Volume 52, Number 4, 2011

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

DEFECT DETECTION AND RESTORATION OF CULTURAL HERITAGE IMAGES

Mihaela CISLARIU Mihaela GORDAN Aurel VLAICU Camelia FLOREA Silvia CIUNGU SUTEU
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, Camelia.Florea@com.utcluj.ro, silviaciungusuteu@yahoo.com

Abstract: One of the major problems of the old artworks is their degradation due to aging. As a result, the digitized artworks images exhibit a series of defects. Whereas scientific methods were applied to the analysis and restoration of the priceless artworks for many decades, over recent years there has been an encouragement for the development of virtual artworks restoration tools, many of them using image processing techniques. In this paper we present a series of tools for defect detection and virtual restoration of digitized artwork images, and illustrate their application of digitized wood paintings. These tools are integrated in user-friendly semi-automatic software applications. The novel algorithms proposed and integrated in our virtual artwork restoration application include: a machine learning based small defect detection, combined with the vector median filter for the small defects removal; a fuzzy image inpainting method for large defect removal (assuming the defect area was manually selected). The results achieved with the proposed software application are presented, for some wood painted icons, showing the good performance of the proposed solution and its suitability for virtual restoration of some digitized artworks.

Keywords: *defect detection, SVM, RGB color space.*

I. INTRODUCTION

In the last few years there has been a significant focus on artwork images acquisition and processing [1]. Encouraged by the interest for cultural heritage restoration, the image processing techniques have been successfully applied to the analysis, restoration, archiving and preservation of the artworks. Among these it is worth mentioning virtual restoration techniques that aim to estimate how the artwork most likely should have been before the degradation, when originally created.

For virtual restoration of old photos, movies and digitized artworks, two steps must be addressed: the detection or identification of the defect areas/pixels (which should be done in a first step), followed by the correction of the previously identified and located defect through different image processing techniques in the second step. These techniques are almost al-ways specific to the type of defect, as no universally applying methods could be derived to correct all the defects at once. Therefore, a defect classification and analysis must be per-formed in order to associate the most suitable defect correction method to the identified defects (this might be even necessary for the defect detection phase).

In [2] the authors propose their own defect taxonomy based on the digital features of the defect and not on its origin. As they describe in the paper there are several types of defects denoted as: spots, semi-transparent spots, scratches, foxing, fold, cracks, deformation, blotches, fading, yellowing, lacking colour, lacking portions, handwriting. In our view, from the image processing perspective, considering the types of algorithms needed to virtually correct these defects, one can distinguish three categories of degradations: small size defects, causing the loss of image information in narrow spatial areas in the image (as: spots, semi-transparent spots, scratches, folds, cracks, possibly handwriting); large size defects, causing the loss of image information in larger spatial areas in the image (as blotches, lacunas, lacking portions); color degradation (as: fading, yellowing, lacking colour), in which case there is no localized loss of scene texture or spatial content, but the color palette is degraded. Whereas different solutions to the correction of these three categories of degradations exist in the literature, only the first two categories will be considered in the following, as the color degradation is outside the scope of this paper. In general, regardless the size of the defect, most existing approaches to image restoration assume two main steps. The first step aims at the detection and localization of the defective pixels, yielding a so-called defect map. The second step processes the image using the defect map and the original scene content, to replace the spatial locations indicated by the defect map with the "most plausible" information that would provide a restored image as similar as possible to an un-degraded one.

In respect to the small size defects detection, there are several approaches in the literature; one must note that in some cases – as spots, spikes, very thin scratches – noise detection algorithms may be applied as well, since (based on their size) such defects are very similar to noise. Also one must note that for the same reason, some artwork restoration solutions even skip the small size detection step [8] and apply directly some color noise filtering methods on the entire image – at the expense of some edge degradation and possibly blurring. However other approaches are more targeted to a larger class of small size defects, like e.g. [3],

