International Journal of Software Engineering and Its Applications
Vol. 9, No. 2 (2015), pp. 143-150
http://dx.doi.org/10.14257/ijseia.2015.9.2.12

Distance weighted Bounding for Fast Exemplar-based Inpainting

Chunduck Park and BaekSop Kim*

Department of Computer Engineering, Hallym University, Korea
**bskim@hallym.ac.kr*

Abstract

Exemplar-based image inpainting method is widely used for recovering the missing regions, especially when the missing regions are large. The method fills in the target region patch by patch, and a target patch is compared to each candidate patch in whole image to get a best one. In this paper we propose a method to speed up the best candidate searching process. Inspired by the fact that the patches near to a target patch are more likely to be the best candidate than the farther ones, the similarity measure is modified to cooperate with the distance weighting and bounded search is adopted. It has been shown by experiments that the proposed method greatly reduces the number of pixel error computations.

Keywords: *image inpainting, fast inpainting, distance weighting, spiral search*

1. Introduction

Image in painting is a process of filling-in missing data in the unknown or deleted regions in an image. It is used not only for recovering scratches or removing missing parts but also for removing an unwanted object from the image and replacing with the background plausible to the human eye. Digital image in painting started when Bertalmio, *et al.*, [1] proposed a partial differential equation (PDE)-based method in 2000. It provided excellent result when the missing region is small, but introduced smoothing effect when the missing region is relatively large. Lots of researches have done since then, but the exemplar-based algorithm proposed by Criminisi, *et al.*, [2] in 2004 stands out since it is not only fast but provides good result for the larger missing region.

The exemplar-based in painting consist of two steps: target patch selection and best candidate patch searching. In the target patch selection step, the target patch with highest priority is selected in the boundary of the missing region. In the candidate patch searching step, the target patch is compared with other patches (candidate patches) in the known regions. Judging from the known pixels in the target patch, the most similar candidate patch is selected and the target patch is replaced with the selected candidate patch. These two steps iterate until the whole missing region is filled.

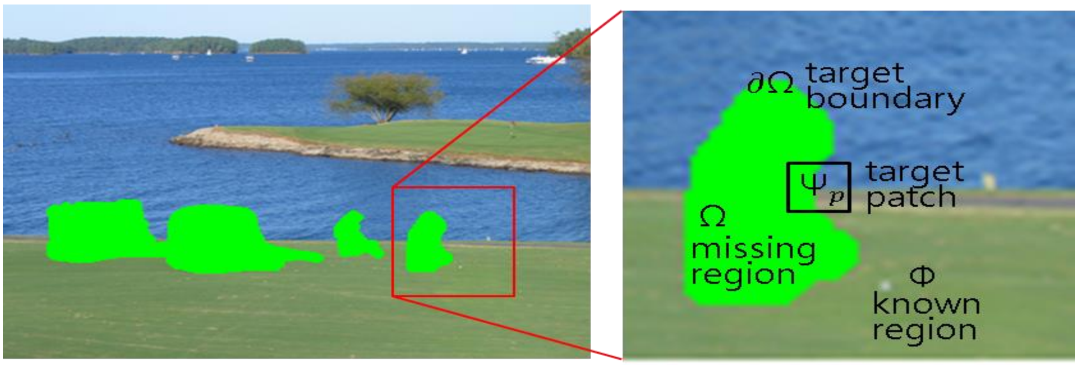
There have been a lot of efforts to improve the image quality and speed of the Crimini's algorithm. Cheng, *et al.*, [3] improve the process of target patch selection. In Crimini's algorithm, the confidence term used in the target patch priority calculation tends to decrease rapidly which makes the priority values undistinguishable. This results in incorrect filling order and the structural information is not in painted well. Cheng, *et al.*, modified the priority function so as to add the confidence and data term instead of multiplication. Zhang and Lin [4] proposed a new priority function for target patch selection. Since Crimini's method uses local gradient to calculate the data term in the priority function, the texture boundary was not in painted well. They used the texture information of the patches neighboring the target patch in the data term to retain the texture boundary.

