RESTORATION OF DIGITIZED DAMAGED PHOTOS USING BIT-PLANE SLICING

E. Ardizzone, H. Dindo, G. Mazzola

Dipartimento di Ingegneria Informatica (DINFO) dell'Università degli Studi di Palermo.
Viale delle Scienze building 6, 90128, Palermo, Italy
phone: +390917028521, fax: + 390916598043,
email: ardizzon@unipa.it, dindo@csai.unipa.it, mazzola@csai.unipa.it

ABSTRACT

Digital image restoration aims to recover damaged zones of a digital image, using surrounding information. In this paper we propose a novel approach, based on bit-plane slicing decomposition, with the purpose to make information analysis and reconstruction process easy, fast and effective. Tests have been made on digitized damaged old photos to restore several classes of typical defects in old photographic prints.

Index Terms**—**image restoration, digital inpainting, bit-plane slicing

1. INTRODUCTION AND RELATED WORKS

Inpanting deals with filling-in gaps of an image in such a way that for a non-expert observer the restored zone is indistinguishable from the rest of the image. The term "inpainting" comes from the art-craftsmen restoration technique of filling holes and cracks in a painting. The need to restore images is naturally extended from paintings to old photos and films, and so to their digitized versions.

Digital restoration is usually made by professional artists using commercial image processing tools. Many efforts have been done by researchers to replicate their performances with semiautomatic digital techniques. There are two different main approaches for a restoration problem in literature: PDE (Partial Differential Equation) or variational methods, and constrained texture synthesis.

Bertalmio et al.[2] pioneered a restoration algorithm based on a 3rd order PDE model. It was the first time the term "inpainting" was used for a digital image processing application. An earlier 2nd order PDE inpainting model was proposed by Masnou and Morel[8] for a disocclusion problem in computer vision. Olivera et al.[9] proposed a faster inpainting method. Missing region is repeatedly filtered with a 3x3 convolution mask to diffuse known information to the unknown pixels. Chan and Shen[3] proposed a Curvature-Driven Diffusion model based on Euler-Lagrange equation. Texture synthesis methods, in contrast, reconstruct an image from a sample texture. For inpainting purposes, region to fill-in is the area into which synthesize the texture, and texture information to replicate is taken from the surrounding pixels. State-of-the-art works about texture synthesis are: Portilla and Simoncelli[10], Efros and Freeman[6], Wei and Levoy[11], Kokaram[7]. Criminisi et al.[4] proposed an hybrid "exemplar-based" method for removing large objects from digital images.

We propose a novel approach to analyze image features, to store uncorrupted information and to reconstruct damaged zones of a digital image. We applied our techniques to digitally scanned old photos with damages.

2. OLD PHOTOS DEFECTS

Old photographic prints may present several types of defects, due to several factors. In most cases, damages are originated by an inaccurate handling and/or store of the original image, or chemical factors[1].

While the knowledge of damages origin on their physical support is an important issue for a manual restoration activity, several defects appear similar on digitized images, and could be processed with similar methods.

We are interested in processing those defects which can be seen as sets of corrupted pixels surrounded by good information. Spots, cracks, foxing are some of the defects with these features (see fig.1).

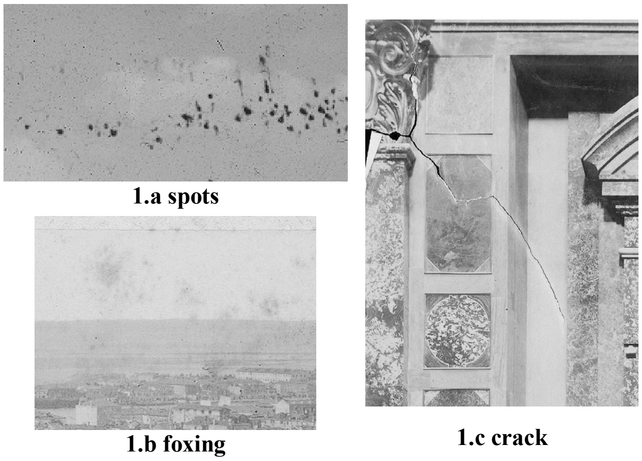
3. PROPOSED ALGORITHM

The main idea is to use a simple observation domain, bit-plane slices, in order to make information analysis and restoration process of damaged areas easy,fast and effective.

