

# Use of two inpainting Techniques to Restore Partially detected Cartographic Features

Ana Luisa Chaves Figueira<sup>1</sup>, Erivaldo Antonio da Silva<sup>2</sup>, Maurício Araújo Dias<sup>1</sup>, Breno Strogueia Maia da Cruz<sup>1</sup>, Guilherme Pina Cardim<sup>2</sup>

<sup>1</sup>Dept. of Mathematics and Computing, UNESP, Presidente Prudente, São Paulo, Brazil

<sup>2</sup>Dept. of Cartography, UNESP, Presidente Prudente, São Paulo, Brazil

**Abstract**— The continuous use of methodologies to extract cartographic features of digital images have been of great importance in the area of cartography. Many techniques can be used by features extraction processes, however, the results obtained by these techniques usually have partially detected features, culminating in loss of quality of the extraction process. To keep searching for better results, it is possible to use techniques based on inpainting, that has as its main purpose image restoration and removal of occlusions. Therefore, the main objective of this article is to show a methodology of reconstruction of partially detected features using two inpainting techniques proposed by [1] and [2], aiming to improve the quality of results in the process of extraction of cartographic features and digital images. Observing the final analysis of the results obtained with the techniques in three entry images, the technique of [1] showed an improvement of 0.61% compared to the extracted feature. While the technique of [2], an improvement of 6.82%. The good results obtained regarding the improvement of the quality of the process of extraction of partially detected cartographic features will be of great use in the area of cartography.

**Keywords**— Remote Sensing, Inpainting, Digital Image Processing, Cartography, Partially Detected Features.

## I. INTRODUCTION

The continuous use of methodologies to extract cartographic features of digital images have been of great importance in the area of cartography. These processes have as focus the identification of existing targets in the terrestrial surface and its changes, which are required to update cartographic products. Many techniques can be used by features extraction processes, however, the results obtained by these techniques usually have partially detected features, culminating in loss of quality of the extraction process. To keep searching for better results, it is possible to use techniques based on inpainting, that has as its main purpose image restoration and removal of occlusions. It operates by gathering information around the damaged area and making a subtle junction of this information with the area of interest. Thus, based on what has been said, the main objective of this article is to show a methodology of reconstruction of partially detected features using two inpainting techniques proposed by [1] and [2], aiming to improve the quality of results in the process of extraction of cartographic features and digital

images. Both techniques are compared, in order to understand which one provided better results.

### A. Bertalmio et al. [1] Inpainting Algorithm

This algorithm is mainly based on nonlinear partial differential equations and the imitation of techniques of artists specialized in restoration of museum paintings.

Being  $A$  the region in which the inpainting process will be carried out and  $bA$  its boundary, the isophote lines that focus on  $bA$  will be prolonged, maintaining its incidence angles. After this procedure, it is defined the contour of the area that will be inpainted. This area will be filled from the extent of the regions around  $A$ . The different regions contained in  $A$ , determined by the contour lines, will be completed by the colours that match the colours of  $bA$ .

The general equation of the algorithm is show in equation (1).

$$I^{(n+1)}(i, j) = I^n(i, j) + \Delta t I_t^n(i, j), \forall (i, j) \in A \quad (1)$$

where  $n$  is the inpainting time,  $i$  and  $j$  are the pixel coordinates,  $\Delta t$  is the rate of improvement,  $I^n(i, j)$  is the

entry image and  $I_t^n(i, j)$  is the improved version of the entry image.

Equation (1) shows that  $I^{(n+1)}(i, j)$ , which is originated from  $I_t^n(i, j)$ , will be an improved version of the entry image. As  $n$  grows, the algorithm tends to have better results.

To ensure the correct definition of the direction field, the diffusion process is intertwined with the inpainting process described, that is, the next step is the application of few iterations of image diffusion. This diffusion prevents the lines from crossing each other, resulting in a smoothing effect. [1] uses the anisotropic diffusion, determined by the following:

$$\frac{\partial I}{\partial t}(x, y, t) = g_\varepsilon(x, y) \kappa(x, y, y) (\nabla I(x, y, t)), \forall (x, y) \in A^\varepsilon \quad (2)$$

where  $A^\varepsilon$  is the dilation of  $A$  with a ball of radius  $\varepsilon$ ,  $\kappa$  is the Euclidean curvature of the isophotes of  $I$  and  $g_\varepsilon(x, y)$  is the smooth function in  $A^\varepsilon$ .

The only input parameters of the algorithm are the image to be restored and the mask that delimits the portion to be inpainted of the input image. The algorithm performs a pre-processing step where the entire original image goes through the smoothness process of anisotropic diffusion. After that, the image enters an inpainting loop, where only some values within  $A$  are modified. At each iteration, an anisotropic diffusion step is applied. This process is repeated until a stable state is reached.