Manuscript received September 1, 2011; revised November 28, 2011

where the authors propose a technique for scratch and dust removal consisting from a detection part followed by a selective color filtering in the defect removal part. In the defect detection, the grey level difference and contrast dissimilarity features are extracted and then used for the classification of pixels as belonging to a defect or not; some post-processing of the resulting defect map is also employed, to reduce the false detections, using global image information and even infrared imaging. Two types of defect maps are generated: a crisp map and a credibility map (generated by a "soft" decision). For the reconstruction, a credibility weighted bilateral filter for local repair is proposed and employed, using the credibility map generated from the defect map. Another approach to small size defects detection is presented by Gupta et al. in [4], using a morphological approach that combines several procedures; in principle, the authors propose a processing chain comprising a bottom-hat transform followed by thresholding and morphological area opening for the small cracks detection. Afterwards a modified adaptive median filter is employed to "fill" the previously detected cracks. Another approach to cracks and breaks detection and elimination in paintings is proposed by Solanki and Mahajan [5]; here the localization of cracks is done by thresholding the output of the top-hat transformed image. To eliminate some of the false detections, a semi-automatic procedure using region growing is further applied. The restoration of the identified crack pixels is done using median filters, as in most existing works devoted to small size defect correction.

In respect to the restoration of large size defects, like lacunas and blotches, the automatic solutions for the localization of the defects are significantly fewer, and this is mainly due to the difficulty of this task, as the large degraded parts of the artworks may vary in size, appearance and shape. This is why most existing approaches to large size defect localization are either manual or semi-automatic (in which case, segmentation procedures are used, as in [8]). Once the large defect region (lacuna, missing or severely degraded area) has been marked, some restoration method must be applied to fill the pixels inside this marked area with the most plausible information. This information may be inferred from the non-deteriorated regions in the current image or alternatively from other similar artwork images; in general the information needed to complete the artwork image in the marked large size defect areas is taken from one or several so-called "source images", and the procedure of filling the large size defect areas with plausible image information is called "image inpainting". Different versions of image inpainting are specifically employed in large size defect restoration. Thus, in [6], Bertalmio and Sapiro propose an inpainting algorithm that propagates information from the surrounding areas in the isophotes direction. One of the most popular inpainting approaches, applied successfully in large size lacuna filling is the exemplar based image inpainting algorithm using a block-based sampling process introduced by Criminisi et al. [7]. Actually most of the existing image inpainting approaches applied in the reconstruction of image information in the large defect areas share the mathematical frameworks introduced by Bertalmio [6] and Criminisi [7], being variations and adaptations of these algorithms to specific restoration/completion tasks. In respect to the integration of small size and large defect detection and removal tools for the virtual restoration of artwork images, this is as well a non-trivial task. Obviously, the significant difference in the size and aspect ratio of the damaged regions (in the sense of the number of pixels in a spatially continuous damaged region as well as its compactness) requires different methods for the localization and correction of the two types of defects: the small size defects are more similar to noise, therefore modified versions of noise detectors and color noise filtering techniques may be applied for their removal. On the other hand, the large size defects cannot be removed by spatial filtering, as no sufficient information is available in the spatial surrounding of the pixels inside the large size defect area. The best solution to this case is image inpainting. Therefore, for a virtual restoration tool of the of artwork images to be useful, it should integrate methods and algorithms covering all the above mentioned cases. Furthermore, having in mind that different approaches and different parameters settings can lead to significantly, as flexible/parametized as possible, and- last but not least- should implement options to support the human interaction in a user friendly manner. Currently there are only a few such solutions reported; an example is the ArtShop tool, developed by Cappellini et al. [8]. Artshop is an art-oriented image-processing tool for cultural heritage applications. It includes several tools for small and large size correction and color calibration procedures but the detection of the defect is not actually included.

In this paper we present some tools for virtual artworks restoration and their integration in a Windows software application designed in a user-friendly manner, which allows for a flexible user interaction in the defect detection/localization and defect correction steps. Although the implemented software tool may be used for degraded digital images in general, it was mainly intended to be applied on wood paintings/old icons. Therefore the results are primarily presented for such cases. The application comprises two modules: one is devoted to the mall size defects detection and devoted to small size defects detection and correction, and the other- to the large size defects detection and correction (by a proposed fuzzy image inpainting strategy). The selection of the area to be inpainted (considered to be a large size defect that needs replacement with plausible information) is done manually by the user from the application's interface, using the mouse. The details of the proposed tools for visual restoration the software tool and some experimental results are presented in the following.

