$

With two weighing functions:

$$w_1(p,q) = e^{-\frac{1}{2}(\frac{d_s}{\sigma_d})^2} \tag{11}$$

and

$$w_2(p,q) = e^{-\frac{1}{2}(\frac{d_r}{\sigma_r})^2} \tag{12}$$

$$d_s = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \tag{13}$$

$$d_r = I(p) - I(q) \tag{14}$$

where, d_S is the Euclidician distance, d_r is the difference between two pixels' value, σ_d and σ_r are two user defined numbers that denote geometric and photometric spread [24] and (x_p,y_p) and (x_q,y_q) are coordinates of pixels p and q, respectively. These weights use two aspects

- How close are the neighbor and the center sample, so that the larger weights designate to closer samples.
- How similar are the neighbor and the center sample, hence, larger weights corresponds to more similar samples.

Hence, in smooth regions, pixel values in a small neighborhood are similar to each other, and the bilateral filter acts essentially as a standard domain filter. This idea is similar in spirit to the 'Beltrami Flow' proposed by Sapira, Sochen and Kimmel [26]. There, the effective weight is calculated as the 'Geodesic Distance', geodesic distance between pixels, between the samples. The calculation of weighting function of a bilateral filter for a one-dimensional signal is depicted by Fig.1.

Bilateral filtering can be considered as a non-iterative alternative to anisotropic diffusion. It has been shown that there is a fundamental relationship between the bilateral filter and anisotropic diffusion [27].

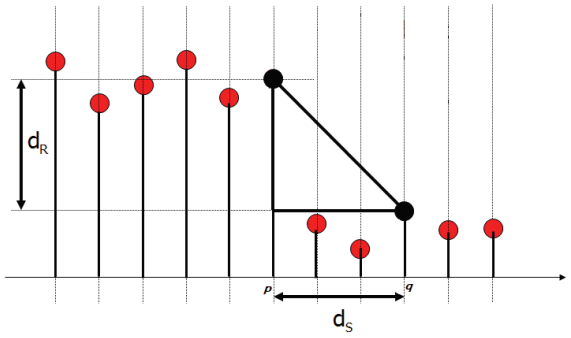


Fig. 1. Calculation of weighting function of a bilateral filter

3. BILATERAL INPAINTING

In this section a new inpainting method is proposed using a convolution with a bilateral averaging kernel. One of the main differences of the current approach with convolution-based inpainting is the fact that they use image information from only a space neighborhood. A more robust approach (in structure and texture preserving) of convolution–based inpainting is using information from both space and range.

Let Ω be the damaged area to be inpainted and ∂Ω be its one-pixel thick boundary. The algorithm consists of convolving the region to be inpainted with a bilateral kernel obtained by multiplying range and space kernels. By this, in regions with small gray level variations (uniform regions), the range kernel,

$w_2(p,q)$, tends to unity (because d_r is small) and the filter becomes a simple low pass Gaussian filter. Hence, we will have a rough propagation. In a similar manner, for edge regions, because of the large gray level variations, the range kernel becomes dominant. Hence, to preserve edges, $w_2(p,q)$ plays a determining role. In the inpainting process, the center of the convolution mask, p, is a degraded pixel and its gray level is not known. Hence, one cannot determine the range filter directly from Eqs. (12) and (14). To overcome this problem, one may replace $I(p)$ in Eq. (14) by its estimated value. To do this, we consider the range kernel as follows.

$$w_2(p,q) = e^{-\frac{1}{2}\left(\frac{\overrightarrow{P_{dir}}.\overrightarrow{\nabla I}}{\sigma_r}\right)^2} \tag{15}$$

Where, $\overrightarrow{\nabla I} = \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix}^T$ is the gradient vector corresponding to pixel q, and

$$\overrightarrow{P_{dir}} = \begin{bmatrix} x_p - x_q & y_p - y_q \end{bmatrix}^T \tag{16}$$

p is the damaged pixel to be inpainted and q presents its undamaged neighbor pixel. $\overrightarrow{P_{dir}}$ is the spatial vector from q to p, and σ_r is the photometric spread. As illustrated previously, (x_p,y_p) and (x_q,y_q) are coordinates of pixels p and q, respectively.