Our approach doesn't focus on automatic damage detection. We observed that the defects discussed above appear much darker or much lighter than surrounding pixels. A simple histogram threshold based method is used to detect damaged pixels in the cropped image. However ,we need an input matrix, with the same image size, in which all the pixels are labelled as good or damaged.

Starting from this classification, our method can be divided into three sub-phases:

- Bit-Plane Decomposition and Gray coding
- Information storage
- Restoration



1.a spots

1.b foxing

1.c crack

Figure 1 – Some typical defects in old photos (courtesy of Alinari Archive in Florence)

3.1. Bit-Plane Decomposition and Gray coding

Image is split with a bit-plane slicing decomposition, and each plane is Gray-coded in order to decorrelate information between different planes.

To store information about pixels is an hard task, both for memory space and access time problems. Working with bit sequences, rather than pixel sequences, helps to save memory space and to speed-up access time, making information memorization and recovering faster and easier. Obviously, reconstruction of each plane cannot be independent from others, since annoying artefacts would be visible into reconstructed image after reassembling phase. Gray coding helps to decorrelate planes (see fig 2). However, a more advanced solution, outlined in the next section, has been found. Note that since most part of information is stored in the most significative planes, lower planes could be processed with less care (i.e. smaller window size), speeding-up the process without losing quality in the reconstructed image. We observed that in several cases, damage can be seen only in the most significative planes (fig. 2). So avoiding to process lower bit planes doesn't affect quality of the reconstructed image.

3.2. Information Storage

Once image is decomposed and coded, our method builds a dictionary to store information about the undamaged area.

According to the 2D Markov Random Field (MRF) theory applied to images[5], each pixel of an image is statistically determined by its neighbourhood. Respect to this, the dictionary is built with pixel statistics information. For each undamaged point b(x,y)̀ in each bit plane we consider a square window W_N (N is the user defined window size) centered on it. Then an index is created with the scan-ordered bit sequence inside the window. Similarly another index is built for the previous significative bit plane, with an M-size square window, and added as header to the above index:

$$k(x,y)=\left[\sum_{(x_j,y_j)\in W_M^{\;i+1}} b^{i+1}(x_j,y_j)\cdot 2^j\right]\cdot 2^{N\cdot N}+\sum_{(x_l,y_l)\in W_N^{\;i}} b^i(x_l,y_l)\cdot 2^l \qquad (1)$$

The dictionary stores the statistics of these sequences, that is the *a posteriori* probability of the bit sequences in the i-plane, conditioned by the corresponding sequence in the previous (i+1)-plane

$$H(i,k)=P\left(W_N^{\;i}\mid W_M^{\;i+1}\right) \qquad (2)$$

3.3. Restoration

It is important to note that the order in which pixels are reconstructed strongly affects results. With a simple scan order, the restoration process tend to reproduce up-to-down left-to-right diagonal shapes. Our method computes the average image gradient vector which suggests us the direction along which pixels have to be processed (left-right, by rows or by columns). This solution helps us to reconstruct the natural bias of the image.

The restoration phase is the dual process of the dictionary building process. For each damaged bit in each bit-plane a N-size square window is considered, filled with uncorrupted, corrected and damaged bits. The corresponding M-size square window is considered in the previous plane, in which the whole information is known (bits are either damaged or corrected).

The first goal of the reconstruction process is to compute the probability that the central bit of the window is 1 or 0, given the known neighbor bits in the plane and the bits in the previous plane. The statistics of each of the submasks of a window can be computed building up the statistics of all the possible windows which share the same submask

$$P\left(\hat{W}_N^{\;i}\mid W_M^{\;i+1},b_c=0\right)=\sum P\left(W_j^{\;i}\mid W_M^{\;i+1},b_c=0\right) \qquad (3)$$

$$P\left(\hat{W}_N^{\;i}\mid W_M^{\;i+1},b_c=1\right)=\sum P\left(W_j^{\;i}\mid W_M^{\;i+1},b_c=1\right)$$
$$W_{j_1}^{\;i}\cap W_{j_2}^{\;i}=\hat{W}_N^{\;i} \qquad (4)$$