II. SMALL SIZE DEFECTS DETECTION AND REMOVAL

For the removal of small size defects that may appear in digital artwork images, such as dust, scratch, cracks, we propose a novel detection tool, followed by a standard approach to their removal (using the well known vector median filter in the RGB color space [9]). However it worth noting that other types of color image filtering may be applied on the detected defective pixels as well, as e.g. our adpative cascaded fuzzy inference systems based filter [10]. In general, an accurate detection of the defect area is the most difficult and crucial part for a good removal of the small size defects, while preserving the other image details; this is the reason of emphasizing here the detection part only. In the proposed solution, we formulate the small size defect detection as a pixel classification problem into "defective" or "not defective". Therefore this becomes a binary classification task, and for solving it we propose to employ support vector machine (SVM) classifiers [11], considering their remarkable performance in pattern classification and their ability to learn with good generalization from a sparse set of training samples (which is expected to be the case in virtual artwork image degradation, as not many manually localized defects examples may be available, especially in the case of small size defects). A simple yet discriminative feature for the discrimination of the pixels belonging to a small size defect (as dust, scratch, crack defects) from the non-degraded image pixels is the joint feature introduced by Bergman et al. in [3], which basically measures the intensity and local contrast dissimilarity between the original image and a median filtered version of the image (called by the authors "detail-less" image). Therefore we decide to use similar features as pixel descriptors in order to classify them as belonging to a small size defect or not. However, in virtual artworks images, the defect may affect each of the three color components: red, green and blue, therefore extracting such features from the three color components would yield a better representation of the pixels for their further classification. Thus, unlike in [3] where only the luminance channel is analyzed, we extract the intensity and local contrast difference in each color channel separately, compute their product in each channel and build a three-dimensional feature vector in each pixel location. The resulting feature vector is used in the SVM classification process (unlike in [3], where a simple thresholding is applied). Among several examined SVM classifier configurations, we choose (based on the performance in the training and validation set in terms of accuracy and generalization) to use a non-linear SVM with Gaussian RBF kernel [11].

The extraction of the features is done as follows. Let us consider the digital color image represented in the RGB color space by the matrices of the intensities of the three color channels, $\mathbf{I}_k[H \times W], k = R, G, B$, with H and W – the image height and width in pixels. On each of the three color channels a median filtering is applied, in an $M \times M$ spatial filtering window, which in our approach was set to M=5, since it proved optimal for our artwork images. We denote the median filtered color channels by $\mathbf{I}_{mf,k}[H \times W], k = R, G, B$. In each spatial location (i, j) in the original image ($i = 0,1,...,H-1; j = 0,1,...,W-1$), the corresponding intensity in the median filtered channel R, G and B, $\mathbf{I}_{mf,k}(i, j), k = R, G, B$, is obtained as the median value of the ordered string of M^2 intensities read from the spatial window centred on (i, j) on the channel k. Using the matrices $\mathbf{I}_k[H \times W]$ and $\mathbf{I}_{mf,k}[H \times W], k = R, G, B$, two types of descriptive features of the pixels found in a small size defect region are computed. The first feature is simply the absolute value of the intensity difference in the current spatial location (i, j) in each color channel k, k=R, G, B:

$$\begin{aligned} &\mathbf{I}_{dif,k}(i, j) = \left|\mathbf{I}_k(i, j) - \mathbf{I}_{mf,k}(i, j)\right|, \\ &\forall i = 0,1,...,H-1, j = 0,1,...,W-1, k = R, G, B. \end{aligned} \tag{1}$$

Since the median filtered color channels have fewer high frequency details than the original color channel, the feature described by (1) will have larger values for the pixels found in the small size defects as spots, thin lines/cracks and smaller values in the non-degraded pixels found in approximately uniform areas. However, large values may also be expected around the edges and in textured areas, therefore an extra feature is needed to distinguish between the two. This second feature is the difference of the local contrast computed in the original and median filtered color channel, represented by the matrices $\mathbf{C}_{dif,k}[H \times W], k = R, G, B$ and defined as follows:

$$\mathbf{C}_{dif,k}(i, j) = \frac{(\sigma_k(i, j) - \sigma_{mf,k}(i, j))^2}{(\sigma_k(i, j))^2 + (\sigma_{mf,k}(i, j))^2 + C}, \tag{2}$$

