AN ADAPTIVE VISUAL DICTIONARY BASED APPROACH TO OLD FILM RESTORATION

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

Abstract: One of the major problems for the old films is their degradation due to aging. Over recent years there has been an encouragement for the development of virtual restoration tools, many of them using image processing. In this paper we present a new method of defect detection and virtual restoration of digitized old films frames. The proposed algorithm includes: a machine learning based small defect detection, combined with an adaptive matching pursuit algorithm for the small defects removal. The results are presented showing the good performance of the proposed solution and its suitability for virtual restoration of some digitized films.

Keywords: *old movie, matching pursuit, fuzzy logic, logistic model trees, defect detection, defect correction.*

I. INTRODUCTION

In the last decade there has been an interest for restoring and preserving the old movies. Encouraged by the interest for old movies restoration, the image processing techniques have been successfully applied to the analysis, restoration, archiving and preservation of the old films. An important class of virtual restoration techniques aims to estimate how the old movie most likely looked like before the degradation, when originally created.

For the virtual restoration of old movies two steps must be addressed: the detection or identification of the defect areas/pixels, followed by the correction of the previously identified and located defect. The result of the detection and localization of the defective pixels yields a so-called defect map. The second step, the correction of the identified defect, uses the defect map and the original scene content to replace the spatial locations indicated by the defect map with the "most plausible" information. The goal is to provide a restored image as similar as possible to an un-degraded one. Different image processing techniques are employed in the correction step. These techniques are almost always 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 performed in order to associate the most suitable defect correction method to the identified defects (this might be even necessary for the defect detection phase).

According to the dictionary the term "defect" represents an imperfection or lack that causes inadequacy or failure, deficiency. For the visual analysis the defect of an image can be classified in different ways. For example in [1] 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 color, 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 color), in which case there is no localized loss of scene texture or spatial content, but the color palette is degraded. Different solutions to the detection and the correction of these three categories of degradations exist in the literature; in this paper we focus only on the first category.

In respect to the small size defects detection, there are sever-al 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. Some approaches are more targeted to a larger class of small size defects, like e.g. [2], 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, in [2], 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 [3], using a morphological approach that combines several procedures. In principle, the authors propose a processing chain

Manuscript received February 26, 2016; revised April 29, 2016

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 [4]; 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.

For the correction/restoration of the old movie we choose to work with the visual dictionary based representation of images, applied on non-overlapping pixels blocks. Commonly, these schemes imply the decomposition of each individual pixel block from the image as a weighted sum of the images from the visual dictionary. The visual dictionary must be at least complete in order to allow the description of each possible pixels block. However the minimum number of non-zero weights, therefore the most compact description of the pixel block, is achieved when the visual dictionary is over-complete. In such a case the resulting representation is sparse, therefore easier to handle/process further. The dictionaries may be generated from standard unitary image transforms (such as Cosine, Walsh, Haar) or may be learnt from training images. Having a visual dictionary whose elements are well correlated in terms of their brightness distribution to the image we wish to represent leads to a sparse representation of the image.

In this paper we present an algorithm for old movie restoration. Our algorithm works in two steps. The first step is the detection of the defect. The second one is an adaptive correction using visual dictionary based pixels block representation via the matching pursuit algorithm. For the first step we use a set of features of the defect which allows us to classify the image pixels by a supervised algorithm into defect or non-defect pixels. A supervised soft classification is applied, to yield a confidence map of the degradation.. The resulting confidence map is processed as a fuzzy set via its α-cuts. Each α-cut is afterwards used in conjunction with the matching pursuit algorithm for the visual dictionary based pixels blocks restoration, generating a series of possible restored blocks. For each such block, a quality factor is defined and computed, taking into account the confidence that the pixels used for the reconstruction are not from a defect region, the mean square reconstruction error and the number of clean pixels available for the block reconstruction. The restoration result is thus adaptively selected as the one giving the highest quality factor among the series corresponding to different α-cuts of the confidence degradation maps. The details of the proposed algorithm, its implementation, an illustration of the results achieved in comparison to the non-adaptive version and some conclusions are presented in the remaining of the paper.

II. SMALL SIZE DEFECT DETECTION

In order to remove small size defects from old movies such as dust, scratch, cracks, we propose a novel detection algorithm, followed by a new approach for the correction of the defects. To have a good correction, an important step is an accurate detection of the defect. This is however far from trivial, as – especially in the case of old movies – some defects like scratches and spots may easily look very similar to some fine details in the movie frames, making almost impossible the task to distinguish between the two.

In the proposed solution, we formulate the small size defect detection as a pixel classification problem into "not defective" or "defective" pixels. For solving the classification problem we propose to employ a supervised learning algorithm. There are currently many supervised classifiers available for binary pattern classification; a significant set of powerful modern learning-based classifiers is provided by the Java open-source data mining tool Weka [5], some of them able 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 [2], 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. Digitized old movies are almost always grayscale, therefore, like in [2], only the luminance channel is used for feature extraction. Similar to the approach proposed by Bergman et al. [2], we use as features for the defect detection the intensity difference and the local contrast difference between the original image and a so-called detail-less image (obtained through a median filtering of the original image). However, since the defect sizes that can appear in the old movies (scratches, spots, stains, cracks) can vary more than the case considered in [2], our experiments show that the size of the median filtering window needed to obtain a detail-less image in which the defect is filtered out cannot be fixed. Rather we chose to generate a set of detail-less images by median filtering the original image in odd-sized square spatial windows of 3×3, 5×5, 7×7, 9×9 and 11×11 pixels, and extract the two above-mentioned features (i.e., the pixel-by-pixel intensity difference and local contrast difference between the original and detail-less image) for the entire set of detail-less images thus obtained. The resulting feature vector is used in the supervised classification process (unlike in [2], where a simple thresholding is applied). Among several examined Weka classifier configurations, we choose (based on the performance in the training and validation set in terms of accuracy and generalization) to use Logistic Model Trees.

