# A FUZZY SET GENERALIZATION OF THE EXEMPLAR-BASED IMAGE INPAINTING

Mihaela CISLARIU, Mihaela GORDAN, Aurel VLAICU
*Technical University of Cluj-Napoca, Cluj-Napoca, Romania*
*28 Memorandumului, 401285, Mihaela.Cislariu@com.utcluj.ro, Mihaela.Gordan@com.utcluj.ro*
*Aurel.Vlaicu@com.utcluj.ro*

**Abstract: Inpainting is a technique of modifying an image in an undetectable way, by removing large objects or damaged areas and replacing them with plausible background. In this paper we have approached the exemplar-based image inpainting method proposed by Criminisi. This method uses patches from the image to fill the damaged region. Here we propose a new fuzzy set generalization of the exemplar-based image inpainting by using a fuzzy window. We use the fuzzy window to try to improve the finding of the most similar patch for filling. Experimental results show that the proposed method yields improvements compared to other methods.**

***Keywords:*** *Image restoration, inpainting, fuzzy logic, RGB color space.*

## I. INTRODUCTION

Inpainting is the art of modifying an image in such a way that for a new observer the new result looks like a natural image. The applications for the inpainting algorithms are various, from the removal/replacement of the objects to the restoration of the damaged photos or paintings (as gaps, stains on the old photos or paintings).

In this paper we have approach the inpainting by the texture-oriented method the so called exemplar-based texture synthesis described in [3] by Criminisi. As described in his paper the removal/replacement of the large object is done by computing the most similar patch for filling this region. In this case this patch is a little part of the already existing information from the image.

A major problem in the exemplar-based texture synthesis is the appearance of artifacts as a direct result of the bad texture propagation. If the template window is bigger than the structural element (texel) in the texture propagation there can appear some distortions that in the next step is considered to be texture. In this way the result of the exemplar-texture inpainting might not be likely for a new person, it can be an un-natural image and as a result it will be consider as a forgery of the image, a fake image. There are some attempts to improve Criminisi's work like [4] where the authors add some new coefficients to the algorithm to improve the refilling of the area with the new information. The values and factor used here are dependent of the observation of the user, and they cannot be automatically adjusted. In [6] the authors propose a new approach to the exemplar-based inpainting using local-means with nonlocal image information from multiple samples and they select the samples used based on the underlying image content. In [7] the authors proposed an algorithm that improves the block matching based on regional segmentation and size adaptive window.

A new challenge for this algorithm is the use of the artificial intelligence [5], but there are very few attempts so far. In [1] the authors propose a general variational framework for a non-local image inpainting, from which important and representative previous inpainting schemes can be derived. They use the fuzzy intelligence as a non-local weight function. When the fuzzy correspondence converges to a dense correspondence map, this iterative process generates a sort of patch work. The size of the patch gives the width of the band around the boundary between the segments, where the inpainting domain is partitioned into arbitrary shaped segments which show an exact copy of some region in the hole's complement.

In this paper we propose a generalization of the exemplar-based inpainting algorithm using the fuzzy sets; this algorithm uses a fuzzy window. This fuzzy window is used for searching and finding a better patch match for filling the missing area. After the selection of the area to be restored, the algorithm automatically fills the region with the existing information. Using a fuzzy window we can fill the missing data using the context information for a better result. The results are in some cases much better than those for the crisp algorithm.

## II. BRIEF SURVEY OF THE EXEMPLAR-BASED IMAGE INPAINTING

Criminisi in his work [3] presented an efficient algorithm that combines the advantage of texture synthesis and PDE-based algorithm. The algorithm achieves the propagation of the structure and texture simultaneous. The main idea used in the exemplar-based inpainting is to use known information from the image to fill the target region (the region without information).

Let us consider the spatial region marked for replacing by $\Omega$ and its contour by $\delta\Omega$. The region $\Omega$ is typically called "the target region", whereas its complement, denoted by $\Phi$, is called "the source region" (from which useful and reliable image information is "copied" in the inpainting process, in order to remove the large size defect). The source region

*Manuscript received May 6, 2011; revised June 17, 2011*

remains fixed throughout the algorithm. For filling the target region it is used a template window $\psi$ centered on the border $\delta\Omega$, in [3] it is chosen to be 9×9 pixels.