where: $\sigma_k(i, j)$ is the standard deviation of the intensities in the k color channel (R, G or B) in an $M \times M$ pixels spatial window centred around the current location (i, j), $i = 0,1,...,H-1; j = 0,1,...,W-1$; $\sigma_{mf,k}(i, j)$ is the standard deviation of the intensities in the median filtered version of the k color channel (R, G or B) in an $M \times M$ pixels spatial window centred around the current location (i, j), $i = 0,1,...,H-1; j = 0,1,...,W-1$; C is a small constant introduced to avoid division by zero (a simple choice would be C=1). Let us consider the typical case of an odd-sized spatial pixels window, M – odd, and denote by M_2 – the integer half of this window, $M_2 = (M-1)/2$. Then $\sigma_k(i, j)$ and $\sigma_{mf,k}(i, j)$ are computed as:

$$\sigma_k(i, j) = \frac{\sqrt{\sum_{p,q=-M_2}^{M_2} (\mathbf{I}_k(i+p, j+q) - \mu_k(i, j))^2}}{M^2 - 1},$$

$$\text{with } \mu_k(i, j) = \frac{\sum_{p,q=-M_2}^{M_2} \mathbf{I}_k(i+p, j+q)}{M^2}, \tag{3}$$

and similarly for $\sigma_{mf,k}(i, j)$, if we replace the index *mf,k* for the index k in (3).

The intensity and local contrast difference in each color channel have better chances together to separate the pixels in the small size defects from the clean image pixels; as also considered in [3], we combine the two types of features by the arithmetic product, thus yielding, for each pixel in the spatial location (i, j), a vector representation $\mathbf{v}_{(i,j)} \in \Re^3$:

$$\begin{aligned} &\mathbf{v}_{(i,j)} = \begin{bmatrix} v_{(i,j)}^{(1)} \\ v_{(i,j)}^{(2)} \\ v_{(i,j)}^{(3)} \end{bmatrix}, v_{(i,j)}^{(k)} = \mathbf{I}_{dif,k}(i, j) \cdot \mathbf{C}_{dif,k}(i, j), \\ &\forall k = R, G, B, i = 0,1,...,H-1, j = 0,1,...,W-1. \end{aligned} \tag{4}$$

Each image pixel is represented in the feature space defined by (4) and input to the binary SVM classifier decision function previously trained from a small sample image in which different types of small size defects (spots, cracks, scratches) are present and manually marked in a mask image. As a result of the classification phase, a binary map identifying the locations of the degraded pixels is obtained. Further, the classical color vector median filter is applied in these identified locations only, leading to a good defect restoration without significantly altering the original image details.

III. LARGE SIZE DEFECTS LOCALIZATION AND CORRECTION

Regarding the restoration of the image information in large size defects as gaps, stains, lacunas, we propose and implement a fuzzy generalization of Criminisi's exemplar image inpainting approach [7]. The localization of the large size defect area is manually done by the user (using the mouse) through the application's interface (as illustrated in Fig. 4 – the area to be replaced from the image is marked in red), or alternatively a previously marked image may be opened and passed directly to the fuzzy inpainting module. The proposed fuzzy inpainting is actually a generalization of the exemplar-based inpainting algorithm, using a fuzzy instead of a crisp window in the block search step. The motivation for using a fuzzy instead of a crisp window in the best match searching process is the possibility it provides, to take into account more of the local context information in the matching process, which is decreasingly important as we move away from the centre of the current pixel, but still it can play an important role in selecting the best candidate exemplar window whose content will be "copied" in the current part of the defect. The results are in some cases much better and natural than those obtained with the crisp algorithm.

Let us consider the spatial region marked for replacing by Ω and its contour by δΩ. The region Ω is typically called "the target region", whereas its complement, denoted by Φ, 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). One must notice that during the evolution of the inpainting process, both Ω and its contour δΩ change (the target region decreasing eventually to zero).

Just as the crisp exemplar based image inpainting, the proposed fuzzy exemplar based image inpainting is an iterative algorithm, in which the filling process described in detail in the following is applied until the target region Ω reduces to zero.