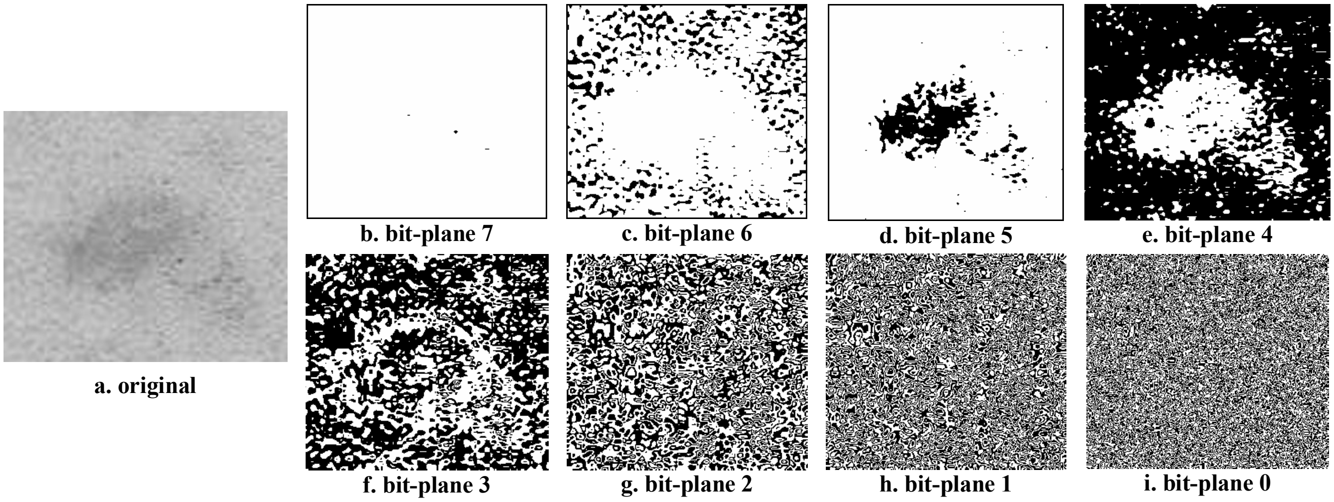
The two statistics we're looking for:

$$S_0^i(x,y)=\sum_{p=0}^{2^{N_D}-1} H\left[i,B^i{}_p(x,y)\right]=P\left(\hat{W}_N^{\;i}\mid W_M^{\;i+1},b_c=0\right) \qquad (5)$$

$$S_1^i(x,y)=\sum_{p=0}^{2^{N_D}-1} H\left[i,W^i{}_p(x,y)\right]=P\left(\hat{W}_N^{\;i}\mid W_M^{\;i+1},b_c=1\right) \qquad (6)$$

where H[i,k] is the dictionary built in the analysis phase, N_D is the number of the damaged bits in the window, B^i_p is the index for the sequence with a "black"(zero) central bit in the window, and W^i_p for the sequence with a "white"(one) central bit. Both of these indexes contain the bits from the $\hat{W}_N$ submask.

The algorithm proceeds by comparing a random generated number with the two statistics, weighted by an user defined parameter α, in order to choice which information (0/1) to put in the central position of the window.



a. original

b. bit-plane 7

c. bit-plane 6

d. bit-plane 5

e. bit-plane 4

f. bit-plane 3

g. bit-plane 2

h. bit-plane 1

i. bit-plane 0

Figure 2 – Damaged image (spot) and its bit-planes decomposition (black means bit 0, white 1)

$$r^i(x,y) = rand\left[0, S_0^i(x,y) + S_1^i(x,y)\right] \qquad (7)$$

$$S_1^i(x,y) > S_0^i(x,y) \rightarrow r^i(x,y) < \alpha \cdot S_1^i(x,y) \quad \begin{cases} b_c = 1 \\ b_c = 0 \end{cases} \qquad (8)$$

$$S_0^i(x,y) > S_1^i(x,y) \rightarrow r^i(x,y) < \alpha \cdot S_0^i(x,y) \quad \begin{cases} b_c = 0 \\ b_c = 1 \end{cases} \qquad (9)$$

By setting α close to 1, bit value is chosen as the result of a random process with the two probabilities:

$$P_0 = \frac{S_0^i(x,y)}{S_1^i(x,y) + S_0^i(x,y)} \quad P_1 = \frac{S_1^i(x,y)}{S_1^i(x,y) + S_0^i(x,y)} \qquad (10)$$

which is what we need to reproduce a highly stochastic pattern. When α>>1 bit value is simply set as the most frequent bit in the window central position with that surrounding conditions. This helps reconstructing zones with strong-oriented lines. Observe that not all the bit-planes have to be processed in the same way, because for less significative planes randomness is more evident.