The extraction of the features is done as follows. Let us con-sider the old grayscale movie digital frame represented by the matrix of luminance values $\mathbf{I}[H \times W]$, with H and W – the image height and width in pixels. On this matrix a median filtering is applied, in an $M \times M$ spatial filtering window, yielding the so-called detailed-less image. We denote the median filtered matrix with the spatial filtering window of size $M \times M$ pixels by $\mathbf{I}_{mf,M}[H \times W]$. In each spatial location (i, j) of the image, the corresponding median filtered intensity $\mathbf{I}_{mf,M}(i, j)$ is obtained as the median value of the ordered string of M^2 intensities read from the spatial window centered on (i, j). Using the original matrix $\mathbf{I}$ and the detail-less matrix corresponding to

the $M\times M$ pixels filtering window $\mathbf{I}_{mf,M}[H\times W]$, two types of descriptive features of the pixels that would help discriminating the defect and the non-defect image regions are computed. The first feature is simply the absolute value of the intensity difference between the original and the detail-less image, computed for each of the detail-less images from the set $\mathbf{I}_{mf,M}(i,j)$, M=3, 5, 7, 9, 11*:*

$$\mathbf{I}_{dif,M}(i,j)=\left|\mathbf{I}(i,j)-\mathbf{I}_{mf,M}(i,j)\right|,\quad \forall i=0,1,...,H-1,\ j=0,1,...,W-1. \tag{1}$$

Since the median filtered matrices have fewer high frequency details than the original grayscale image, the feature described by the Equation (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, since large values are also expected around the boundaries and in the textured areas (due to the smoothing effect of the median filtering), an extra feature is needed to help distinguishing between the defect and the useful fine detail information. This second feature, introduced by Bergman et al in [2], is the difference of the local contrast computed in the original and the detail-less image. In our case, since we have a stack of five detail-less images obtained through a set of median filters with different spatial window sizes, we will have also a set of five local contrast difference features for each pixel. On the global image level, these features are represented by the set of matrices $\mathbf{C}_{dif,M}[H\times W]$, defined as:

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

where: $\sigma(i,j)$ is the standard deviation of the intensities in the $M\times M$ pixels spatial window centered around the current location (i,j); $\sigma_{mf,M}(i,j)$ is the standard deviation of the intensities in the median filtered image in the $M\times M$ pixels spatial window centered 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 (e.g. C=1). Denoting by $M_h=(M-1)/2$ - the integer part of half of the spatial window size, $\sigma(i,j)$ and $\sigma_{mf,M}(i,j)$ are computed as:

$$\sigma(i,j)=\frac{\sqrt{\sum_{p,q=-M_h}^{M_h}\sum\left(\mathbf{I}(i+p,j+q)-\mu(i,j)\right)^2}}{M^2-1},\qquad \mu(i,j)=\frac{\sum_{p,q=-M_h}^{M_h}\sum\mathbf{I}(i+p,j+q)}{M^2} \tag{3}$$

and

$$\sigma_{mf,M}(i,j)=\frac{\sqrt{\sum_{p,q=-M_h}^{M_h}\sum\left(\mathbf{I}_{mf,M}(i+p,j+q)-\mu_{mf,M}(i,j)\right)^2}}{M^2-1},\qquad \mu_{mf,M}(i,j)=\frac{\sum_{p,q=-M_h}^{M_h}\sum\mathbf{I}_{mf,M}(i+p,j+q)}{M^2}. \tag{4}$$

The intensity and local contrast difference features, extracted for the five possible odd values of M (from 3 to 11), are combined into a single feature, as suggested by Bergman [2], using a t-norm; the simplest t-norm, employed in our approach, is the algebraic product. Since for each spatial location (i,j) in the image we have a set of five intensity differences $\mathbf{I}_{dif,M}(i,j)$ and local contrast differences $\mathbf{C}_{dif,N}(i,j)$, M,N=3, 5, 7, 9, 11, their pair-wise combination by the Cartesian product of the two sets would yield a set $\left\{\left(\mathbf{I}_{dif,M}(i,j),\mathbf{C}_{dif,N}(i,j)\right)\middle|\forall M,N=3,5,7,9,11\right\}$ of 25 feature pairs. The two features of each pair are joint into a single one by their algebraic product, yielding for each image pixel from the location (i,j) in the image a vector $\mathbf{v}_{(i,j)}$ of 25 features:

$$\mathbf{v}_{(i,j)}=\left[v_{(i,j)}(1)\ \ ...\ \ v_{(i,j)}(k)\ \ ...\ \ v_{(i,j)}(25)\right],\quad v_{(i,j)}(k)=\mathbf{I}_{dif,M}(i,j)\cdot\mathbf{C}_{dif,N}(i,j),\quad k=5\cdot\left(\frac{M-1}{2}-1\right)+\frac{N-1}{2},\ \forall M,N\in\{3,5,7,9,11\}. \tag{5}$$

