AN INPAINTING SYSTEM FOR AUTOMATIC IMAGE STRUCTURE - TEXTURE RESTORATION WITH TEXT REMOVAL

Eftychios A. Pnevmatikakis[†] and Petros Maragos[‡]

[†] Columbia University, Dept. of E.E., New York, NY 10027, USA. eap2111@columbia.edu
[‡] National Technical Univ. of Athens, School of E.C.E., Athens 15773, Greece. maragos@cs.ntua.gr

ABSTRACT

In this paper we deal with the inpainting problem and with the problem of finding text in images. We first review many of the methods used for structure and texture inpaintings. The novel contribution of the paper is the combination of the inpainting techniques with the techniques of finding text in images and a simple morphological algorithm that links them. This combination results in an automatic system for text removal and image restoration that requires no user interface at all. Examples on real images show very good performance of the proposed system and the importance of the new linking algorithm.

Index Terms— inpainting, texture synthesis, text detection, mathematical morphology.

1. INTRODUCTION

Inpainting is the process of filling-in missing or damaged image information. Its applications include the removal of scratches in a photograph [1], repairing damaged areas in ancient paintings, recovering lost blocks caused by wireless image transmission, image zooming and super-resolution [2], removing undesired objects from an image [3], or even perceptual filtering [4]. In the inpainting problem, the user has to select the area to be filled in, the inpainting region, since the areas missing or damaged in an image cannot be easily classified in an objective way. However, there are some occasions where this can be done. One such example, are the text characters printed in an image. Detecting and recognizing these characters can be very important in bi-modal search of internet data (image and text), and removing these is important in the context of removing indirect advertisements, and for aesthetic reasons.

In this paper we deal with the problem of *automatic* inpainting-based image restoration after text detection and removal from images. That is, given an image with text characters how to detect the exact position of the characters in the text, remove them and fill-in the resulting gaps via inpainting methods. Our contribution is a new system for automatic text detection and removal from images, that requires no user interaction. This system is important because the selection of the area to be inpainted has been done manually by previous inpainting systems.

2. BACKGROUND AND REVIEW OF INPAINTING

For the inpainting problem it is essential to proceed to the discrimination between the structure and the texture of an image. As *structure* we can define the main parts - objects of an image, whose surface is homogeneous without having any details. As *texture* we can define the details on the surface of the objects which make the images more realistic.

2.1. Structure Inpainting

The term inpainting was first used by Bertalmio et al. in [1]. The idea is to propagate the isophotes (lines with the same intensity) that arrive at the boundary of the inpainting region, smoothly inside the region while preserving the arrival angle. In the same context of mimicking a natural process, Bertalmio et al. suggested another similar model, where the evolution of the isophotes is based on the Navier Stokes equations that govern the evolution of fluid dynamics [5].

Apart from physical processes, images can also be modeled as elements of certain function spaces. An early related work under the word "disocclusion" rather than inpainting was done by Masnou and Morel [6]. In [2] Chan and Shen derived an inpainting model by considering the image as an element of the space of Bounded Variation (BV) images, endowed with the Total Variation (TV) norm. The solution of the inpainting problem comes from the minimization of an appropriate functional. This TV inpainting model was extended in [7, 8] to take into consideration curvature and the the so called connectivity principle according to which the human eye tends to reconstruct the broken edges. Along similar lines, Mumford and Shah [9] proposed another inpainting model, which takes explicit care of the edges on the functional to be minimized. Its extension to account for curvature was proposed in [10] using the Euler's elastica .

2.2. Texture Inpainting

The problem of texture inpainting is highly connected with the problem of texture synthesis. A very simple and highly effective algorithm was presented by Efros and Leung [11]. In this algorithm the image is modeled as a Markov Random Field and texture is synthesized in a pixel by pixel way, by picking existing pixels with similar neighborhoods in a randomized fashion. This algorithm performs very well but it is very slow since the filling-in is being done pixel by pixel. In [12] the speed was greatly improved by using a similar and simpler algorithm, which filled in the image in a block by block of pixels way.

An algorithm specific for texture inpainting was presented by Criminisi et al. in [3]. The algorithm uses the same texture synthesis techniques as the Efros-Leung algorithm. The only difference is that the pixels that are placed along the edges of the image are filled in with high priority. This slight difference is adequate to give better results for image inpainting as is presented in the results in [3].

2.3. Simultaneous Structure and Texture Inpainting

Since, in most images both structure and texture are present, a natural thing is to combine structure and texture inpainting techniques to obtain more effective techniques, regarding image inpainting. This was done in [13]. The algorithm is very simple. First the image is decomposed into its structure and texture components, and then inpainting techniques are performed for both structure and for texture. For structure texture decomposition many variational methods can be used, like [14], which was first used for image denoising, or [15] which discriminates between texture and noise.