To avoid the “growing garbage” problem, if no statistics match the actual sequence in the dictionary, the random generation process works using the following probabilities, without increasing execution time

$$\begin{array}{l} P_0 = P\left(b^i(x,y) = 0 \mid b^{i+1}(x,y)\right) \\ P_1 = P\left(b^i(x,y) = 1 \mid b^{i+1}(x,y)\right) \end{array} \qquad (11)$$

After all planes are restored, bit planes are merged to reconstruct the whole image.

3. COMPUTATIONAL COST

Computational cost depends on damaged area size and on the size of the windows used in the analysis and reconstruction phases:

$$\begin{array}{l} O(n-d) + O\left(d \cdot 2^{S-T}\right) \\ S = N \times N + M \times M \end{array} \qquad (12)$$

where d is damaged pixels number, n is image size, T is the table index size.

The first term of eq. 12 is due to the dictionary building phase. It also depends on windows size. The second term is the computational cost of the reconstruction phase. If d<<n and windows are small, first term is predominant and computational cost is O(n). Increasing M, N and d, computational cost becomes exponential in the worst case, that is much far from the real execution time measured with our experiments.

4. EXPERIMENTAL RESULTS

We tested our algorithm on the photos from the Alinari Archives in Florence (about 100 images since 1840) composed of high resolution, color and b/w digital images, concerning the Mediterranean coast, digitized from original pictures, negatives, and slides.

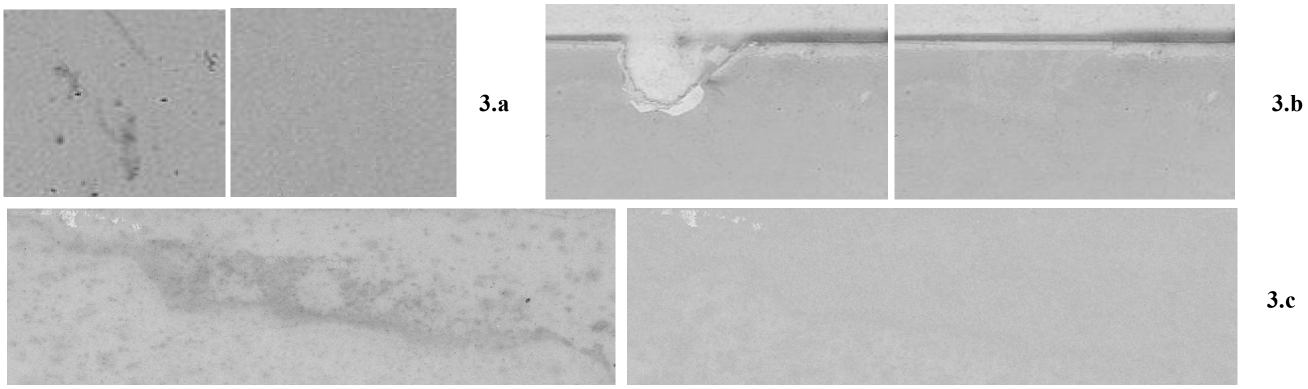
The algorithm has been implemented in ANSI-C, and executed on an AMD Athlon 2 GHz PC with 1 GB RAM.

For small sized images, execution time is much lower than 1 s. Experimental results show that to reconstruct large defects our algorithm takes no more than 10 s.

Figure 3 shows some results of our algorithm on a set of damaged images. We provide significant statistical parameters in order to compare image characteristics before and after restoration. Statistical features for restored images are very similar to those from undamaged pixels. In addition, our method does not introduce any blurring, and even the finest granularity is reconstructed.