Each image pixel, represented by the feature vector defined by Equation (5), is input to a previously trained soft classifier. Here we used the 'logistic model trees' classifier, implemented in Weka. The classifier was previously trained from a small sample image in which different types of small size defects (spots, cracks, scratches) are manually marked. The real value of the decision function of the classifier for each pixel from the image, scaled between 0 and 1, is proportional to a confidence in the correct classification of the pixel as belonging to a defect area in the movie frame – the value zero thus corresponding to a clean, non-altered pixel, and the value of one – to a pixel that definitely comes from a defect (scratch, dust, spot) from the film. When these decision values for all the pixels is the image are represented in the form of a matrix, the result is a soft confidence degradation map corresponding to the original image $\mathbf{I}$. This degradation map, whose values are in the range [0;1], can also be regarded as a fuzzy set $\mathbf{M}_D[H\times W]$ of confidences in the membership of each image pixel (from the location (i,j)) to a defect region. In the approach we propose, in order to make effective use of $\mathbf{M}_D$ in the adaptive visual dictionary based restoration of the image $\mathbf{I}$, we need to decompose it into a set of α-cuts – each α-cut being afterwards treated as a binary degradation map, indicating by zeros the non-corrupted pixels, and by ones – the pixels that may belong to a defect region, therefore cannot be used in the reconstruction. We denote each α-cut of the fuzzy set $\mathbf{M}_D$ by $\mathbf{M}_{D\alpha}=\{(i,j)\in\{0,...,H-1\}\times\{0,...,W-1\}\mid\mathbf{M}_D(i,j)\geq\alpha\}$ and we compute the α-cuts for $0.25\leq\alpha<1$, with an increment of α of 0.05, yielding a set of 15 possible binary

defect maps. It should be mentioned that the lowest limit of α=0.25 was found to be, for most film frames, a reasonable choice, as for smaller values of α, almost all the frame pixels will be (falsely) classified as defective. This is clearly illustrated by the case in Fig. 1.(b). The increment was also chosen based on trial and error, as the smallest increment for which the difference between two consecutive α-cuts of the fuzzy set $\mathbf{M}_D$ is noticeable (visually and in terms of the MSE between the two binary maps); lowering the increment more would therefore, based on our experiments, only increase the computational complexity, without a significant benefit on the restoration. With each of these binary defect maps we apply the adaptive visual dictionary based image reconstruction algorithm presented in the next section. An example of a selected sub-set of binary defect maps obtained as the α-cuts of the fuzzy set $\mathbf{M}_D$ for an old movie frame, for $\alpha \in \{0.25,0.45,0.65,0.75,0.95\}$, is presented in Fig. 1.

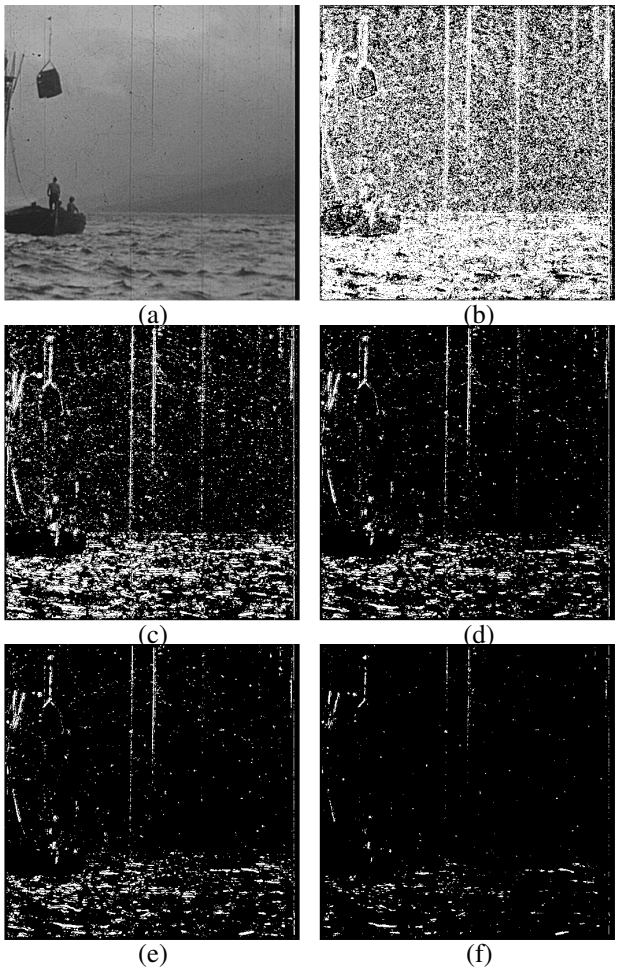


Figure 1. a) The original image, a digitized old movie frame, b)-f)The defect pixel map (white pixels represent the defect) for: b) α=0.25; c) α=0.45; d) α=0.65; e) α=0.75; f) α=0.95.

III. THE PROPOSED ADAPTIVE VISUAL DICTIONARY BASED RESTORATION PROCEDURE

Although the detection of the defect pixels is to some extent achieved by the procedure presented in the previous section, the reconstruction of the image information from the defect area is not a trivial task. Furthermore, as illustrated in Fig. 1, chances are that during the defect detection process, some image pixels found in textured regions or around the edges are mistakenly classified as defect. The preservation of these areas is hard to be done by classical recovery algorithms, like median filters, rank filters, bilateral filters or combination of these. A better recent solution is provided by the sparse image representation framework, where sparse visual dictionaries are used together with the matching pursuit to render an accurate image recovery, including with the preservation of boundary and texture information, provided the visual dictionary is well adapted to the brightness variation patterns in the image. Sparse representation has been widely used for image information recovery tasks [6-11] due to the above mention advantages.