The user selects the target region $\Omega$ to be removed and filled with the new information. Once these parameters are established the filling process is automatically done. During the course of the algorithm there are some steps that need to be done. For each template window $\psi_p$ centered on the pixel $p$, and $p$ is a pixel on the border $\delta\Omega$ we compute a priority value. In the beginning the function $C(p)$ is initialized to 0 for all the pixels inside the target region and 1 for all the pixels from the source region. The priority is computed as a product of two terms, a confidence term $C(p)$ and a data term $D(p)$.

$$P(p) = C(p)D(p) \tag{1}$$

The two terms are defined as follows:

$$C(p) = \frac{\sum_{q \in \psi_p \cap (I-\Omega)} C(q)}{|\psi_p|}, \tag{2}$$

$$D(p) = \frac{|\nabla I_p^{\perp} \cdot n_p|}{\alpha}, \tag{3}$$

where $|\psi_p|$ is the area of the $\psi_p$, $\alpha$ is a normalization factor, $n_p$ is a unit vector orthogonal to the front $\delta\Omega$ in the point $p$, $\nabla I_p^{\perp}$ is the isophote (direction and intensity) at point $p$, and $\perp$ denotes the orthogonal operator. After computing all the priorities of the windows centered on the pixels of the border $\delta\Omega$, the window with the highest priority $\psi_{\hat{p}}$ is the chosen one to be filed first. Then we have to search for the most similar information to be filed in the most priority window. The search is made in the source region $\Phi$ for a window which is the most similar by computing:

$$\psi_{\hat{q}} = \arg \min_{\psi_q \in \Phi} d(\psi_{\hat{p}}, \psi_q) \tag{4}$$

where the distance $d(\psi_a, \psi_b)$ between two generic windows is defined as a sum of square differences for the already filled pixels in the two windows.

After finding the source window $\psi_{\hat{q}}$ each pixel from this window $p'|p' \in \psi_{\hat{p}} \cap \Omega$ is copied from the corresponding position from $\psi_{\hat{q}}$ into the $\psi_{\hat{p}}$.

The last step is to update the confidence term for the filled pixels in $\psi_{\hat{p}}$:

$$C(p) = C(\hat{p}) \quad \forall p \in \Psi_{\hat{p}} \cap \Omega \tag{5}$$

The algorithm repeats until the entire target region is filled.

## III. OVERVIEW OF FUZZY SETS

Fuzzy sets were introduced as extensions of the classical notation of a set. In the classical set the elements have membership that is associated with a binary term (an element can belong to the set or cannot belong to a set). In fuzzy set theory an element can belong in a certain degree to a fuzzy set and more than this it can belong to more than one fuzzy set and in several membership degrees. Fuzzy set theory was introduced by the mathematicians L.Zadeh [8].

A fuzzy set is a class of objects with continuum grades of membership. Such a set is characterized by a membership function which assigns to each object a grade of membership ranking between zero and one.

Let **X** be a space of points, with the generic element of X denoted by $x$. A fuzzy set $\boldsymbol{A}$ is characterized by a membership function which associates with each point in X a real membership number in the interval [0, 1], with the value representing the membership degree of $x$ in $\boldsymbol{A}$**.** The maximum of the membership degree is 1 and the minimum is 0.

## IV. THE FUZZY SETS GENERALIZATION OF EXEMPLAR-BASED IMAGE INPAINTING

In the inpainting algorithm for the texture propagation using a fuzzy window we will obtain an improved result, a new improved image. The reason of using a fuzzy window is the consideration of a much larger space region in a sub unitary manner which should give us a more "natural" result image than before; this approach is compatible with the human perception of the spatial regions. This is a generalization of the crisp inpainting algorithm which can be seen as a particular method, as a result of the replacement of the fuzzy window with a crisp window.

Using the same way to fill the target region as in the exemplar-based texture synthesis algorithm for obtaining a smooth texture we used a fuzzy window for our algorithm.

We have adopted the same notation like in [3] and similar to the one used in the literature. The region to be filled, the target region is denoted as $\Omega$ and its contour as $\delta\Omega$. The contour delimits the region with information of the region without information (the target region) which in the next steps will be filled. The contour changes with the evolution of the algorithm. The source region $\Phi$ is the region that contains information considered to be information with maximum confidence and which is used for the filling process.

We will present all the steps for one iteration of the algorithm for showing the evolution of the texture for the exemplar-based texture synthesis.