5. REMARKS AND FUTURE WORKS

The key-point of our method is to create a consistent statistics dictionary with uncorrupted information. It supposes that the number of damaged pixels is much lower than undamaged, so that good matching will be found between the sequences from the region to restore and those in the dictionary. For the same reason, windows size is typically set to 3x3 or 5x5. Setting higher window



STATS	spots			lacuna			foxing			
	good pics	damaged	restored	good pics	damaged	restored	good pics	damaged	restored	restored(5 p)
mean	165,2690	163,4415	165,1830	184,7088	186,3909	183,9592	189,9209	184,9979	189.9977	189,9877
std dev	3,9103	7,9855	4,1024	13,4507	14,9383	13,5544	7,1406	11,3380	7,2012	7,3315
skew	0,1154	-3,8227	-0,066	-0,8785	-0,4826	-0,9230	0,27544	-0,8026	0,2350	0,2357
kurtosis	3,3652	34,2812	4,1986	8,4864	6,2318	8,0128	3,6515	5,4040	3,8162	3,7046
size	112x94			284x162			1663x482			
def %	16,8			16			32,3			
dict build time	0,03 sec			0,16 sec			2,2 sec			1.9 sec
restoration time	0,06 sec			0,29 sec			7,6 sec			4.8 sec

Figure 3 – Experimental results for typical defects in old photos (3.a spots 3.b lacuna 3.c foxing). Statistical parameters are listed for: uncorrupted pixels in the original image, the whole original image, the restored image. Two restored foxing images: 8 planes and most significative 5 planes processed. No sensible differences in statistical parameters, less execution time

size would lead to a sparse histogram of sequences statistics. It supposes also that uncorrupted pixels should exhibit some regularity. We are currently working on the improved version of the algorithm, able to process highly textured images. Furthermore, we are extending the presented approach to several other types of defects commonly found in images. However, an automatic damage detection method still remains an open question.

6. CONCLUSIONS

In this paper we presented a new simple, fast and effective method to restore defects in digitized old photos.

Bit-plane slicing decomposition is used to observe image features in a simple domain, to analyse information, and to reconstruct missing pixels to restore corrupted zones, respecting boundary conditions.

Testing has been made on a set of defects that may be typically seen on digitized old photographic prints.

ACKNOWLEDGEMENTS

This work has been partially funded by the MIUR (Italian Ministry of Education, University and Research) project FIRB 2003 D.D. 2186 - Ric December 12th 2003. The authors wish to thanks the Alinari Photo Archives in Florence for having permitted the use of their photos in this research.

REFERENCES

[1] E. Ardizzone, H. Dindo, U. Maniscalco, G. Mazzola, "Damages of Digitized Historical Images as Objects for Content Based Applications," EUSIPCO 2006, Florence, Italy, September 2006

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in Proceedings SIGGRAPH, 2000.

[3] T. F. Chan and J. Shen. "Mathematical models for local non-texture inpaintings," SIAM Journal of Applied Math., 62(3):1019–1043, 2002

[4] A. Criminisi, P. Perez, K. Toyama. "Region Filling and Object Removal by Exemplar-Based Inpainting" 2004 IEEE Transactions on Image Processing 13 (9) pp. 1200-1212,

[5] A. C. Cross, and A. K. Jain, "Markov Random Field Texture Models," IEEE Transactions on Pattern Analysis and Machine Intelligence 5 (1983), 25–39.

[6] A. A. Efros and W- T. Freeman, "Image quilting for texture synthesis and transfer," in Proceedings SIGGRAPH, 2001, pp. 341–346.

[7] A. Kokaram , "Parametric texture synthesis for filling holes in pictures," in Proceedings of International Conference on Image Processing 2002 Volume 1, 22-25 Sept. 2002 Page(s):I-325 - I-328 vol.1

[8] S. Masnou, J.M. Morel, "Level Lines based Disocclusion," Proceedings of the 1998 IEEE International Conference on Image Processing, pp 259 - 263, 1998.

[9] M. Oliveira, B. Bowen, R. McKenna, and Y. -S. Chang, "Fast Digital Image Inpainting," In Proc. VIIP 2001, pp. 261—266, [CITY]: [PUB], 2001.

[10] J. Portilla and E. P. Simoncelli, "A Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients," International Journal of Computer Vision, 40(1):49–71, December 2000.

[11] L. Wei, M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", SIGGRAPH 2000 pp 479-488