Copyright ⓒ 2015 SERSC

Komodakis, *et al.,* [5] proposed a global optimization method for the exemplar-based in painting. Since Crimini's method relies on the greedy search without backtracking, the search error in early stage can be propagated. The proposed method can remove inconsistency caused by greedy methods, but takes a huge amount of time. To reduce computation time, two techniques called priority-based message scheduling and dynamic label pruning are proposed. Still, this method takes longer time than Crimini's. Jawas and Suciati [6] proposed the exemplar-based in painting which is processed by dilation and erosion morphological operations.

The exemplar-based in painting method requires a lot of computations, mainly because it searches the candidate patch exhaustively for filling in a target patch. In 2010, Anupam, *el al.*, [7] limited the search space based on the location of the target patch. The idea is that since the best candidate patch is more like to be in the vicinity of the target patch, discard the candidate patches that are far away. However, the quality of the in painted image can be degraded if the best matched candidate patch were in the discarded region. Liu, *et al.*, [8] also limited the search region of the candidate patches according to the distance to the target patch. The limit is set to be inverse proportional to the confidence value of the target patch. They also proposed a hybrid similarity measure between the target and candidate patch. Kim and Park [9] added the distance weighting to the similarity measure between the target and candidate patch. They reported that the in painted result was improved by appending higher priority to the nearer candidate patch. However the computing time is not improved. Recently, Kim, *et al.*, [10] proposed a fast algorithm which adopts an bounding strategy with pixel and patch reordering, and showed experimentally that the number of pixel distance calculations is greatly reduced compared to that of Criminisi's.

In this paper we extend the method of [10] so that the number of the distance calculation is reduced further by using distance weighting and spiral search. Distance weighting can reduce the number distance calculations if it used in conjunction with bounding strategy and spiral search.



(a) **(b)**

Figure 1. (a) Original Image with Missing Region Marked in Green (b) Glossaries used for Exemplar-based in Painting Algorithm

2. Background

Let Ωbe missing region to be inpaintedthis is marked manually in green color in Figure 1-(a).The rest of the missing region is the known region denoted by Φ, that is, Φ=I-Ω. I means the whole image. As shown in Figure 1-(b), $\partial\Omega$ denotes the boundary of the missing region Ω. $\Psi_p$is a patching region centered at a pixel pon $\partial\Omega$. The patch is usually set to 9x9 pixels, and a part of the pixels in $\Psi_p$are in the missing region Ω, and the rest is in the known region.

 Copyright ⓒ 2015 SERSC

Crimini's exemplar-based algorithm is shown as a pseudo code in Figure 2.

fill initial confidence C(.).
whileΩ is not empty **do**
$\partial\Omega$:= boundary of Ω
Target patch selection :
Select a pixel *p* having maximum priority value among those in $\partial\Omega$.
Ψ_p := the target patch centered at *p*.
Best candidate patch searching :
Find a candidate patch in Φ that is most similar to Ψ_p.
Ψ_q := the best candidate patch.
Copy Ψ_q onto Ψ_p.
Update the known region $\Phi = \Phi \cup \Psi_p$.
Update the confidence C(.) for the pixels in $\Psi_p \cap \Omega$.
end // while Ω

Figure 2. Criminisi's Exemplar-based in Painting Algorithm

The confidence values C(.) for the pixels in Ω (unknown pixels) is set to 0, and those in Φ (known pixels) are set to 1 initially. When a target patch is copied from a candidate patch, as in the last line of the algorithm in Figure 2, a part of the unknown pixels are filled with the known pixel values. The confidence values of the filled pixels, in this case, are updated as follows.

$$\mathrm{C(p)} = \frac{\sum_{q \in \Psi_p \cap \Omega} C(q)}{|\Psi_p|} \quad (1)$$

Here, $|\Psi_p|$ means the number of pixels in Ψ_p.