Usually, the sparse representation of an image in terms of a visual dictionary assumes the decomposition of the image $\mathbf{I}[H\times W]$ into non-overlapping pixels blocks (usually square, of size $P\times P$, P typically an integer power of two) and the representation of each such block as a sparse linear combination of dictionary elements or so-called basis images. Statistically, the natural images admit sparse representations in terms of some dictionaries, so that only few bases will contribute to generate the image. The dictionary, which can be complete or overcomplete, may be generated from unitary image transforms such as the Discrete Cosine Transform (DCT) and Wavelets (which are well adapted to the natural image model). In our approach, we use an overcomplete DCT dictionary.

Mathematically, the representation of $\mathbf{I}$ as a set of non-overlapping pixels blocks of $P\times P$ pixels can be written as: $\mathbf{I} = \left\{\mathbf{I}_{pq}\left[P\times P\right] \mid p = 0,...,\left\lfloor (H-1)/P \right\rfloor, q = 0,...,\left\lfloor (W-1)/P \right\rfloor\right\}$, where (p,q) indicates the position of each block in the image. The visual dictionary contains a set of basis images of size $P\times P$ each: $\mathbf{D} = \left\{\mathbf{D}_k\left[P\times P\right] \mid k = 0,1,..., K-1\right\}$, where K is the number of basis images in the visual dictionary; for overcomplete dictionaries, $K>P^2$. The visual dictionary based representation of $\mathbf{I}$ implies the decomposition of each block $\mathbf{I}_{pq}$ as a weighted sum of basis images $\mathbf{D}_k$, with some approximation up to a small error in the form:

$$\tilde{\mathbf{I}}_{pq} = \sum_{k=0}^{K-1} w_{pq}(k)\mathbf{D}_k, \text{ with } \mathbf{I}_{pq} \cong \tilde{\mathbf{I}}_{pq}, \tag{6}$$

that is, $\tilde{\mathbf{I}}_{pq}\left[P\times P\right]$ is the approximation of $\mathbf{I}_{pq}$ up to some error. In the expression above, each $w_{pq}(k) \in \Re$ represents the weight by which the basis image $\mathbf{D}_k$ contributes to the approximation of the current block $\mathbf{I}_{pq}$. In practice, many of the coefficients $w_{pq}(k)$ are zero or very close to zero, therefore the decomposition is sparse.

The coefficients $w_{pq}(k)$ are found by applying some matching pursuit (MP) algorithm [7, 12]. Various versions of MP algorithms are available in the literature [13], but most of them rely on Mallat's basic version. In principle, MP implies an iterative procedure, starting from a rough approximation of the pixels block $\mathbf{I}_{pq}$ by the most similar basis image $\mathbf{D}_k$, from which more and more accurate

approximations are generated by adding the next most similar visual dictionary basis images, until the convergence is reached. The similarity is defined by the inner product-like operation between a residual matrix $\Delta\mathbf{I}_{pq}$ (computed as the difference between the original block $\mathbf{I}_{pq}$ and its approximation in the previous iteration) and the basis image from the dictionary.

To apply the above described image representation framework in image restoration, one should consider that, together with the pixels block from the original image, $\mathbf{I}_{pq}$, a binary mask block $\mathbf{M}_{pq}$ of the same size as $\mathbf{I}_{pq}$ ($P\times P$ pixels) is given. Each element of the mask is an indicator of whether the current pixel carries reliable, un-corrupted image information or not. The recovery of the image information can then be treated as the problem of estimating "plausible" values for the corrupted pixels in the pixels block $\mathbf{I}_{pq}$, based on the uncorrupted pixels in the block only. From the visual dictionary based representation perspective, this is equivalent to an approximation of the pixels block $\mathbf{I}_{pq}$, described only by its un-corrupted pixels, as a weighted sum of the basis images from the visual dictionary **D**, using a modified version of the MP algorithm which allows to disregard the corrupted pixels in $\mathbf{I}_{pq}$. This can be done by "masking out" as indicated by $\mathbf{M}_{pq}$ the corrupted pixels in both $\mathbf{I}_{pq}$ and each residual $\Delta\mathbf{I}_{pq}$ at every iteration, thus performing the computation of the dot products, the error and the residual solely in terms of the un-corrupted pixels.

In the proposed approach, as described in the previous section, from the fuzzy defect map $\mathbf{M}_D$ of the whole image **I** we generate a set of 15 binary defect maps $\{\mathbf{M}_{D\alpha}|\alpha=0.25;0.3;...;0.95\}$ as the α-cuts of this fuzzy set, with the following binary encoding:

$$\mathbf{M}_{D\alpha}(i,j)=\begin{cases}1, & \text{if } \mathbf{I}(i,j)-\text{corrupted (defect)}\\ 0, & \text{otherwise}\end{cases}, \quad (7)$$
$$i=0,1,...,H-1; \quad j=0,1,...,W-1.$$

and accordingly, for each pixel block $\mathbf{I}_{pq}$ from the image, we will have a set of 15 binary defect block maps $\{\mathbf{M}_{D\alpha,pq}\}$. The binary complement of each of these maps, $\mathbf{M}_{C\alpha,pq}=1-\mathbf{M}_{D\alpha,pq}$, will indicate by ones the values in the pixels block that carry reliable information, and by zeros – the ones suspected to be corrupted, thus not taken into account in the block recovery using visual dictionary decomposition by matching pursuit.