Let us denote by **p** – the spatial location of a pixel on the boundary δΩ of the target region Ω, $\mathbf{p}=\begin{bmatrix}p_x & p_y\end{bmatrix}^T$, where p_x and p_y are the horizontal and vertical coordinates of the pixel position inside the image. Let us further denote by $\tilde{\psi}_{\mathbf{p}}$ – a fuzzy spatial window with a given support – say $M\times M$ pixels, centred in the location **p**, whose contents needs to be partially replaced in the inpainting process. For the sake of simplicity, we will actually use the notation $\tilde{\psi}_{\mathbf{p}}$ for the membership function of the fuzzy window, so that for any other spatial position $\mathbf{q}=\begin{bmatrix}q_x & q_y\end{bmatrix}^T$ in the $M\times M$ pixels neighbourhood around **q**, $\tilde{\psi}_{\mathbf{p}}(\mathbf{q})$ is the membership degree of **q** to the spatial window centred in **p:**

$$\tilde{\psi}_{\mathbf{p}}:\left[-\frac{M}{2}+p_x:\frac{M}{2}+p_x\right]\times\left[-\frac{M}{2}+p_y:\frac{M}{2}+p_y\right]\to[0;1].$$

Since the central pixel **p** of the fuzzy window is always located on the boundary of the target region and since the fuzzy window is symmetrical around **p**, in the support region of $\tilde{\psi}_{\mathbf{p}}$ we will always find pixels from both the target region Ω and the source region Φ. However the pixels from the source region Φ represent valid image information, therefore their values must be kept unchanged, and only those pixels from the current window belonging to the target region Ω should be replaced by values "copied" from another window inside the source region – that is, from the most similar window with the current $\tilde{\psi}_{\mathbf{p}}$.

Before explaining the similarity computation, one more aspect should be discussed, namely, the selection mechanism of the current pixel **p** in which the inpainting is applied. So far we just considered **p** to be "some" pixel on the boundary δΩ; however the success of the inpainting depends a lot on the filling order, i.e. on the selection mechanism of the windows centres to be first filled. In the exemplar based inpainting, it makes sense to attempt to complete the information in the spatial windows that have the fewest pixels in the target region and therefore contain the largest amount of valid image information. On the other hand, the direction of the edges (if any) in the current spatial window centred on **p** is also important, as this direction should be preserved in the inpainting process; in Criminisi's approach [7], the first to be filled are the windows centered around points **p** in which the edge direction is close to the normal to δΩ in **p**. These two criteria are mathematically expressed through a so-called confidence term and data term, and their product gives the overall priority for starting the inpainting in a certain point **p** from δΩ. In our fuzzy approach, for each candidate location **p**, we define a fuzzy confidence term denoted by $\tilde{C}(\mathbf{p})$ and we keep the definition of the data term $D(\mathbf{p})$ from [7]:

$$\tilde{C}(\mathbf{p})=\frac{\sum_{\mathbf{q}\in S(\tilde{\psi}_{\mathbf{p}})\cap(I\setminus\Omega)}\tilde{C}(\mathbf{q})\cdot\tilde{\psi}_{\mathbf{p}}(\mathbf{q})}{|\tilde{\psi}_{\mathbf{p}}|},\qquad(5)$$

where $S(\tilde{\Psi}_{\mathbf{p}})$ denotes the support region of the fuzzy window $\tilde{\psi}_{\mathbf{p}}$; $\left|\tilde{\Psi}_{\mathbf{p}}\right|$ is the cardinal of the fuzzy set $\tilde{\psi}_{\mathbf{p}}$, $\left|\tilde{\Psi}_{\mathbf{p}}\right|=\sum_{\mathbf{q}\in S(\tilde{\Psi}_{\mathbf{p}})}\tilde{\Psi}_{\mathbf{p}}(\mathbf{q})$; $\tilde{C}(\mathbf{q})$ are the confidence terms from the previous inpainting iteration.

$$D(\mathbf{p})=\frac{|\nabla I_{\mathbf{p}}^{\perp}\cdot n_{\mathbf{p}}|}{\alpha},\qquad(6)$$

where α is a normalization factor; $n_{\mathbf{p}}$ is a unit vector orthogonal to the front $\delta\Omega$ in the point **p**; $\nabla I_{\mathbf{p}}^{\perp}$ is the isophote (direction and intensity) at point **p**. The gradient $\nabla I_{\mathbf{p}}$ is computed as the maximum gradient of the image in $S(\tilde{\Psi}_{\mathbf{p}})\cap\Phi$.