Let us further denote by $\psi_p \in \Omega$ the window centred in the location **p**, whose contents needs to be partially replaced in the inpainting process. In the source region we look for a window with the same size and which is the most plausible window that contains the information to fill the pixels without information from $\psi_p$, the window centered on the contour. If the best window for filling is $\psi_{\hat{q}} \in \Phi$, than the filling is made for the pixels without information from the window $\psi_p$ with the corresponding pixels from the window $\psi_{\hat{q}}$.

For a given image, the user selects the target region $\Omega$ to be filled with new information. We have the region $\Phi$ which is the entire image minus the target region.

The next step is to choose the size of the window $\psi$. In the case of the crisp exemplar-based algorithm the window $\psi$ can be seen as a bi-dimensional crisp membership function where all the pixels in the window have the membership degree of 1 and all the pixels outside the window have the membership degree of 0. In the case of a fuzzy membership function $\tilde{\psi}$ there are pixels which have the membership degree of 1, and the membership degree of 0, and other membership degrees between 0 and 1. The bi-dimensional fuzzy window with the membership function $\tilde{\psi}:\Re \rightarrow [0,1]$, where $\Re$ is a finite spatial region in the image. Choosing the size of the fuzzy window $\tilde{\psi}$ supposes the specification of the prototype spatial region of the window (the region $\Re_p$ which satisfies $\tilde{\psi}(\Re_s) > 0$ ). In the proposed algorithm, the shape of the membership function is linear on the portions (this is a bi-dimensional generalization of a isosceles trapeze, an equilateral quadrangle truncated pyramid). By choosing the size of the window we allow this window to be bigger than the texture element which can be distinguished by the user. After choosing these parameters the algorithm makes the next steps automatically.

The algorithm considers that all the pixels in the source region are known values, and the pixels in the target region are completely unknown. Also all the pixels in the image have a value of confidence, which reflects the degree of confidence in the color pixel, this value is 1 for all the pixels in the source region and remains unchanged throughout the algorithm. For pixels in the target region, the initial degree of confidence is 0, and then it changes with the filling of the area/pixel. In each cycle of the algorithm, the windows centered over the contour, also have a temporary priority value, which determines the order in which they are filled. For all pixels in the target area the algorithm follows the next steps.

For each window centered on the contour the computation of the priority term is made. Then the search for the window with the greatest priority is made. Once this window is found the search for the similar window is made. Then the pixels are copied form the corresponding position in the window but only for the pixels with no information. Then the update for the confidence term for the filled pixels is made.

Let us consider a fuzzy window $\tilde{\psi}_p$ centered in the pixel $p$ for $p \in \delta\Omega$ . For this window the priority $\tilde{P}(p)$ is:

$$\tilde{P}(p) = \tilde{C}(p) \cdot D(p) \tag{6}$$

The priority $\tilde{P}(p)$ is computed for each window centered on the contour using separate fuzzy windows for each pixel from the border of the target area.

Let us denote by $\tilde{C}(p)$ the fuzzy confidence term defined similar as in the crisp case (2) as follows:

$$\tilde{C}(p) = \frac{\sum_{q \in S(\tilde{\psi}_p) \cap (I \setminus \Omega)} \tilde{C}(q) \cdot \tilde{\psi}_p(q)}{|\tilde{\psi}_p|} \tag{7}$$

where $S(\tilde{\psi}_p)$ is the support region of the fuzzy membership function $\tilde{\psi}_p$ (which is a crisp membership function), $\tilde{\psi}_p(q)$ is the membership degree of the pixel $q$ to the bi-dimensional membership function which describes the fuzzy window, $|\tilde{\psi}_p|$ is the cardinal of the fuzzy membership function $\tilde{\psi}_p, |\tilde{\psi}_p| = \sum_{q \in \Re_s} \tilde{\psi}_p(q), with\ \Re_s = S(\tilde{\psi}_p)$ .

The function $\tilde{C}(p)$ is at the beginning initialized by: $\tilde{C}(p) = 0, \forall p \in \Omega$ , and $\tilde{C}(p) = 1, \forall p \in I - \Omega$ .

The confidence term $\tilde{C}(p)$ can be seen as a measure of confidence of the amount of information around the pixel $p$. The intention is to complete first those windows that have most of their pixels already filled, with the preference given by the pixels that have already been filled earlier (or were not part of the target region).

The term $\tilde{C}(p)$ gives the order of completing as desired. With the advancement of front because of the completion, the outer parts with fewer pixels to fill using of the confidence term will be completed first, and pixels in the center of the target region will be of lower confidence and will be completed later.