In the restoration loop  $X$  inpainting steps occurs using equation (1), then  $Y$  diffusion steps with equation (2), and again  $X$  steps of equation (1), and so on. The total number of steps is  $T$ . This number may be pre-determined or the algorithm may stop when image changes are below the given limit. The value of  $T$  depends on the size of  $A$ .

### B. Deng et al. [2] Inpainting Algorithm

The algorithm proposed by Liang-Jian Deng Ting-Zhu Huang, Xi-Zhao is not based on partial differential equations (EDPs). It fills regions of interest by copying and pasting the portions of the source regions, so that the texture of the image remains the same. The type of technique exploited by this algorithm is called exemplar-based.

Originally, exemplar-based algorithms are based on two attributes: a confidence term and a data term. The data term propagates the target region geometrically, and the term of confidence describes the dependence of the area of the patch to be copied and pasted in relation to the neighbouring pixels of the source region, that is, the texture propagation of the original image. If there are more pixels

of the source region around a pixel  $p$ , the confidence term of  $p$  will get a higher value.

Equations (3) and (4) define the priority of a patch, so we select the one with the highest priority, and fill the target region with the patch from the source region that is most similar to it.

$$\begin{cases} C(p) = 0, \forall p \in \Omega \\ C(p) = 1, \forall p \in \omega \end{cases} \quad (3)$$

$$D(p) = -0.1, \forall p \in \Omega \cup \omega \quad (4)$$

where  $C(p)$  and  $D(p)$  is the confidence term and data term of a pixel, respectively,  $\Omega$  is the area of interest and  $\omega$  is the region that doesn't belong to the area of interest.

The similarity between two patches is measured by the following equation:

$$\gamma_p = \underset{\gamma_q \in \Theta}{\operatorname{argmin}} d(\gamma_p, \gamma_q) \quad (5)$$

Each pixel  $p$  is filled with the corresponding pixel in  $\gamma_q$ , by using equation (6):

$$p' \in \gamma_p \cap \Omega \quad (6)$$

Then, the confidence term is updated to:

$$C(q) = C(p), \forall q \in \gamma_p \cap \Omega \quad (7)$$

All of these processes are repeated iteratively until the target region is completely filled. What differentiates the technique proposed by [2] from the common exemplar-based algorithms is a new definition of the priority of the patches taken and the similarity equation. The new priority definition is described in equation (8).

$$P(p) = \begin{cases} D(p), & \text{for the first phase} \\ C(p), & \text{for the second phase} \end{cases} \quad (8)$$

The first phase concentrates the geometric propagation of the target region, and the second, the propagation of the texture. The algorithm automatically estimates the number of iterations required for the execution of the first phase.

As for the similarity equation, it was changed to equation (9).

$$\gamma_p = \underset{\gamma_q \in \gamma'_q}{\operatorname{argmin}} d(\gamma_q, \gamma_p) \quad (9)$$

where  $\gamma_p$  and  $\gamma_q$  are patches being compared,  $\gamma'_q$  is the largest patch with its center being  $\gamma_q$ 's center and  $d(\gamma_q, \gamma_p)$  is the sum of the quadratic differences of the pixels that already filled the two patches.

### C. Quantitative Metrics

In [3], [4], [5] the lack of quantitative metrics to evaluate the results of an inpainting process is addressed. The reason why this happens is that there is usually no reference image, and because the content of the area to be

rebuilt is unknown. Therefore, in most cases, a visual evaluation is used, where it is verified if the result is the appropriate one. However, visually analysed results are complex and unpredictable due to human factors that are difficult to control. Thus, an alternative is to use known quality metrics in the area of digital image processing, among them the most used ones are: MSE, PSNR and SSIM.

The MSE is the mean square error of an estimator, its value is always positive and the results close to zero are better.

The PSNR is a term for the relation between the maximum signal value and the maximum noise value that affects the fidelity of a representation. To calculate it, the MSE is needed.

SSIM is an index that predicts the quality of images and videos, when measuring the structural similarity between two images. SSIM was created as an enhancement of MSE and PSNR comparing methods.

The main difference between SSIM and its predecessors is that SSIM is a method based on visual perception [6], [7] and [11] reiterate that SSIM is more efficient when compared to MSE and PSNR methods. This is due to the latter not detecting distortions perceptible by the human visual system. The reason why both work that way is that they only consider the individual state of each pixel and not its structural information, contrary to how SSIM operates.

Also in [7], [8], [9] and [10] it is argued that MSE and PSNR are not suitable for binary images. In this case, the MSE represents the number of differences between two images, and the large number of different pixels does not always result in a large structural difference, because binary images do not have many texture details and their pixel distribution is simpler.

Thus, this work included the manual construction of a reference image, based on the original unprocessed image and applied the SSIM metric, to evaluate the quality of the results obtained. The metrics were applied in the entry images and in the inpainted results. When compared with the SSIM of the entry image, the results obtained after the application of the technique evidences the improvement of the quality of the process.