With the above notations, the MP-based visual dictionary restoration algorithm for each block of pixels in the image, $\mathbf{I}_{pq}$, having a set of previously determined 15 corresponding binary masks as indicators of un-corrupted pixels $\{\mathbf{M}_{C\alpha,pq}\}$, can be summarized as follows.

Step 1. Set the approximation error threshold ε.
For each $\alpha\in\{0.25;0.3;...;0.95\}$, with a step of 0.05, do:

Step 1.1. MP initialization:
Set the initial residual to the masked block of pixels, $\Delta\mathbf{I}^{(0)}_{\alpha,pq}(i,j)=\mathbf{I}_{pq}(i,j)\,\mathbf{M}_{C\alpha,pq}(i,j), i,j=0,1,...,P-1.$
Set the initial coefficients, $w^{(0)}_{pq}(k)=0, k=0,1,...,K-1$.
Set the iteration step to $t=1$.

Step 1.2. Find the basis image $\mathbf{D}_s$ most similar to $\Delta\mathbf{I}^{(t-1)}_{\alpha,pq}$:

$$s=\arg\max_{k=0,1,..,K-1}\left\langle\Delta\mathbf{I}^{(t-1)}_{\alpha,pq},\mathbf{D}_k\right\rangle^2,$$

where $\left\langle\Delta\mathbf{I}^{(t-1)}_{\alpha,pq},\mathbf{D}_k\right\rangle$ denotes the dot product-like between $\Delta\mathbf{I}^{(t-1)}_{\alpha,pq}$ and $\mathbf{D}_k$, and $\Delta\mathbf{I}^{(t-1)}_{\alpha,pq}$ denotes the residual at the iteration step *t*-1.
Update the corresponding coefficient corresponding to the current iteration step *t*, $w^{(t)}_{pq}(s)$:

$$w^{(t)}_{pq}(s)=w^{(t-1)}_{pq}(s)+\left\langle\Delta\mathbf{I}^{(t-1)}_{\alpha,pq},\mathbf{D}_s\right\rangle.$$

Step 1.3. Update the residual for the current iteration:

$$\Delta\mathbf{I}^{(t)}_{\alpha,pq}=\Delta\mathbf{I}^{(t-1)}_{\alpha,pq}-\left\langle\Delta\mathbf{I}^{(t-1)}_{\alpha,pq},\mathbf{D}_s\right\rangle\mathbf{D}_{s,M},$$

where $\mathbf{D}_{s,M}$ is the masked basis image, computed as:
$\mathbf{D}_{s,M}(i,j)=\mathbf{D}_S(i,j)\,\mathbf{M}_{C\alpha,pq}(i,j), i,j=0,1,...,P-1.$

Step 1.4. Check for convergence (i.e., if the residual energy is under the error limit ε):
If $\left\langle\Delta\mathbf{I}^{(t)}_{pq},\Delta\mathbf{I}^{(t)}_{pq}\right\rangle\le\varepsilon$, go to *Step 1.5*, else increment the iteration step *t*=*t*+1, go to *Step1.2*.

Step 1.5. Set the final coefficients $w_{pq}(k)$ for the current α to: $w_{pq}(k)=w^{(t)}_{pq}(k)$.
Reconstruct the pixels block for the current α: $\mathbf{I}^{res}_{\alpha,pq}=\sum_{k=0}^{K-1}w_{pq}(k)\,\mathbf{D}_k$.
Add the restored block to the set $I^{res}_{pq}=\{\mathbf{I}^{res}_{\alpha,pq}|\alpha=0.25;0.3;...;0.9;0.95\}$.
Compute and store the mean square error, $MSE_{\alpha,pq}$, between the un-corrupted pixels of the original block, $\mathbf{I}_{M,pq}(i,j)=\mathbf{I}_{pq}(i,j)\mathbf{M}_{C\alpha,pq}(i,j)$, and the corresponding pixels of the reconstructed (restored) pixel block $\mathbf{I}^{res}_{\alpha M,pq}(i,j)=\mathbf{I}^{res}_{\alpha,pq}(i,j)\mathbf{M}_{C\alpha,pq}(i,j), i,j=0,1,...,P-1$, for each value of α (this will be further used in the adaptation step of the proposed restoration algorithm):

$$MSE_{\alpha,pq}=\frac{\sum_{i=0}^{P-1}\sum_{j=0}^{P-1}\left(\mathbf{I}^{res}_{\alpha M,pq}(i,j)-\mathbf{I}_{M,pq}(i,j)\right)^2}{\sum_{i=0}^{P-1}\sum_{j=0}^{P-1}\mathbf{M}_{C\alpha,pq}(i,j)}. \quad (8)$$

Add the mean square error values to the set $MSE_{pq}=\{MSE_{\alpha,pq}|\alpha=0.25;0.3;...;0.9;0.95\}$.
STOP.