We estimate the damaged pixel by:

$$I_f(p) = \frac{\sum_{q\in N_a} w(p,q) I(q)}{\sum_{q\in N_a} w(p,q)} \tag{17}$$

$$w(p,q) = w_1(p,q) \times w_2(p,q) \tag{18}$$

where N_a is a w×w (where w is the size of neighborhood) neighborhood around the damaged pixel containing only non-damaged pixels. Calculating the kernel used for a specified pixel is the most time-consuming step of the proposed inpainting process. Hence, to gain time, N_a must be a tight neighborhood. All of the above steps are repeated for all damaged pixels until a stopping condition is reached which may be either quality of image or a predefined number of iteration, and the number of iterations may be determined for each inpainting domain by checking whether any of the pixels belonging to the domain had their values changed by more than a certain threshold during the previous iteration. According to our experimental studies, using only two iterations provides very adequate results in most cases. Fig. 2 shows the flowchart of the algorithm using K iterations (concerning each damaged pixel).

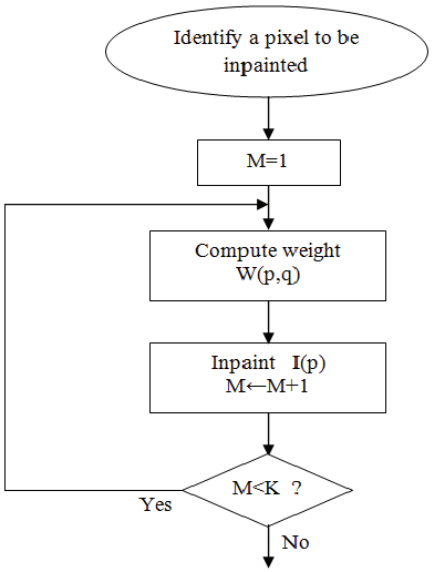


Fig. 2. The flowchart of the algorithm using K iterations

4. EXPERIMENTAL RESULTS

This section includes three parts. In the first part, we examine the role of algorithm parameters on the quality of the inpainted image, considering both theoretical and experimental aspects. In the second part the number of iterations is discussed. The last part is devoted to a comparative study. To do this, we inpaint several damaged images (of different regions) using some benchmark existing methods as well as the proposed method. We compare these methods considering the obtained image quality and the run-time as judging elements.

a) Parameter setting

This part of the paper is devoted to a discussion on the setting parameters of the proposed algorithm. As we explained so far, our proposed method has three parameters: σ_d , σ_r and w (size of neighborhood or window).

Both σ_d and σ_r determine the level of smoothness. Setting σ_r to zero reduces the bilateral filter to a simple Gaussian smoothing filter with a cut-off frequency proportional to 1/ σ_d. Hence smaller values of σ_d better preserve fine details and edges.

When the variance of the range filtering kernel is large (100 or 300) with respect to the overall range of values in the image (0 through 255), the range component of the filter has little effect, because all pixel values in any given neighborhood have about the same weight from range filtering, and the domain filter acts as a standard Gaussian filter. For smaller values of the range filter variance (10 or 30), range filtering preserves edges.

The size of the window must be determined in such a manner that at least one known pixel exists in the window region at the first iteration. Since σ_d actually determines the effective size of the window, w has no influence on quality.

A small value of σ_d causes a small considered neighborhood and therefore the inpainting process needs more iterations and, as Fig. 3 illustrates, only a small part of the damaged area will be reconstructed. A large value of σ_d will provide a blurring effect in both texture and structure regions. A midrate value of σ_d can inpaint the missed region adequately. The zoomed regions in Figs. 4 and 5 confirm the above discussion.

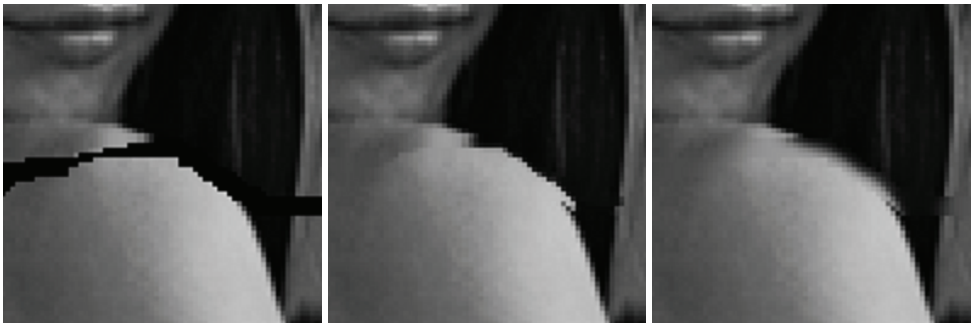
(a) (b) (c)