3. NEW SYSTEM

Our system aims in the automatic detection of text and its removal from images. First we make an initial detection of the text characters in the image. This is done by the algorithm presented in [16]. We briefly repeat it for completeness. The algorithm assumes that the intensity of the text characters is different by their background by at least a certain threshold, for example 30 in an image with 8-bit intensity, i.e., the characters correspond to the strong edges of the image. As a first step the image is being blurred and the morphological gradient is calculated and being passed over a threshold. After that a morphological closing operator is being applied and the connected components that represent regions with text are derived. The character positions are then derived by a second thresholding of each connected component.

The algorithm is very simple and gives satisfactory results, when used for OCR purposes. When used for the determination of the inpainting region we need to be careful. The reason is that when we need to extract text for recognition, we only need to extract the basic structure of each character, so

that this character can be recognized. On the other hand, when we need to extract the text and then remove it, then we have to make sure, that all the pixels that belong to the characters are selected. To ensure this we introduce a new technique. What we need is to dilate the text region that was derived from the algorithm in order to cover all the text character pixels and only those. This goal leads us to two morphological operators: Conditional dilation and reconstruction opening.

Conditional dilation is defined as:

$$\delta_B(M|X) \triangleq (M \oplus B) \cap X, \quad (1)$$

where X is the reference frame inside which the dilations are allowed, and M plays the role of a marker to be expanded inside the frame. The reconstruction opening is the limit of iterative conditional dilation with the same reference frame and it is defined as the operator

$$\rho_B^-(M|X) \triangleq \lim_{n \to \infty} \delta_B^n(M|X) = \cup_{n=1}^\infty \delta_B^n(M|X). \quad (2)$$

Here we apply iterative conditional dilation but with an adaptive reference frame. To calculate the reference frame at each iteration of the algorithm, we use the hypothesis that the intensity of the pixels in the characters is different is different by at least a certain threshold γ than their background and the hypothesis that the all the pixels that correspond to characters have similar intensity.

The algorithm is as follows: Let M the binary picture that represents the initial guess of the text position, and let u be the original image.

1. Initialization: $M_1 = M$

2. Calculation of the reference frame X_n:

   - Calculation of the mean value of the original image in the mask area: $s_n = \frac{\sum(u \cdot M_n)}{\sum M_n}$.
   - Dilation with a unitary disk B, of the image M_n. $D_n = M_n \oplus B$.
   - $X_n = \{[i,j] \in D_n : |u[i,j] - s_n| < \gamma\}$, where γ is the threshold mentioned above (eg. γ=30).

3. Conditional dilation for the new prediction:

$$M_{n+1} = \delta_B(M_n|X_n).$$

4. Repeat steps 2-3 until $M_{n+1} = M_n$.

We have experimentally found that a good empirical value for the threshold is $\gamma = 30$ and that the algorithm converges quickly, e.g. in less than 10 repetitions. After the termination of this algorithm we have the exact positions of the text characters in the image. We can also apply an unconditional morphological dilation operation with a unitary disk, to ensure that we have captured all the pixels that correspond to text characters.
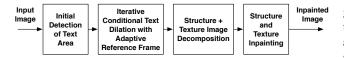
Fig. 1. New system for automatic text removal and image inpainting.

As a final step of our system, we now use these positions derived above as the inpainting area and apply some of the techniques (structure and/or texture inpainting) mentioned in section 2 to remove them and reconstruct the image. At the most general setup simultaneous structure and texture inpainting is performed. Using the method presented in [14] if f is the initial noiseless image then the structure component u is given from solving the minimization problem:

$$\inf_{u \in L^2} F(u) = \int_{\Omega} |\nabla u| dx dy + \lambda \int_{\Omega} |f - u|^2 dx dy, \quad (3)$$

where ∇ denotes the gradient and λ is a regularization constant. The texture component v is given by $v = f - u$. For structure inpainting we use the algorithm of [1]. The propagation of the isophotes is achieved by the numerical solution (inside the inpainting region) of the PDE:

$$\frac{\partial u}{\partial t} = \nabla(\triangle u) \cdot \nabla^{\perp} u, \quad (4)$$

where $\triangle$ and $\nabla^{\perp}$, denote the Laplacian, and the orthogonal gradient (isophote-direction), respectively. For texture inpainting the Efros-Leung algorithm is used [11]. A schematic representation of our system is given in Fig. 1.
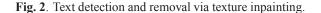
4. EXPERIMENTAL RESULTS

We present results of the system that we described above. It is important to note that, the system requires no user interaction, since both the text detection (inpainting region) and the inpainting process are being done automatically.