As a result of the matching pursuit based visual dictionary restoration of each pixel block from the image, one gets a set of possible reconstructions, depending on the value α of the α-cut computed for the fuzzy degradation map $\mathbf{M}_D$. The lower the value α, the higher is the confidence that no corrupted pixel was misclassified as un-corrupted (thus, the lower is the false rejection rate of the classifier, if the positive class is considered to be the defect region), as the α-cut of the fuzzy set gives the set of values which belong to the fuzzy set in a membership degree of at least α. However this means that the false acceptance rate of the classifier, that is, the likelihood of misclassifying un-corrupted pixels as corrupted pixels (most often, such pixels being around a boundary in the image or in the details regions) may be large for low values of α. On the other hand, a too large value of α may result in a high false rejection rate, leaving corrupted pixels from the defect areas undetected, thus not removed from the image in the restoration process. To make a decision on the best value of α in the defect detection process or even globally, after image restoration has been performed, on the entire image, is not an optimal solution, since not all the areas in the old films are evenly affected by degradation. Therefore a better solution is to consider computing a set of restored blocks for various α, as presented above, and afterwards perform a locally adaptive selection of the best restoration as the one which maximizes a quality factor that we introduce as follows. We consider the quality of a restored block to be:

- direct proportional to the confidence that we did not consider any corrupted pixels in the reconstruction of the block ;
- direct proportional to the un-corrupted pixel density in the block, as a large number of un-corrupted pixels means that we had sufficient information for reconstructing the block;
- inverse proportional to the mean square error between the restored block and the original block, $MSE_{\alpha,pq}$, estimated on the un-corrupted pixels from the block.

The three terms described above are numerically described as follows:

1. The first term of the quality factor of the block, which describes the confidence of taking into account only un-corrupted pixels when estimating the restored block by the matching pursuit algorithm, denoted herein by Q_1, is proportional to $1-\alpha$, since the lower the value of α, the lower is the risk of misclassifying a corrupted pixel as un-corrupted. In other terms, as any α-cut value tells us that, with a confidence higher than α, all the corrupted pixels are included in the defect mask of the current block, $\{\mathbf{M}_{D\alpha,pq}\}$,it follows that with a confidence of at least $1-\alpha$, all the pixels included in the un-corrupted pixels block mask $\{\mathbf{M}_{C\alpha,pq}\}$ are indeed un-corrupted. However, the defect masks (as the one shown in Fig. 1, where only some samples are depicted) show that the decrease in the number of corrupted detected pixels is non-linear with α, the changes being almost unnoticeable for the values of α close to 0.25 or to 0.95. To account for this situation, we propose to describe the values $1-\alpha$, denoted herein by $\tilde{\alpha}=1-\alpha$, by three fuzzy sets. We will afterwards generate the first term of the quality factor Q_1 as the output of the Takagi-Sugeno fuzzy logic system shown in Fig. 2. The input fuzzy sets of the proposed system are denoted by *Small*, *Medium* and *Large*. The output variable of the fuzzy logic system is described by the singleton output fuzzy sets *Low*, *Avg* and *High*, described by the scalars 0.25, 0.5 and 0.75. In the current implementation, we have chosen the most simple, piecewise linear shape of the three memberships functions (the membership functions of the fuzzy sets *Small* and *Large* are trapezoidal, whereas the membership function of the fuzzy set *Medium* is triangular).

Each rule in the fuzzy logic system rule base associates an input fuzzy set to the corresponding output fuzzy set, that is, *Small* to *Low*, *Medium* to *Avg* and *Large* to *High*, as shown in Fig. 2. The weighted average method is used in the defuzzification step, leading to the following simplified expression of Q_1 in terms of $\tilde{\alpha}$ and of the membership functions of the input fuzzy sets:

$$Q_1=\frac{0.25Small(\tilde{\alpha})+0.5Medium(\tilde{\alpha})+0.75Large(\tilde{\alpha})}{Small(\tilde{\alpha})+Medium(\tilde{\alpha})+Large(\tilde{\alpha})} \quad (9)$$

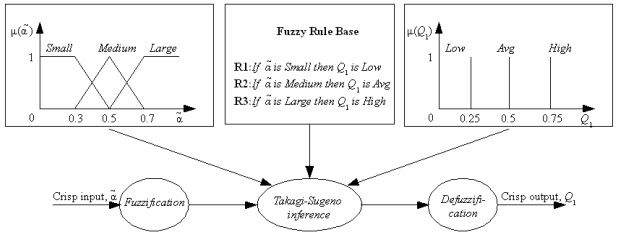


Figure 2. Fuzzy logic system block diagram.

2. The second term of the quality factor of the block simply describes the density of the un-corrupted pixels within the block, since the higher the number of reliable pixels used in the block's reconstruction, the higher the confidence in a correct reconstruction of the block will be. This second term is denoted by Q_2 and it can easily defined for each reconstructed pixels block $\mathbf{I}^{res}_{\alpha,pq}$ from the set I^{res}_{pq} starting from the corresponding un-corrupted binary pixels mask $\mathbf{M}_{C\alpha,pq}$ as the count of the number of ones in the mask (each value 1 indicating an un-corrupted pixel) divided to the block size for normalization:

$$Q_2=\frac{\sum_{i=0}^{P-1}\sum_{j=0}^{P-1}\mathbf{M}_{C\alpha,pq}(i,j)}{P^2}. \quad (10)$$