(d)

Fig. 3. Effect of σ_d with 2 iterations and σ_r=25.5 a) damaged image b) inpainted with σ_d=0.01 c) inpainted with σ_d=0.5 d) inpainted with σ_d=5

(a) (b) (c)

Fig. 4. Zoomed texture part of images (b), (c) and (d) in Fig. 3



(a) (b) (c)

Fig. 5. Zoomed structure part of images (b), (c) and (d) in Fig. 3

According to our experimental results, choosing large values for σ_r will blur both structure and texture regions. The blurring in structure regions is due to weighted averaging of pixels with high difference intensities, while in the texture region it is due to weighted averaging of a large number of pixels in the local neighborhood. A small value of σ_r causes a sharpening effect on the edges and, as Fig. 6 shows, in some regions the algorithm cannot properly inpaint the damaged region. A midrate value for σ_r provides a good reconstruction quality.

(a) (b) (c)

(d)

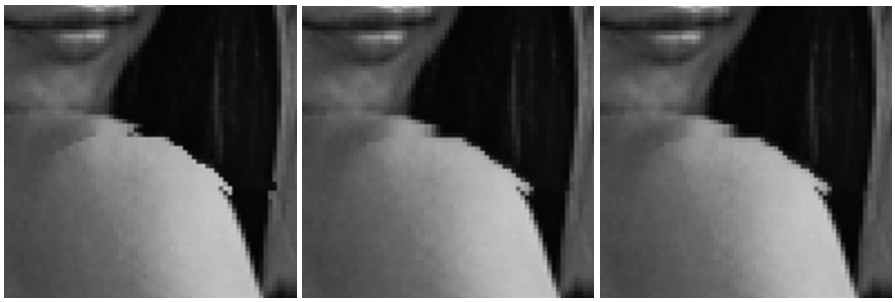
Fig. 6. Effect of σ_r with 2 iterations and σ_d=0.5. a) damaged image b) inpainted with σ_r=2.55 c) inpainted with σ_r=255 d) inpainted with σ_r=2250

b) Adaptive number of iterations

The effect of the number of iterations on the quality of the restored image is illustrated in Fig. 9, by considering three different values of iteration. As this figure suggests, inpainting using a small number of iterations (e. g. 2 iterations) results in a good restoration of texture regions, but provides a blurring effect in the structure regions. As Fig. 9 shows, a larger number of iterations provide better edge reconstructing, however, texture regions will be blurred, which is undesirable. To overcome this problem, we propose using a different number of iterations for texture and structure regions. To determine the type of the region, a threshold is applied on the local variance of the known neighborhoods of the missed pixel. This idea is depicted in Fig. 12. White regions in Fig. 12d are areas where the algorithm needs more iterations (structure regions) and Fig. 12e shows the final result obtained by the adaptive number of iterations.

(a) (b) (c)

Fig. 7. Zoomed texture part of images (b), (c) and (d) in Fig. 6



(a) (b) (c)

Fig. 8. Zoomed structure part of images (b), (c) and (d) in Fig. 6

(a) (b) (c)

(d)

Fig. 9. Effect of iteration with σ_d=0.5 and σ_r =25.5 a) damaged image b) inpainted with 2 iterations c) inpainted with 10 iterations d) inpainted with 100 iterations

(a) (b) (c)

Fig. 10. Zoomed texture part of images (b), (c) and (d) in Fig. 9

(a) (b) (c)

Fig. 11. Zoomed structure part of images (b), (c) and (d) in Fig. 9

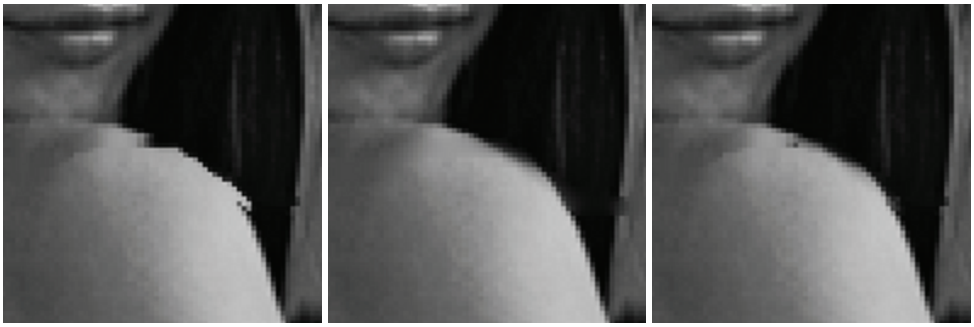
(a) (b) (c)