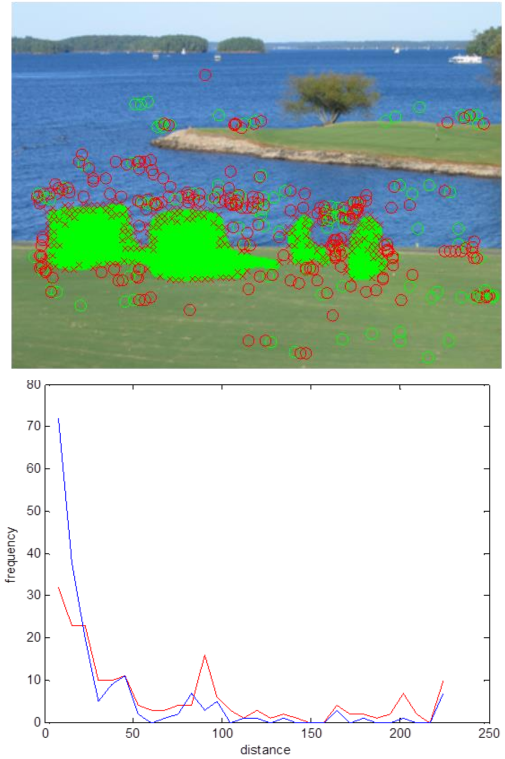
In the target patch selection step, a pixel with a highest priority values is selected from those in the target boundary. The priority of a pixel *p* is defined as P(*p*)=C(*p*)D(*p*), where C(*p*) is the confidence term. D(*p*), which is called the data term, is defined as follows.

$$\mathrm{D(p)} = \frac{\nabla I_p^+ \cdot n_p}{\alpha}, \quad (2)$$

whereis the isophote at *p* and is the normalization factor. Is the unit normal vector orthogonal to $\partial\Omega$ at *p*.

Let Ψ_p ∇I_p^+ be the target patch centered at *p*, α a pixel with a highest priority n_p value. In the best candidate searching step, the target must be compared to every patch centered at the pixels in Φ (called *candidate patch*) to get the most similar one. To compute similarity of two patches, the SSE(Sum of Squared Error) of the known pixels is used. This is a very time-consuming job, which takes long time especially when executed on devices such as mobile devices. Note that this process is iterated until the whole unknown pixel is filled.

Many efforts have done to speed up the best candidate searching time. The most simple and efficient way is to limit the search space as in Anupam, *el al.*, [7] and Liu et al, [8]. In Figure 3-(a), the best candidate patches found during inpainting are shown by the green circles. Let the good patch set is a set of the candidate patches whose error(SSD) is not larger than that of the best candidate patch by 1dB. A candidate patch that is closest to the target patch in the good patch set is called the closest good patch. The red circles in Figure 3-(a) shows the closest good patches. The in painted result using the closest good patches are almost the same as that of the best candidate patches. The most of the closest good patches are in the vicinity of the target patches, as shown in Figure 3-(a). Although we can limit the search region of the candidate patches in most cases, a few closest good patches are in the far region. Figure 3-(b) shows the distribution of the best candidate patches in red line and that of the closest good patches in blue line. Figure 3-(b) shows that although the number of the closest good patches is much smaller than that of the best candidate patches, they are not negligible.



(a) (b)

Figure 3. (a) Locations of the Best Candidate Patches (Green Circles) and Closest Good Patches (Red Circles), (b) their Distributions Blue and Red Line are for the Best Candidate Patches and Closest Good Patches, Respectively

Limiting the search region is effective way for speed up computation; however the inpainted result may be degraded when the best candidate patch outside the search region cannot be neglected. It is safer to give more importance to the closer candidates rather than neglecting the farther candidates. Kim and Park [9] added the distance weights to the similarity measure so that the closer candidates are more easily selected than the farther candidates. They showed experimentally, that the quality inpainted image are improved by this way. The computing time is not improved because any candidate patch is not ignored.