3. The third term depends on the mean square error between the restored block and the original block, $MSE_{\alpha,pq}$, estimated only on the un-corrupted pixels from the block, as described in the expression (8) above: the smaller $MSE_{\alpha,pq}$, the higher the quality of the current reconstructed block, for the given α, $\mathbf{I}^{res}_{\alpha,pq}$. Since $MSE_{\alpha,pq}$ falls always in the range [0;1], the third term of the quality factor of the current reconstructed block (denoted here by Q_3) may be described by the complement of $MSE_{\alpha,pq}$ in respect to one:

$$Q_3 = 1 - MSE_{\alpha,pq}. \quad (11)$$

We integrate the three quality terms above into a single factor $Q_{pq}(\alpha)$, describing the quality of any given reconstructed block $\mathbf{I}^{res}_{\alpha,pq}$, by a simple arithmetic mean:

$$Q_{pq}(\alpha) = \frac{Q_1 + Q_2 + Q_3}{3}. \quad (12)$$

Finally, in the restored image, each pixels block will be replaced by the particular $\mathbf{I}^{res}_{\alpha^*,pq}$ from the set $I^{res}_{pq} = \{\mathbf{I}^{res}_{\alpha,pq} | \alpha = 0.25;0.3;...;0.9;0.95\}$ maximizing $Q_{pq}(\alpha)$, $\alpha^* = \arg\max_{\alpha} Q_{pq}(\alpha)$. Some examples of operation of the proposed algorithm and an illustration of the effect of the pixels block selection are given in the next section.

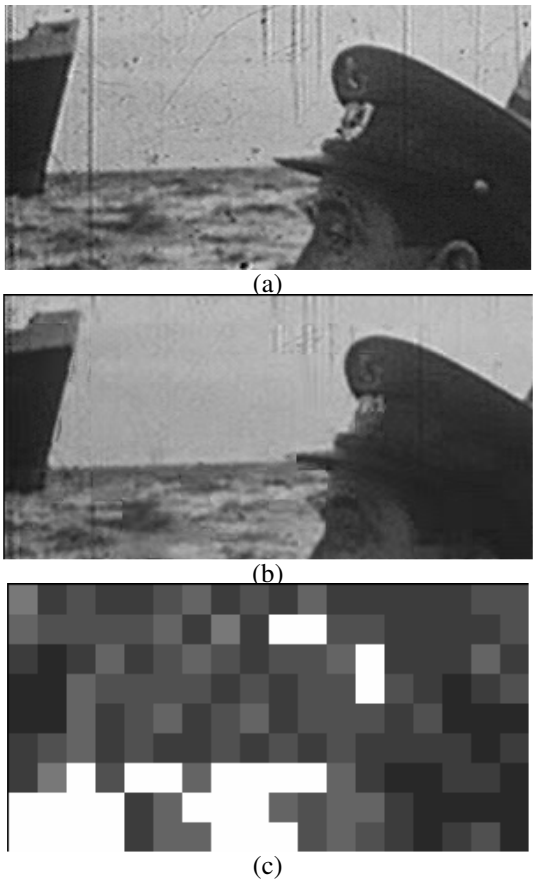
IV. IMPLEMENTATION AND RESULTS

The proposed algorithm for the detection and correction of the defects in digitized old movie frames was implemented using a combination of the Java open-source data mining tool Weka [5] and an own developed C++ software application. The powerful soft classification tools provided by Weka allowed us to select the optimal classifier configuration to distinguish between the corrupted and uncorrupted pixels in the input image (that is, the current grey scale movie frame to be restored) in the feature space described in Section II. The features used for pixel classification were extracted in the C++ software application that we developed. For the classification we used the Logistic Model Tree (LMT) algorithm, as its performance in the correct separation of the pixels into corrupted and uncorrupted, assessed on a validation set, were the best among a series of supervised classifiers implemented in Weka.

Overall, the small size defect detection process described in Section II works as follows. The LMT classifier is trained with a training set generated based on the manual labeling of a movie frame in which different types of small film defects (scratches, spots) and different types of useful details are present. Afterwards, in our software application, the trained classifier is loaded. For each movie frame to be restored, this movie frame is loaded, the features described in Section II for the description of the pixels are extracted, and each pixel is classified with the already trained soft LMT classifier. The resulting values of the decision function of the LMT classifier are further regarded as a fuzzy set $\mathbf{M}_D[H \times W]$ of confidences in the membership of each image pixel to a defect region, and the α-cuts (for α=0.25....0.95 with a step of 0.05, yielding a total of 15) of this fuzzy set are computed and stored as binary maps $\{\mathbf{M}_{D\alpha} | \alpha = 0.25;0.3;...;0.95\}$.

The second part of the algorithm is the reconstruction of the image using the adaptive matching pursuit-based algorithm described in Section III above, relying on the complements of the binary defect maps (which yield the uncorrupted pixels maps) $\mathbf{M}_{C\alpha,pq} = 1 - \mathbf{M}_{D\alpha,pq}$. In the current implementation, we generate a 2-fold overcomplete visual dictionary based on the discrete cosine transform, and the size of each pixels block in the visual dictionary (therefore also in the matching pursuit algorithm) is 16×16 pixels, P=16. The matching pursuit based image restoration is applied for each possible α value, from 0.25 to 0.95, yielding a set of 15 possible reconstructions of each square 16×16 pixels block from the movie frame, out of which, in the end of the block's reconstruction process, the reconstruction that maximizes a quality factor Q (computed according to the expressions (9)-(12) above) is selected as the best choice. This final selected block of pixels will replace the original block in the frame.