(d) (e)

Fig. 12: σ_d=0.5 and σ_r =25.5 a) damaged image b) inpainted with 2 iterations without threshold c) inpainted with 100 iterations without threshold d) regions that need more iterations e) inpainted with 2 iterations for texture areas and 100 iterations for structure regions

(a) (b) (c)

Fig. 13. Zoomed texture part of images (b), (c) and (e) in Fig. 12



(a) (b) (c)

Fig. 14. Zoomed structure part of images (b), (c) and (e) in Fig. 12

c) Comparative study

To evaluate the performance of the proposed bilateral inpainting, in this section we perform a comparative study using our algorithm and five well known inpainting algorithms: A texture synthesizing based method (the exemplar based inpainting proposed by Efros et al.[19]), a benchmark diffusion based one (Bertalmio algorithm [1]), the TV algorithm (Chan an Shen [5]), convolution based inpainting algorithm (the Oliveira algorithm[15]), and the fast image inpainting based on coherence transport (Bornemann and Marz [17]).

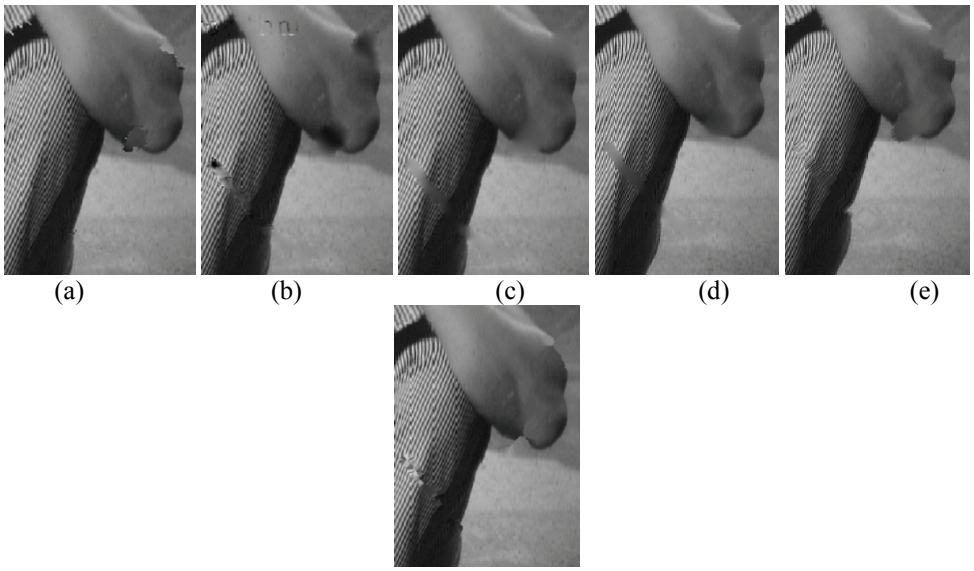
We used several pictures of different contents and damaged areas of different types and sizes. Because of the similar results, we present here only the results of two images "Barbara" and "Nature".

We opt for these images because both have textural and structural regions. In addition, they have damaged regions with different types and shapes. These images are illustrated in Figs. 15-17.

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

(e) (f) (g)