Initially, the confidence terms of all the pixels in the image $\tilde{C}(\mathbf{q})$ are set to $\tilde{C}(\mathbf{q}) = 0, \forall \mathbf{q} \in \Omega$, and $\tilde{C}(\mathbf{q}) = 1, \forall \mathbf{q} \in \Phi$. These initial values are used in the first iteration to compute the $\tilde{C}(\mathbf{p})$ for all the pixels $\mathbf{p} \in \delta\Omega$ according to (5). The data terms $D(\mathbf{p})$ are computed with (6). Using these values, the overall fuzzy priority associated to each pixel $\mathbf{p} \in \delta\Omega$ is evaluated as: $\tilde{P}(\mathbf{p}) = \tilde{C}(\mathbf{p}) \cdot D(\mathbf{p})$. The pixel $\hat{\mathbf{p}}$ with the highest fuzzy priority, $\tilde{P}(\hat{\mathbf{p}}) = \max_{\mathbf{p} \in \delta\Omega} \tilde{P}(\mathbf{p})$, is the first candidate around which we center the fuzzy window and proceed with the next step of the fuzzy inpainting, that is, the search for the fuzzy window in the source image Φ yielding the highest similarity with the fuzzy window centred around $\hat{\mathbf{p}}$. The windows similarity is estimated pixel-wise, however, since not all the pixels have a high confidence in the correctness of the image information they contain, the pixels confidences $\tilde{C}(\mathbf{q})$ must also be used as weights in the similarity computation. Let $\tilde{\psi}_{\hat{\mathbf{p}}}$ be the fuzzy window to currently fill, and $\tilde{\psi}_{\mathbf{q}}$ – any fuzzy window of the same size centred in some pixel $\mathbf{q}$ such that $\tilde{\psi}_{\mathbf{q}} \subseteq \Phi$; then the most similar fuzzy window to $\tilde{\psi}_{\hat{\mathbf{p}}}$, denoted by $\tilde{\psi}_{\hat{\mathbf{q}}}$, is given by:

$$\psi_{\hat{\mathbf{q}}} = \arg \min_{\tilde{\psi}_{\mathbf{q}} \subseteq \Phi} d_f(\tilde{\psi}_{\hat{\mathbf{p}}}, \tilde{\psi}_{\mathbf{q}}). \qquad (7)$$

Here, $d_f(\cdot,\cdot)$ denotes the distance between two fuzzy windows defined as follows. Let $\tilde{\psi}_{\mathbf{a}}$ and $\tilde{\psi}_{\mathbf{b}}$ be two fuzzy windows centred in the spatial locations **a** and **b**. Let $\mathbf{x}_{\mathbf{q}} = [R_{\mathbf{q}}\ G_{\mathbf{q}}\ B_{\mathbf{q}}]^T$ be the pixel in the spatial location **q** described by its RGB color. Let **q'** denote the displacement of **q** by **a**-**b**, to account for the distance between the centres of $\tilde{\psi}_{\mathbf{a}}$ and $\tilde{\psi}_{\mathbf{b}}$, $\mathbf{q}' = \mathbf{q} - \mathbf{a} + \mathbf{b}$. Then we define:

$$d_f(\tilde{\psi}_{\mathbf{a}}, \tilde{\psi}_{\mathbf{b}}) = \sum_{\substack{\mathbf{q} \in S(\tilde{\psi}_{\mathbf{a}}) \\ \cap (I-\Omega)}} (\mathbf{x}_{\mathbf{q}} - \mathbf{x}_{\mathbf{q}'})^T (\mathbf{x}_{\mathbf{q}} - \mathbf{x}_{\mathbf{q}'}) \min\{\tilde{\psi}_{\mathbf{a}}(\mathbf{q}), \tilde{\psi}_{\mathbf{b}}(\mathbf{q}')\},$$

which reduces to a simpler form in the common case of using the same membership function shape for $\tilde{\psi}_{\mathbf{a}}$ and $\tilde{\psi}_{\mathbf{b}}$:

$$d_f(\tilde{\psi}_{\mathbf{a}}, \tilde{\psi}_{\mathbf{b}}) = \sum_{\mathbf{q} \in S(\tilde{\psi}_{\mathbf{a}}) \cap (I-\Omega)} (\mathbf{x}_{\mathbf{q}} - \mathbf{x}_{\mathbf{q}'})^T (\mathbf{x}_{\mathbf{q}} - \mathbf{x}_{\mathbf{q}'}) \tilde{\psi}_{\mathbf{a}}(\mathbf{q}). \quad (8)$$

Once the source window $\psi_{\hat{\mathbf{q}}} \in \Phi$ used as exemplar for filling the unknown part from $\tilde{\psi}_{\hat{\mathbf{p}}}$ is found, the zero confidence pixels belonging in the degree 1 to $\tilde{\psi}_{\hat{\mathbf{p}}}$ are replaced with their counterparts from $\psi_{\hat{\mathbf{q}}}$. Finally, the confidence terms are updated by:

$$\tilde{C}(\mathbf{p}) = \tilde{C}(\hat{\mathbf{p}}), \quad \forall \mathbf{p} \in \Omega \text{ and } \psi_{\hat{\mathbf{p}}}(\mathbf{p}) = 1. \quad (9)$$

With the advancement of the front $\delta\Omega$ due to the partial image completion, the outer parts with fewer pixels to fill will be completed first; pixels in the centre of the target region have lower confidence, thus will be completed later.