The data term *D(p)* is defined exactly as (3) as in the crisp algorithm. The term *D(p)* is a function of the strength of the isophote lines to reach the contour $\delta\Omega$ at every iteration. This gives the higher priority to a window where an isophote "encounters". This factor is of fundamental importance in the algorithm, because it encourages the linear structures to be synthesized first and then propagated in the target area safely.

The gradient $\nabla I_p$ is computed as the maximum gradient of the image in $S(\tilde{\Psi}_p) \cap I$ .

After computing all the priorities on the contour the fuzzy window with the highest priority is found and is denoted by $\tilde{\psi}_{\hat{p}}$ . Then, the filling is made with data extracted from the source region $\Phi$.

The algorithm propagates the texture by sampling direct from the source region. Thus we seek in the source region the most similar fuzzy window with the fuzzy window $\tilde{\psi}_{\hat{p}}$ is:

$$\psi_{\hat{q}} = \arg \min_{\tilde{\psi}_q \subseteq \Phi} d_f(\tilde{\psi}_{\hat{p}}, \tilde{\psi}_q) \tag{8}$$

where $d_f(\cdot,\cdot)$ is the distance between two fuzzy windows defined as fallows. Let $\tilde{\psi}_a$ and $\tilde{\psi}_b$ to be two fuzzy windows centered in the spatial location *a* respectively *b* described by their bi-dimensional membership functions $\tilde{\psi}_a : \Re_a \rightarrow [0,1]$ and $\tilde{\psi}_b : \Re_b \rightarrow [0,1]$ with $\Re_a$ =the spatial region of $\tilde{\psi}_a$ and $\Re_b$=the spatial region of $\tilde{\psi}_b$. Let $x_q = [R_a\ G_a\ B_a]^T$ be a pixel described by his color in the RGB color space at spatial location q in the image then

$$d_{f(}\tilde{\psi}_a, \tilde{\psi}_b) = \sum_{\substack{q \in S(\tilde{\psi}_a) \\ \cap (I-\Omega)}} (x_q - x_{q-a+b})^T (x_q - x_{q-a+b}) \min\{\tilde{\psi}_a(q), \tilde{\psi}_b(q-a+b)\} \tag{9}$$

As it can be seen if the membership functions $\tilde{\psi}_a = \tilde{\psi}_b$ (as we consider in the proposed algorithm) the expression of the fuzzy distance simplifies as follows:

$$d_{f(}\tilde{\psi}_a, \tilde{\psi}_b) = \sum_{\substack{q \in S(\tilde{\psi}_a) \\ \cap (I-\Omega)}} (x_q - x_{q-a+b})^T (x_q - x_{q-a+b}) \tilde{\psi}_a(q) \tag{10}$$

because $\tilde{\psi}_a(q) = \tilde{\psi}_b(q-a+b)$

After we have found $\tilde{\psi}_{\hat{q}}$ the source window the value for each pixel to be filled $p' | p' \in \tilde{\psi}_{\hat{p} \cap \Omega}$ for which $\tilde{\psi}_{p \cap \Omega}(p') = 1$ the pixel is copied from the corresponding position from the window $\tilde{\psi}_{\hat{q}}$ in the window $\tilde{\psi}_{\hat{p}}$.

After all the spatial positions for which $\tilde{\psi}_{\hat{p}}(p')=1$ for the windows $\tilde{\psi}_{\hat{p}}$ were completed with the new pixels value the confidence terms C(p) is updated as follows:

$$\tilde{C}(p)=\tilde{C}(\hat{p}) \qquad \forall p\in \tilde{\Psi}_{\hat{p}}\cap\Omega,\ if\ \psi_{\hat{p}}\cap\Omega(p)=1 \qquad (11)$$

This update allows us to keep a measure of confidence in the already filled pixels.