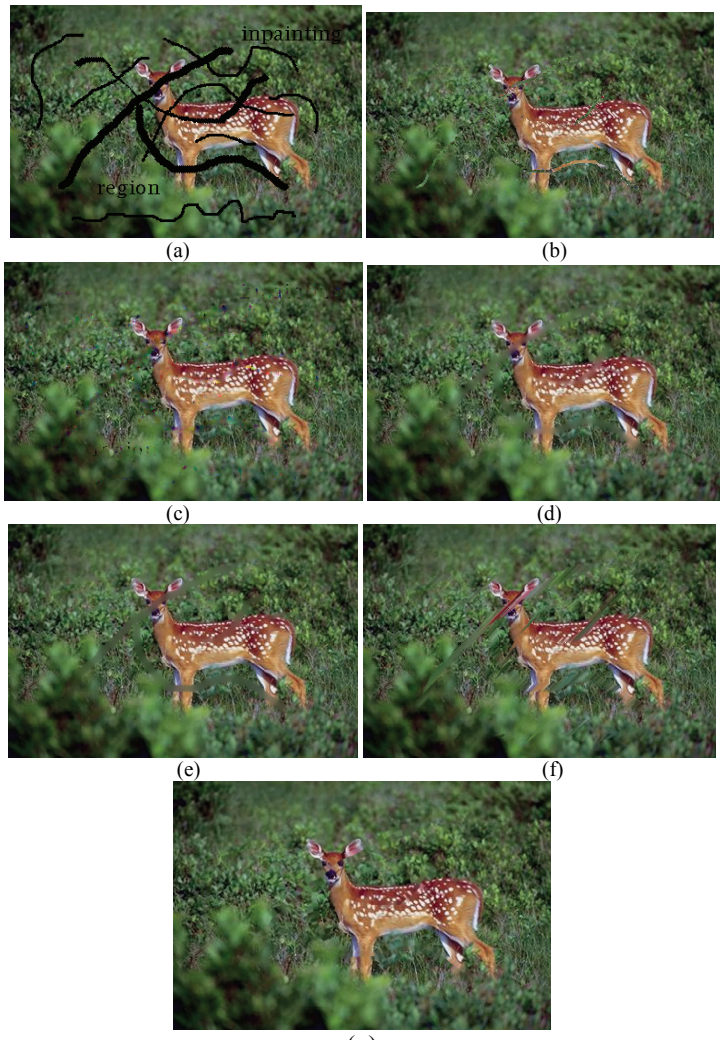
Fig. 15. a) damaged image b) inpainted with Efros model c) Bertalmio's algorithm result d) restored by Oliveira method e) result of TV algorithm. f) result of Bornemann's approach. g) proposed algorithm result with σ_d=0.5 and σ_r =25.5 and 2 iterations for texture regions and 80 iterations for structure regions



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

(f)

Fig. 16. Zoomed of Fig. 15 a) result of Efros model b) restored by Bertalmio method c) Oliveira approach d) result of TV inpainting. e) result of Bornemann algorithm. f)proposed algorithm



(a) (b)

(c) (d)

(e) (f)

(g)

Fig. 17. a) damaged image b) inpainted with Efros model c) Bertalmio's algorithm result d) restored by Oliveira method e) result of TV algorithm. f) result of Bornemann approach. g) proposed approach result with σ_d=0.5 and σ_r =25.5 and 2iterations for texture and 100 iterations for structure regions

All six mentioned methods were applied to these images. The qualitative results are shown in Figs. 15 -17. As Fig. 15 suggests, Efros algorithm provides the best result in texture regions but it cannot restore large structure damaged regions like the elbow of Barbara. However for the "Nature" image, Efros algorithm has good results in some regions but not in other regions. Bertalmio's algorithm gives a good result in small damaged structure area, but is not able to reconstruct large structure and texture damaged regions. TV approach has very good results in thin damaged edges, but results in blurring in the texture regions and large damaged edges and does not have good results in natural texture images like 17. Oliveira method blurs both structure and texture regions. Bornemann method concludes in very good results in small structure and texture regions, but it cannot inpaint large damaged edges and natural textures, leading to poor effects in these images. The proposed algorithm provides good results in both large structure areas and texture regions.

All algorithms are implemented using a 2.4GHz PC in MATLAB 7.8 environment. A quantitative comparison of the runtimes is given in Table 1. According to this table, for the Barbara image, the Bornemann's algorithm is the most rapid and the proposed one is the second. For "Nature", Oliveira method is faster than the others, but with a quality of the inpainted image much lower than the proposed algorithm and again the presented model is the second. Therefore, we can conclude that the presented algorithm has acceptable speed.

Table 1. Implementation times of algorithms (s)

Number of figures	Efros	Bertalmio	Total variation	Oliveira	Bornemann	Proposed method
15	270.33	3610.65	37.19	48.65	4.92	21.42
17	526.94	947.05	50.3	31.22	84.16	73.25

5. CONCLUSION

In this paper, an inpainting method based on bilateral filtering was proposed. We utilized the multiplication of gradient of known pixels and direction between known and damaged pixels, instead of their differences in the range filter of a conventional bilateral filter. The proposed method is a convolution based iterative algorithm, and provides good and rapid restoration. To evaluate the performance of the algorithm, several comparative experiments were performed. The results confirmed the efficiency of the algorithm.

Acknowledgment: This work is supported in part by Iran Telecommunication Research Center.