This paper proposes a candidate searching method which uses the distance weights to the similarity measure as in [9], but the computational speed is greatly improved.

3. Distance Weighted Bounding Algorithm

In the best candidate patch searching step, a patch whose error to the target patch is selected. The patch error between the target patch $\Psi_p$and the candidate patch $\Psi_q$is defined by the sum of squared error (SSE) of the pixels, which is

$$\mathrm{E}(\Psi_p\,,\ \Psi_q) = \sum_{i,c_i \in \Psi_p \cap \Phi} \left[(R_i - R'_{c_i})^2 + (G_i - G'_{c_i})^2 + (B_i - B'_{c_i})^2 \right] \qquad (3)$$

 Copyright ⓒ 2015 SERSC

Here, (R_i,G_i, B_i) denotes R, G, B values of the pixel i in $\Psi_p \cap \Phi$and (R_i', G_i', B_i') is the pixel values of the pixel in i in $\Psi_q \cap \Phi$. C_i' means the pixel in Ψ_q whose position corresponds to the pixel i in Ψ_q. It is worth to be noted that the patch error is the sum of the errors of the pixels in the patch, which means the patch error is accumulated by the pixel errors.In order to select a best candidate patch, the patch error is no longer need to be accumulated if the patch error exceeds the current minimum patch error during accumulation. Let the current minimum patch error refer to as the bound. The bound is monotonically decreasing as search proceeds. Let us call that the patch is "discarded", if it stops error accumulation during calculating patch error.

The faster the bound is decreasing during search, the more the discarded patch is. The distance weighting function, discussed in the above section, can be viewed as reducing the bound for the farther candidate patches, thus they are more prone to be discarded. As shown in Figure 3-(b), the distribution of the best candidate patches is high where their distance to the target patch is small. Based on these facts, a best candidate searching algorithm is proposed, which is shown in Figure 4.

Given : target patch Ψ_p
Find : best candidate patch$\Psi q \in \Phi$
Method
S := Φ.
Emin := large number.
whileS is not empty **do**
Finda pixel $r \in$S closest to p, the center of the target patch.
Let $\Psi_r$be the patch whose center is at r.
Remove r from S.
bound :=*Emin* / $w(\|p - r\|)$.
T :=$\Psi_r \cap \Phi$.
E := 0.
while T is not empty **do**
Select a pixel t in T.
Remove t from T.
e:= $(R_p - R_t')^2 + (G_p - G_t')^2 + (B_p - B_t')^2$
E := E + e.
if$E \geq bound$**goto** discard.
end // while T
if$E < Emin$**do**
Emin := *E*.
q := *r*.
end
discard :
end // while S

Figure 4. Best Candidate Searching Algorithm that is based on Distance Weighted Bounding

There are two while-loops in Figure 4. In the outer while loop, a candidate patch selected in order of proximity to the target patch. The patch error between the candidate patch and the target patch is calculated by accumulating the pixel errors e, as shown in the inner while loop. The calculation is stopped and that candidate patch is discarded if the accumulated pixel errors exceed the bound. The bound is the current minimum SSE weighted by the distance between the target and candidate