In this algorithm the fuzzy window that processes the image is meant to give a weight to the neighboring pixels that will not be copied in the currently processed window. The fuzzy window form is illustrated in Figure 1, for different values of the parameters window (the generic fuzzy window $\tilde{\psi}$ is described by a membership function by the type of a regular quadrangular truncated pyramid; the parameters that should be define for the specifications of the functions are the large and the small base of a truncated pyramid). The window is made from a crisp part of pixels that will be processed and another part made by fuzzy weights for a number of pixels for which information is important to a certain manner.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0,5 | 0,5 | 0,5 | 0,5 | 0,5 | 0 |
| 0 | 0,5 | 1 | 1 | 1 | 0,5 | 0 |
| 0 | 0,5 | 1 | 1 | 1 | 0,5 | 0 |
| 0 | 0,5 | 1 | 1 | 1 | 0,5 | 0 |
| 0 | 0,5 | 0,5 | 0,5 | 0,5 | 0,5 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*(a)*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0,3 | 0,3 | 0,3 | 0,3 | 0,3 | 0,3 | 0,3 | 0 |
| 0 | 0,3 | 0,6 | 0,6 | 0,6 | 0,6 | 0,6 | 0,3 | 0 |
| 0 | 0,3 | 0,6 | 1 | 1 | 1 | 0,6 | 0,3 | 0 |
| 0 | 0,3 | 0,6 | 1 | 1 | 1 | 0,6 | 0,3 | 0 |
| 0 | 0,3 | 0,6 | 1 | 1 | 1 | 0,6 | 0,3 | 0 |
| 0 | 0,3 | 0,6 | 0,6 | 0,6 | 0,6 | 0,6 | 0,3 | 0 |
| 0 | 0,3 | 0,3 | 0,3 | 0,3 | 0,3 | 0,3 | 0,3 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*(b)*

*Figure 1 Some examples for the fuzzy windows (a)size 5×5; (b) size 7×7.*

We chose this window to be a fuzzy window because it is important to take into consideration the surrounding pixels for finding a good match of the window and at the same time to ensure continuity for the texture. So if we provide continuity for the texture the result will be an image that is credible and seems to be a natural image.

The "weights" of the pixels in the window are fuzzy membership grades of the current pixel in the window and that can be seen as "degrees of importance" given to the pixels in filling the current plan.

One advantage of this fuzzy window is the flexibility given to the texture filling and the way of filling the target region. By using fuzzy window we can avoid the artifacts like corners of windows (squares) in the new textured region.

The way of going back from a fuzzy window to a crisp window can be made at any time as the crisp window is a particular case of the fuzzy window.

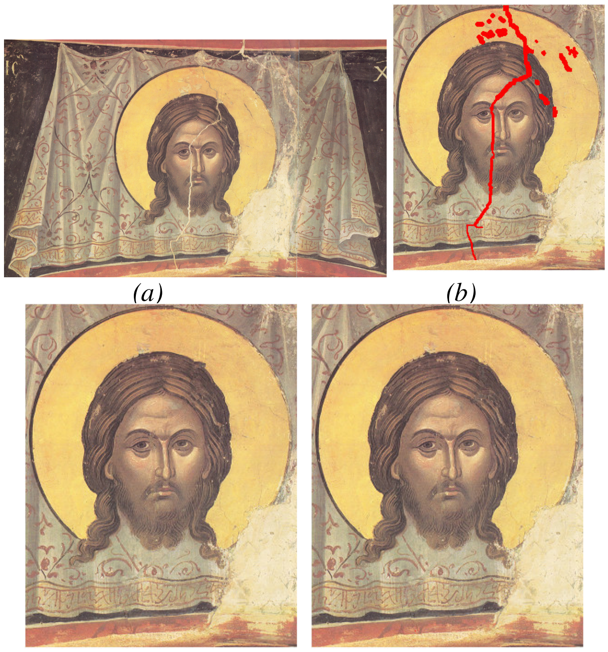
## V. IMPLEMENTATION AND RESULTS

The performance of the proposed algorithm has been evaluated and compared with the crisp algorithm.

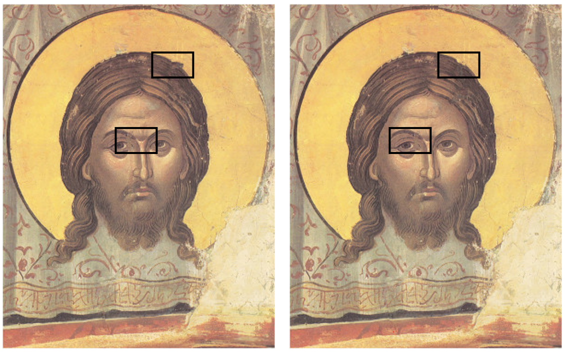
The algorithm that uses the fuzzy window for the inpainting algorithm proposed has been implemented in C++. For comparing with another algorithm we have implemented Criminisi's algorithm described in paper [3].