IV. IMPLEMENTATION AND RESULTS

The tools presented in the previous sections have been implemented in C++ in the form of a Windows application, with a user friendly graphical interface. The operation of the resulting application for some artwork wood paintings (courtesy to the Astra Museum) and the restoration results are illustrated below. Fig.1 shows the localization and correction of small size defects. The application's interface for the small size defect detection and removal is show in Fig.3 – including options for SVM classifier training for the identification of small size defects, SVM classification using a previously trained classifier, and color vector median filtering for the correction of the detected small size defect.

The result of the correction of manually marked large size defects through the proposed fuzzy image inpainting is illustrated in Fig.2. As mentioned before after the user marks the defect the filling process is done automatically. The user can set the fuzzy window size as it can be seen in the Fig.4 were the interface is illustrated for the large defect correction tool.

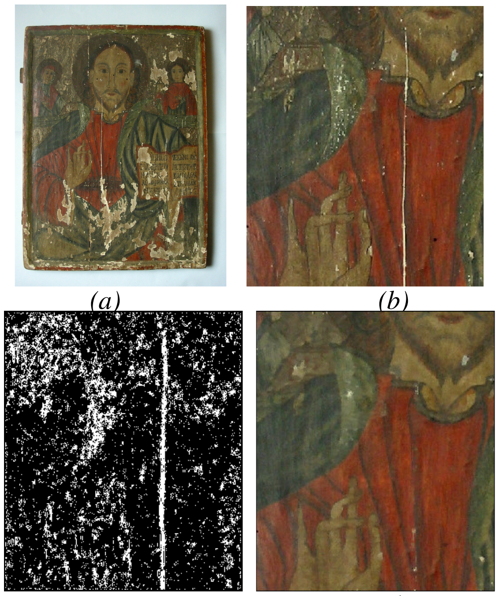
(a) (b) (c) (d)

Figure 1 (a)The original image "Christ Pantocrator" ASTRA Museum Collections, Religious artifacts, no.309; (b) Detail area of the original image; (c)The mask of the defect detected pixels; (d) The restored image after the vector median filtering of the defect.


(a) (b)

(c) (d)

Figure 2 (a), (c) Two details of the image "Archangel Michael" ASTRA Museum Collections, Religious artifacts, no.299, with the defect clearly marked; (b), (d) The results of the proposed algorithm for the images (a), (c).

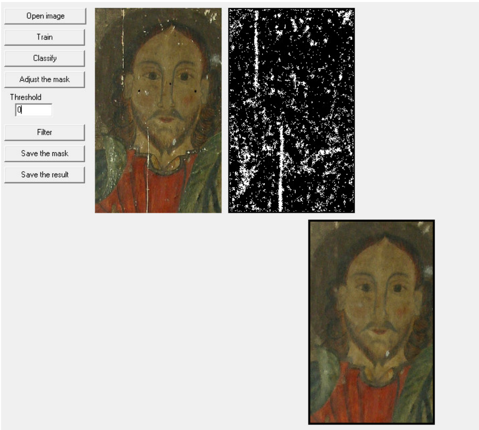

Figure 3 Our application interface for small defect detection and correction. First image is the original image; on its right we have the detection of the defect and below we have the result of the correction of the defects detected.