Copyright ⓒ 2015 SERSC

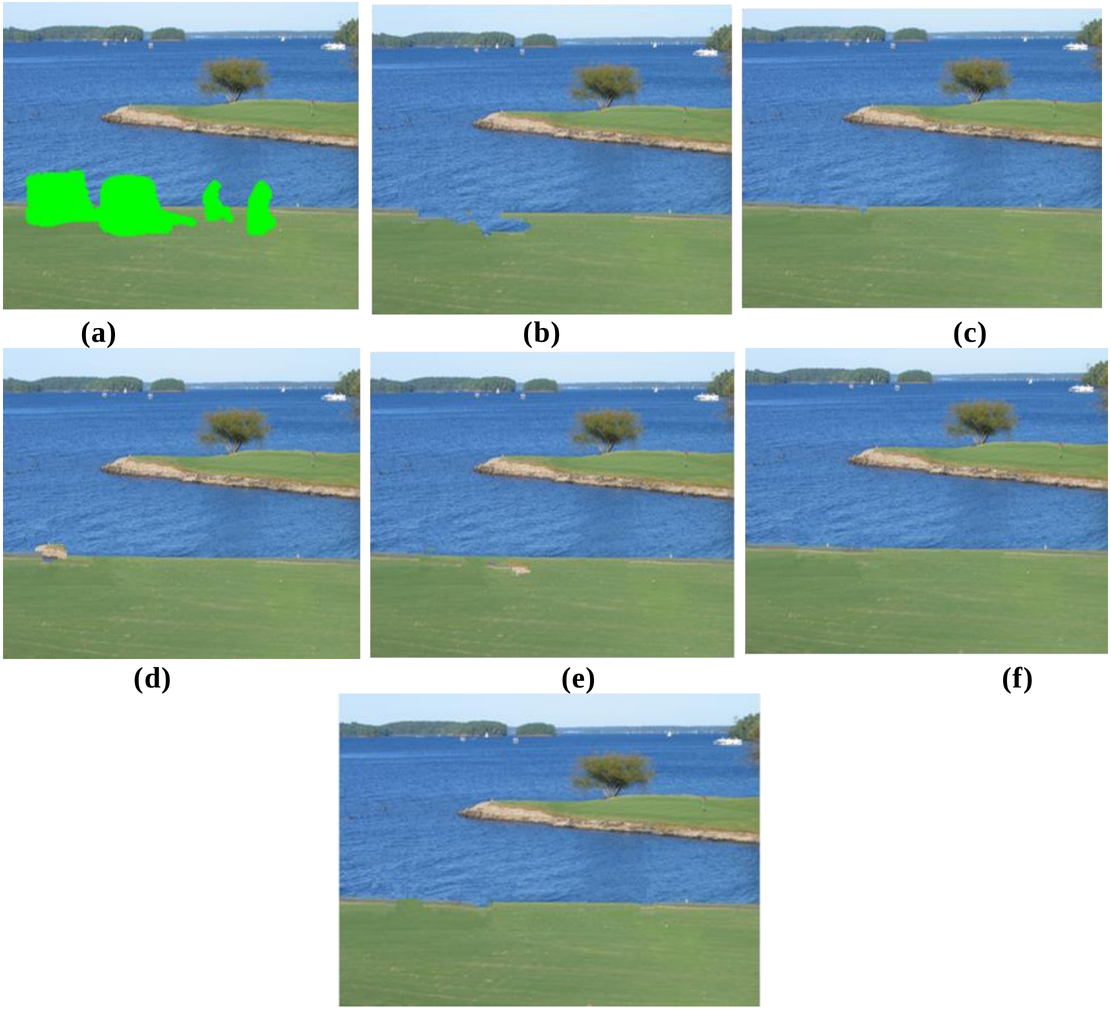
patch. The weight is a function of the distance from p to r. Let $d = \|p - r\|$ be the distance from p to r. The weight function is defined by

$$w(d) = \max\{1, \ \beta \cdot (d - d_t)/d_{max}\} \quad (4)$$

Here, β and $d_t$are the user-defined parameters.$d_{max}$is the diagonal size of the input image, that is, $d_{max} = \sqrt{width^2 + height^2}$.