For testing the performance of the algorithm a set of color images have been chosen. The selection of the target region (the region that needs to be filled) is made by manual marking. The user can select the size of the fuzzy window. Some results can be seen in the Figure 2, 3, 4 and 5. The results of the crisp algorithm and our proposed algorithm are illustrated. The results for our proposed algorithm are more "natural" images compared with the result images for the crisp algorithm. In Figure 5 we have an image without a defect where we marked a small zone for filling. We know what to expect as a result. As it can be seen the result for the crisp algorithm is not good as the filling of the marked zone is a new artifact and we can see that this is a fake image. The result for our proposed algorithm is a natural image as the filling of the marked zone is one plausible and the image seems to be a natural image.

For a numerical measure unfortunately there is no standard measure for inpainting algorithms. New metrics to measure the "likeliness" or the "naturalness" of the new image are still developing.



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



*(e)* *(f)*

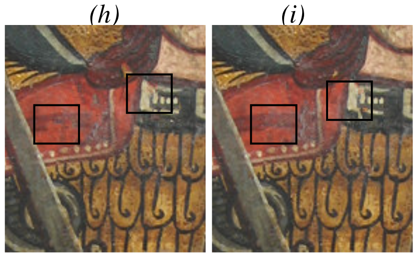
*Figure 2(a)The original image;(b) Detail of the image with the defect clearly marked;(c)The image result for the crisp algorithm in [3];(d)The result of the proposed algorithm; (e)-(f)the images (c) and (d) with the differences marked.*

*(a)* *(b)*

*(c)* *(d)*

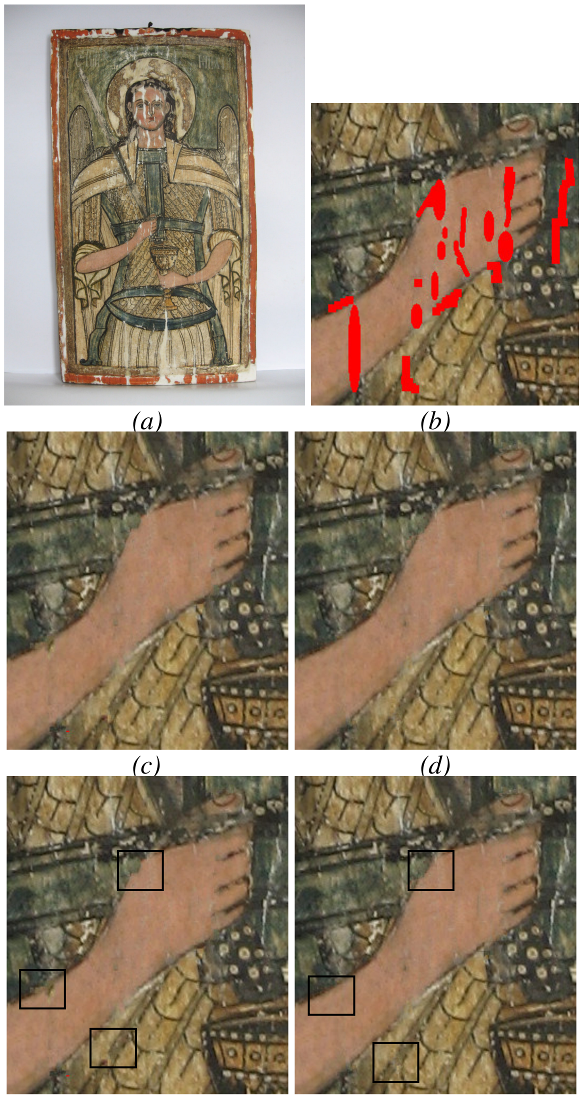
*(e)* *(f)*

*(g)* *(h)* *(i)*



*(j)* *(k)*

*Figure 3 (a)The original image;(b) Detail of the image with the defect clearly marked;(c)The image result for the crisp algorithm in [3]; (d)The result of the proposed algorithm;(e)-(f) the same images as in (c)and (d) with the differences marked (g)Another detail of the image with the defect clearly marked;(h) The result for the crisp algorithm [3];(i) The result of the proposed algorithm (j)-(k)the same images as in (h)and (i)with the differences marked.*



*(a)* *(b)*

*(c)* *(d)*

*(e)* *(f)*

*Figure 4(a)The original image;(b) Detail of the image with the defect clearly marked;(c)The image result for the crisp algorithm in [3];(d)The result of the proposed algorithm;*