REFERENCES

1. Bertalmio, M., Sapiro, G., Caselles, V. & Ballester, C. (2000). Image inpainting. *Proceeding of SIGGRAPH 2000 Computer Graphics Processing*, pp. 417-424.
2. Au, W. & Takei, R. (2002). Image inpainting with the navier-stokes equations. APAM 930.
3. Torkamani-Azar, F. (2001). Image recovery using diffusion equation embedded neural network. *Iranian Journal of Science and Technology, Transaction B, Engineering*, Vol. 25, No. B1, pp. 27-35.
4. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G. & Verdera, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Signal Processing*, pp.1200-1211.
5. Chan, T. & Shen, J. (2001). Mathematical models for local nontexture inpainting. *SIAM J. Appl, Math*., Vol. 62, No. 3, pp. 1019-1043.
6. Chan, T. F., Shen, J. & Kang, S. H. (2002). Euler's elastica and curvature based inpainting. *SIAM journal of Applied Mathematics*, pp. 564-592.

7. Chan, T. & Shen, J. (2001). Non-texture inpainting by curvature-driven diffusions (CDD). *J. vis. Commun. Image Represen*., Vol. 12, No. 4, pp. 436-449.
8. Grossauer, H. (2004). A combined PDE and texture synthesis approach to inpainting. Lecture Note in Computer Science, Vol. 3022, pp. 214-224.
9. Bertalmio, M., Vese, L., Sapiro, G. & Osher, S. (2003). Simultaneous structure and texture image inpainting. *UCLA CAM report* 02-47.
10. Chan, T. F., Shen, J. & Zhou, H. M. (2005). A total variation wavelet inpainting model with multilevel fitting parameters.
11. Tai1, X. C., Osher, S. & Holm, R. (2005). Image inpainting using a TV-stokes equation. *IEEE Trans. Image Process*.
12. Feng, Zh., Chi, Sh., Yin, J., Zhao, D. & Liu, X. (2007). A variational approach to medical image inpainting based on mumford-shah model. IEEE, 1-4244-0855.
13. Li, L. & Yu, H. (2009). Nonlocal curvature-driven diffusion model for image inpainting. *IEEE Conference on Information Assurance and Security*, 978-0-7695-3744, 2009.
14. Tlea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, Vol. 9, No. l, ACM, press.
15. Manuel, O., Bowen, M. B., McKenna, R. & Chang, Y. S. (2001). Fast digital image inpainting, *Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP. 2001) Marbella*, Spain, pp. 261-266. (ISBN 0-88986-309-1).
16. Hadhoud, M. M., Moustafa, K. A. & Shenoda, S. Z. (2005). Digital images inpainting using modified convolution based method. *International Journal of Signal Processing*, Image Processing and Pattern Recognition.
17. Bornemann, F. & Marz, T. (2007). Fast image inpainting based on coherence transport. *Journal of Math. Imaging and Vision*, Springer, Vol. 28, pp. 259-278.
18. Weickert, J. (2003). Coherence-enhancing shock filters. *Pattern Recognition. Lecture Notes in Computer Science*, Vol. 2781, Springer, New York.
19. Efros, A. & Leung, T. (1999). Texture synthesis by non-parametric sampling. *proc. IEEE International Conference Computer Vision*, pp.1033-1038, Corfu, Greece.
20. Efros, A. & Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. *Proceeding of SiGG RAPH*, pp. 341-346.
21. Criminisi, A., P´erez, P. & Toyama, K. (2004). Object removal by exemplar-based inpainting. *IEEE, Trans. Image Processing*, Vol. 13, No. 9, pp. 1200-1212.
22. Wong, A. & Orchard, J. (2008). A nonlocal-means approach to exemplar-based inpainting. *IEEE*, 978-1-4244-1764.
23. Buades, A., Coll, B. & Morel. J. M. (2005). Image denoising by non-local averaging. *IEEE*, 0-7803-8874.
24. Tomasi, C. & Manduchi, R. (1998). Bilateral filtering for gray and color images. *IEEE, Trans*., 96-1-0007.
25. Chan, T. F., Osher, S. & Shen, J. (2001). The digital TV filter and nonlinear denoising. *IEEE Trans*. on Image Processing, Vol. 10, No. 2, pp. 231–241.
26. Sapira, A., Kimmel, R. & Sochen, N. A. (2007). A short-time beltrami kernel for smoothing images and manifolds. *IEEE Trans. On Image Processing*, No 6, Vol. 16, pp. 1628-1636.
27. Barash, D. (2002). A fundamental relationship between bilateral filtering, adaptive smoothing and the non-linear diffusion equation. *PAMI*, Vol. 24, No. 6.