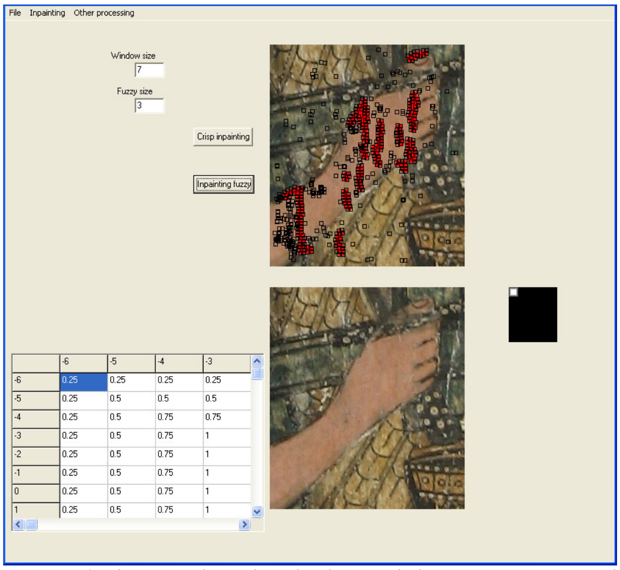

Figure 4 The interface for the large defect correction tool. Up we have a part of an image with the large defects marked by the user. Below is the result of the proposed fuzzy inpainting algorithm.

IV. CONCLUSION

We proposed and presented a series of algorithms and tools for the restoration of degraded artworks, integrated in a user-friendly Windows application, which allows to a significant extent the human interaction and parameters selection in the virtual restoration of artwork images, but in the same time automates as much as possible the defect detection and correction process. The application allows the user to experiment some image processing tools like image filtering, defect detection, virtual restoration for small defects like cracks, scratches and for large defects like missing areas, lacunas. Some results achieved with the proposed software application are illustrated on wood painted icons, showing the good performance of the proposed solution and its suitability for virtual restoration of some digitized artworks. In the future work, we aim to include other restoration tools in this framework, both devoted to small size and large size defects – among which an automatic method to large size defect localization in artworks would be a priority. Also the possible development and inclusion of color restoration tools will be considered.

ACKNOWLEDGMENT

This work has been supported by the Doctoral Studies Support Project "PRODOC", POSDRU/6/1.5/S/5 ID 7676.

REFERENCES

[1] M. Barni, A. Pelagotti, A. Piva, "Image Processing for the Analysis and Conservation of Paintings: Opportunities and Challenges", IEEE Signal Processing Magazine, vol. 22, no. 5, September 2005, pp. 141-144

[2] E. Ardizzone, H. Dindo, G. Mazzola, "Content-Based Image Retrieval as Validation for Defect Detection in Old Photos", Recent Advances in Signal Processing, Book edited by: Ashraf A Zaher, ISBN: 978-953-307-002-5, Publisher: InTech, Publishing date: November 2009

[3] R. Bergman, R. Maurer, H. Nachlieli,; G. Ruckenstein, G. Chase, D. Greig, "Comprehensive Solutions for Removal of Dust and Scratches from Images", HPL-2007-20, HP Laboratories Israel, February 2007.

[4] A.Gupta, V. Khandelwal, A.Gupta, M. Srivastava,"Image Processing Methods for the Restoration of Digitized Paintings", Thammasat Int. J. Sc. Tech Vol. 13, No. 3, pp. 66-72, 2008

[5] S. Solanki, A.R.Mahajan, "Cracks Inspections and Interpolation in digitized Artistic Picture using Image Processing Approach", International Journal of Recent Trends in Engineering, Vol. 1, No. 2, pp. 97-99, May, 2009.

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

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

[8] V.Cappellini, M. Barni, M. Corsini, A.D. Rosa, and A. Piva, "ArtShop: an art-oriented image-processing tool for cultural heritage applications", presented at Journal of Visualization and Computer Animation, 2003, pp.149-158.

[9] Koschan A., Abidi M., "A comparison of median filter techniques for noise removal in color images", University of Erlangen-Nurnberg, Institute of Computer Science, vol. 34, no. 15, pp. 69-79, October, 2001

[10] Mihaela Cislariu, Mihaela Gordan, Aurel Vlaicu, "A Novel Fuzzy Color Median Filter Based on an Adaptive Cascade of Fuzzy Inference Systems", in Leszek Rutkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh and Jacek M. Zurada(Eds.): Artificial Intelligence and Soft Computing, 10th International Conference, ICAISC 2010, Zakopane, Poland, June 13-17, 2010, Part I. Proceedings ICAISC (1), LNAI 6113 - subseries of Lecture Notes in Computer Science, pp. 27-34, Springer-Verlag, Berlin Heidelberg 2010

[11] V.N. Vapnik, "Statistical Learning Theory", J. Wiley, N.Y., 1998