*(e)-(f)the same images as (c)/(d)with the differences marked.*

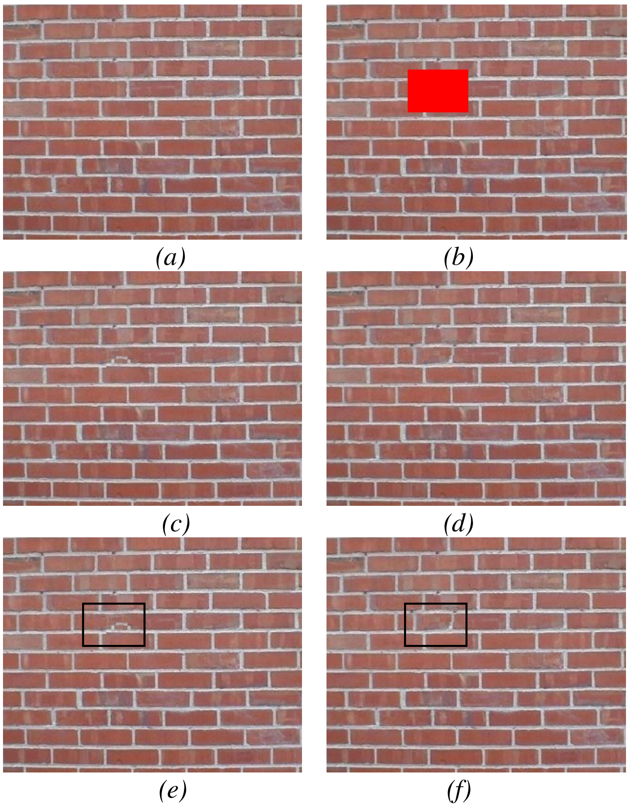



*Figure 5 The comparative results of the algorithms; (a)The original image; (b)The target region marking; (c)The result image of the Criminisi's algorithm (MSE:48.28, PSNR:31.29); (d)The result image of the proposed algorithm(MSE:46.32, PSNR:31.47 ); (e)the image in (c) with the new filled area marked; (f) the image in (d) with the new filled area marked.*

## VI. CONCLUSIONS

In this paper we presented a fuzzy set generalization of the exemplar-based image inpainting. We proposed a new method using a fuzzy window for scanning the image and finding the most similar window for filling the window currently processed. This is an adaptive window because it takes into account the neighborhood pixels in a certain degree and in this manner the filling of the target region can be more accurate. In our approach, the experimental results show that the new method is more efficient in filling the missing part (target area) of the image.

For our future work we intend to try to improve the results by using in the inpainting algorithm an adaptive fuzzy window.

## REFERENCES

[1] P. Arias, G. Facciolo, V.Caselles, G. Sapiro, "A Variational Framework for Exemplar-Based Image Inpainting", International Journal of Computer Vision, Vol.93, Number 3, pp.319-347, 2011.

[2] M. Bertalmio, G.Sapiro, "Image inpainting" Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, pp:417-424, New York, USA, 2000.

[3] A. Criminisi, P. Pérez, and K. Toyama, "Object removal by exemplar-based inpainting," presented at the IEEE CVPR, 2003.

[4] Y. Qin, F. Wang, "A curvature constraint Exemplar-based image inpainting", International Conference on Image Analysis and Signal Processing (IASP), pp. 263 – 267, 2010.

[5] S.K. Timothy, C.Rong-Chi, C. Su-Mei, C. Yu-Ping, and T. Huan-Chi Huang, Digital Inpainting-Image Inpainting, Video Inpainting, Holes Removal on 3-D Surfaces, Summary - inpainted mechanism region information damaged picture spikes pixels, University, Taiwan.

[6] A.Wong, J.Orchard, "A nonlocal-means approach to exemplar-based inpainting", Proceedings of ICIP'2008, pp.2600-2603, 2008.

[7] L. Yang, T. Xiao-jian, W. Qing, S. Shang-xin, S. Xiao-lin, "Image inpainting algorithm based on regional segmentation and adaptive window exemplar", 2nd International Conference on Advanced Computer Control (ICACC), Vol. 5, pp. 656 – 659, 2010.

[8] L.A. Zadeh, "Fuzzy Sets*", Information and Control, No.8, pp.338-353, 1965.