The results of the reconstruction are exemplified in Fig. 3 and Fig. 4 for two frames affected by small size defects on the film (scratches and spots), from two different movies. In Fig. 3, we also present the binary defect maps for a series of α values and the outcome of the reconstruction of the frame in case no selection of the optimal reconstructed pixels block (based on Q) would be performed, but a single defect map (a single α) would be used. One can see that in this case, the result would be under-optimal, as compared to the adaptive selection proposed shown in Fig. 3.b). Also, for the movie frame in Fig. 3 we present graphically, in the form of an image, the map of the optimal values α* mapped to grey levels (0.25 represented in black, 0.95 in white, the other values mapped to intermediate grey levels) for each frame. Their variation from one pixel block to another shows that no single optimal α-cut value or single optimal pattern of α values exists in practice, therefore the adaptation scheme proposed is needed for the best quality of the restoration.



(a) (b) (c)

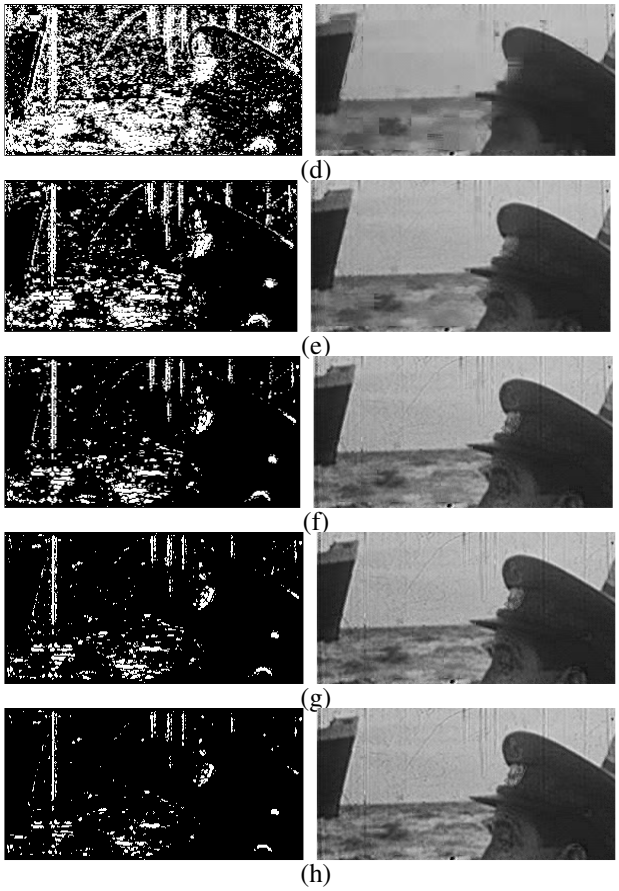


Figure 3. (a)The original image, a digitized old movie frame; (b) The resulting image using the proposed algorithm; (c) The map of the optimal values α; (d-h)The defect pixel map and the reconstructed image for this defect pixel map with the following α values: 0.3 (d), 0.5 (e), 0.7 (f), 0.8 (g), 0.9 (h)*

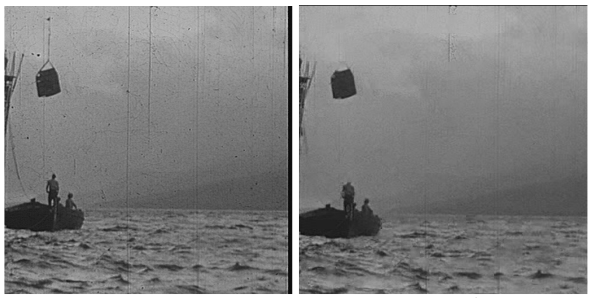


Figure 4. Another movie frame restoration example: (a)The original image; (b) The resulting image using the proposed algorithm

V. CONCLUSIONS

We proposed a new adaptive method of detection and correction of small defects from old movies frames, using supervised soft classifiers and a fuzzy logic adaptive formulation of visual dictionary-based image restoration. The experimental results show a better quality of the restoration as compared to the non-adaptive version of this algorithm, at the cost of an increased computational complexity. In our future work, we will investigate other potential applications of this approach, e.g. by the use of learnt visual dictionaries.

ACNOWLEDGEMENT

This work was supported by a grant of the Romanian MEN–UEFISCDI, project number PN-II-PT-PCCA -2013-4-1791, contract number 296/2014.

REFERENCES

[1] 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

[2] 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.

[3] 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

[4] 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.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update", SIGKDD Explorations, Volume 11, Issue 1, 2009

[6] O.G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising" - Part I: Theory, IEEE Transactions On Image Processing, March 2006, vol. 15, pp. 539–554.

[7] O.G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising" - Part II: Adaptive algorithms, IEEE Transactions On Image Processing, March 2006, vol. 15, pp. 555–571.

[8] M. Aharon, M. Elad, A. M. Bruckstein, The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations, IEEE Transactions on Signal Processing, 2006, 54 (11), pp: 4311–4322.

[9] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Transactions on Image Processing, 2008, 17 (1), pp: 53—69.

[10] G. M. Farinella, S. Battiato, "On the Application of Structured Sparse Model Selection to JPEG Compressed Images", In Proceedings of IAPR 3rd Computational Color Imaging Workshop (CCIW), Milan, Italy, April 2011, Lecture Notes in Computer Science, 2011, Vol. 6626, pp.137-151.

[11] G. Sapiro, "Learning Dictionaries and Sparse Image Representation" SIAM News, 43 (7), September 2010.

[12] S. G. Mallat, Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries", IEEE Transactions on Signal Processing, December 1993, pp. 3397–3415.

[13] G. Rath, C. Guillemot, "A complementary matching pursuit algorithm for sparse approximation", in Proc. European Signal Processing Conference (EUSIPCO), Lausanne, Switzerland, Aug. 2008.