4. Results and Discussion

Experiments have done to compare the result with the original Criminisi's method and hard bounding methods. Figure 5 shows the result for Golf image.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 5. (a) Original Image with Missing Regions, (b)~(e) Inpainted with Distance Limiting. Limit was Set to 0.3, 0.5, 0.7, and 0.9, Respectively. (f) Inpainted by Criminisi's Method(g) Inpainted by the Proposed Method

Figure 5-(b)~(e) are the result when the search region is limited to 30%, 50%, 70% and 90% of d_{max}, respectively. It shows that the result is sensitive to the limit parameter, and the larger bound does not necessarily produce better result: Image quality was better when the bound is set to 50% than that of 70%. When the bound was enlarged, the result became almost the same as that of Criminisi's.The proposed

 Copyright ⓒ 2015 SERSC

method was not so sensitive to the user defined parameters. Figure 5-(g) is the result of the proposed method, which is almost the same as the Criminisi's in Figure 5-(f).

The same experiment has done for Bungee image, whose results are shown in Figure 6. In this case, the image quality of the hard limiting method was better when the bound is set to 30% than that of 50% or 70%. This is because that human is more sensitive to the structure than the textures. When the search region is limited to 30% in the figure, the missing regions on the top of the roof is copied from the near patches, which results the roof looks connected. This is not true for every case. Note that, in case of Golf image, the image whose limit is set to 30% was the worst. This is because that the good candidate patch was not within a small search region. In both case, the result converge to that of Criminisi's when the limit is set to large. But this requires lots of computations. The proposed method showed reliable results in any cases, because it does not ignore any candidate patch.