In Fig. 2 we used texture inpainting and the algorithm from [11] since the area around the text characters is an area with texture and no structure at all. In Fig. 3 the image contains both structure and texture; thus, both the decomposition of the image in the two components and the separate processing of each one are needed. In general, an automatic system is needed to decide about the appropriate inpainting method.

We also present the inpainting results when the inpainting area was the initial text guess as taken by the algorithm in [16], to illustrate the efficiency of the morphological algorithm we proposed to improve the text detection. In Fig. 2(b) we can easily recognize the text, but this area does not contain all the character pixels. After applying the simple recursive algorithm we can see that the final guess contains all the pixels, and thus the inpainting algorithms can be applied effectively. When we apply the inpainting over the initial text

guess area, we see that the whole area is not restored because the text has not been captured completely. However the text area that has been captured is restored satisfactory. This happens because we applied texture inpainting that fills-in the image by using information not from the neighbor pixels, but from the whole image.

(a) Original Image

(b) Initial Text guess

(c) Final Text Guess (Inpainting Area)

(d) Reconstructed Image with initial text guess

(e) Reconstructed Image

Fig. 2. Text detection and removal via texture inpainting.

This is not the case in Fig. 3. As we can see, the image inpainting, when the inpainting area is the initial text guess (Fig. 3(h)) gives very poor results. The reason is that in this example we applied simultaneous structure and texture inpainting. Structure inpainting uses the neighboring pixels to generate new information. Since the initial text guess has not captured all the text-pixels, the neighboring pixels are still text-pixels and thus they don't generate the appropriate information. By using the new algorithm, we make accurate text detection, and this is reflected at the final inpainted image (Fig. 3(i)). In the two text detection images of Fig. 3(b),(c), we also observe that at the initial guess we have a small area (at the left of the image) that does not correspond to the text. However, in the final guess this erroneous area has vanished. This is so because the intensity of the pixels in this area is different than the intensity of the pixels of characters, and therefore this area vanished gradually from the reference frame

(a) Original Image   (b) Initial Text guess   (c) Final Text Guess   (d) Structure Image   (e) Texture Image

(f) Structure Inpainting   (g) Texture Inpainting   (h) Reconstructed Image with initial text guess   (i) Reconstructed Image with new algorithm

Fig. 3. Automatic text detection and removal via simultaneous structure and texture inpainting.

during the iterations of the algorithm. This also shows the importance of the adaptive reference frame X_n in the linking algorithm. If reconstruction opening was used with the initial reference frame X_1, then this erroneous area would not have vanished.

5. CONCLUSIONS

In this paper we dealt with the combined problems of inpainting and finding text in images. We have proposed a new simple morphological algorithm inspired from the reconstruction opening operation. This algorithm captures all the pixels that correspond to text characters and thus its output can be considered as the inpainting region. By applying then an appropriate inpainting method, we have developed a system for automatic text removal and image inpainting.

6. REFERENCES

[1] M. Bertalmio, V. Casseles, C. Ballester, and G. Sapiro, *Image Inpainting*, ACM Press, 2000.

[2] T. Chan and J. Shen, "Mathematical models for local nontexture inpaintings," *SIAM J. on Appl. Math.*, vol. 62, pp. 1019–1043, 2002.

[3] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar based inpainting," in *Proc. IEEE-CVPR*, 2003, vol. 2, pp. 721–728.

[4] M. Dimmiccoli and P. Salembier, "Perceptual filtering with connected operators and image inpainting," in *Proc. ISMM*, 2007.

[5] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics and image and video inpainting," *Proc. of IEEE-CVPR*, pp. 355–362, 2001.

[6] S. Masnou and J.M. Morel, "Level lines based disocclusion," in *Proc. IEEE-ICIP*, 1998, pp. 259–263.

[7] T. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[8] T. Chan, S. Kang, and J. Shen, "Euler's elastica and curvature based inpaintings," *SIAM J. on Appl. Math.*, 2002.

[9] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.

[10] S. Esedoglu and J. Shen, "Digital inpainting based on the Mumford-Shah-Euler image model," *European Journal of Applied Mathematics*, vol. 13, no. 4, pp. 353–370, 2002.

[11] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE-ICCV*, Corfu, Greece, 1999, vol. 2, pp. 1033–1038.

[12] A. Efros and W.T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. ACM Conf. Comp. Graphics (SIGGRAPH)*, Eugene Fiume, 2001, pp. 341–346.

[13] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Im. Proc.*, vol. 12, no. 8, pp. 882–889, 2003.

[14] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1-4, pp. 259–268, 1992.

[15] Y. Meyer, "Oscillating pattern in image processing and nonlinear evolution equations," *AMS*, 2001.

[16] Y. Hasan and L. Karam, "Morphological text extraction from images," *IEEE Trans. Im. Proc.*, vol. 9, no. 11, pp. 1978–1983, 2000.