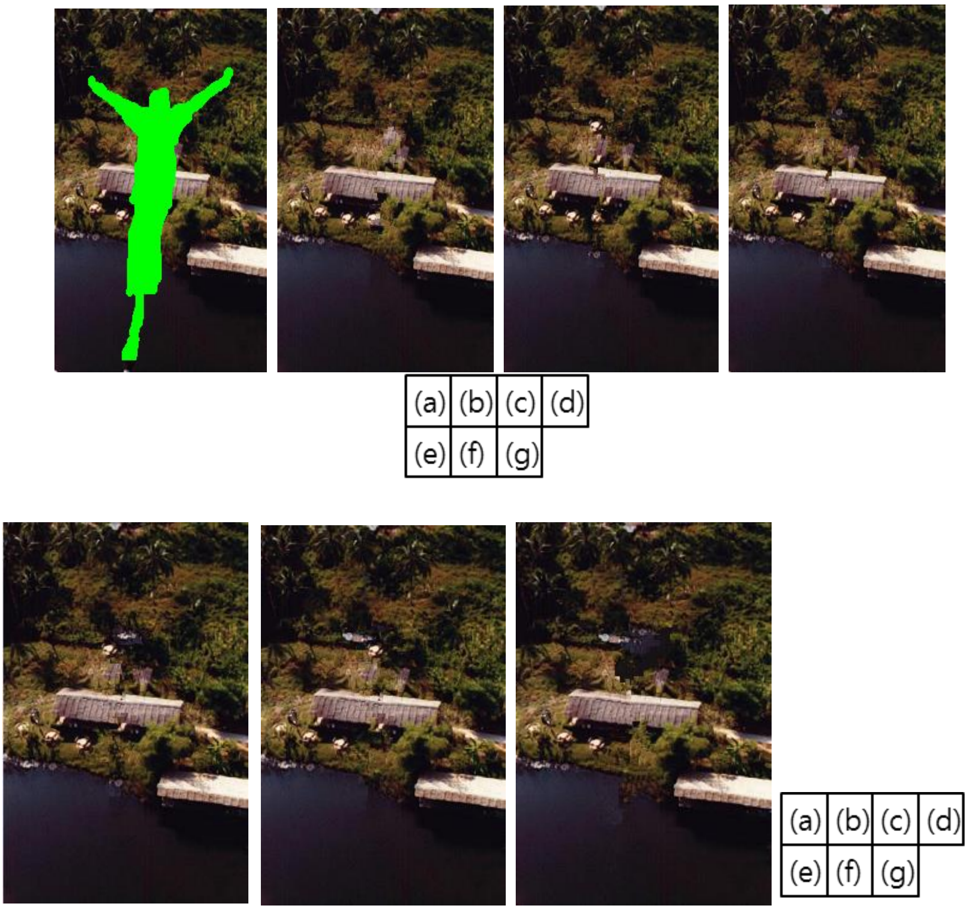


Figure 6. (a) Original Image with Missing Regions, (b)~(e) Inpainted with Distance Limiting. Limit was Set to 0.3, 0.5, 0.7, and 0.9, Respectively. (f) Inpainted by Criminisi (g) Inpainted by the Proposed Method

The numbers of pixel error computations of each algorithm are shown in Table 1. The proposed method required about 1/5 pixel error computations compared to the Criminisi's method. It amounts to 30~50% hard limiting. But the image qualities of the hard limiting methods are not reliable unless the limit is large, while those of the proposes method are almost the same as that of Criminisi's.

Copyright ⓒ 2015 SERSC

Table 1. Number of Pixel Distance Computations of Each Algorithm (in Millions)

Algorithm	Golf image	Bungee image
Criminisi's	1036	1032
30% Hard Limiting	90	58
50% Hard Limiting	313	212
70% Hard Limiting	602	441
90% Hard Limiting	873	658
Proposed	181	204

5. Conclusion

A distance weighted bounding algorithms for the exemplar-based inpainting is proposed. Theexamplar-based inpainting fills in the missing region patch by patch. During filling in a target patch, it should be compared to every candidate patch, which requires a lot of computations. Inspired by the fact that the patches near to a target patch are more likely to be the best candidate than the farther ones, the similarity measure is modified to in cooperate with the distance weighting. The proposed algorithm greatly reduces the number of pixel error computations by utilizing the distance weighting during bounded searching.

Experiments show that the image quality of the proposed method is almost same to that of Criminisi's, while the number of pixel error computations reduces as much as 1/5. The proposed method has user-defined parameters. The performance of the proposed algorithm may be improved if they are adjusted in each image.

Acknowledgement

This research was supported by Hallym University Research Fund, 2013(HRF-201310-013).

References

[1]. M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image Inpainting," Proceedings of SIGGRAPH, (**2000),** pp. 417-424.

[2]. A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," IEEE Transactions on Image Processing, vol. 13, **(2004).**

[3]. W. Cheng, C. Hsieh, S. Lin, C. Wang and J. Wu, "Robust algorithm for exemplar-based image inpainting", CGIV, (**2005),** pp. 64–69.

[4]. Q. Zhang and J. Lin, "Exemplar-based Inpainting using Color Distribution Analysis", Journal of Information Science and Engineering, **(2011)**.

[5]. N. Komodakis and G. Tziritas, "Image completion using global optimization", IEEE CVPR, (**2006),** pp. 442–452.

[6]. N. Jawas and N. Suciati, "Inpainting using Erosion and Dilation Operation", International Journal of Advanced Science and Technology, vol. 51, (**2013)** February.

[7]. P. G. Anupam and S. Diwakar, "Fast and enhanced algorithm for exemplar based inpainting", 4th PSIVT, **(2010).**

[8]. J. Liu, S. Zhang, W. Yang and H. Li, "A Fast Image Inpainting Method Based on Hybrid Similariy-distance," 20th ICPR, (**2010),** pp. 4432-4435.

[9]. B. Kim and J. Park, "Exemplar Based Image Inpainting on a Projection Framework", International Journal of Software Engineering and Its Applications, vol. 7, no. 3, (**2013)** May.

[10]. B. Kim, J. Kim and J. So, "Pixel and Patch Reordering for Fast Patch Selection in Exemplar-Based Image Inpainting", IEICE Transactions on Information and Systems, E96-D, vol. 12, (**2013),** pp.2892-2895.

Copyright ⓒ 2015 SERSC