## II. METHODOLOGY AND RESULTS

### A. Images Used In The Tests

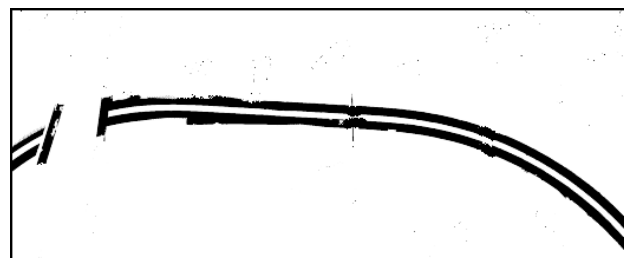
Three images containing partially detected features were used, those are presented in Figures 1, 2 and 3.



1. Test image 1.



2. Test image 2.



3. Test image 3.

### B. Softwares Used

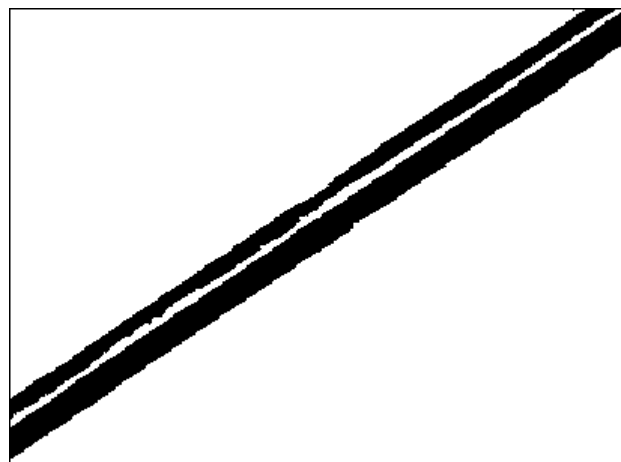
The software used for the implementation was the Matlab R2017a, 64-bit version. The processing tests were made on an Intel Core i7 processor computer with an Nvidia GeForce 940MX 2GB graphics card.

### C. Results

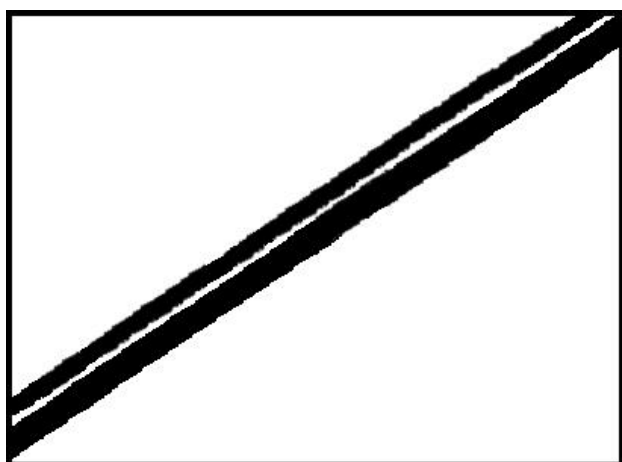
Figure 4 shows the test image 1, figure 5, the reference image, and in figures 6 and 7, the results obtained with the implementation of the inpainting algorithm of Bertalmio et al. [1] and Deng et al.[2] respectively.



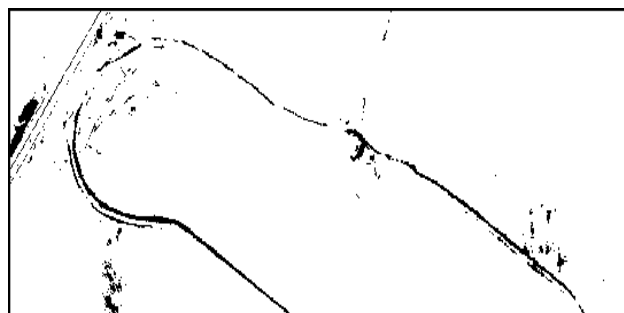
4. Test image 4.



7. Result of the inpainting technique of [2].



5. Reference image.



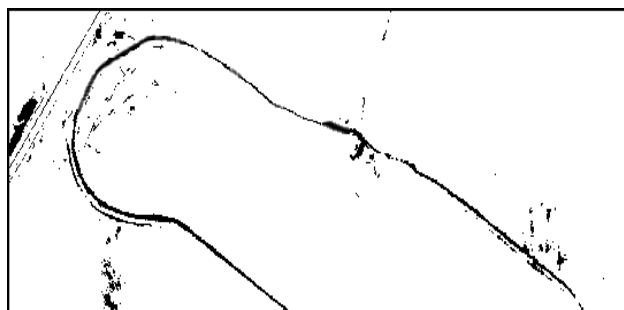
8. Test image 2.



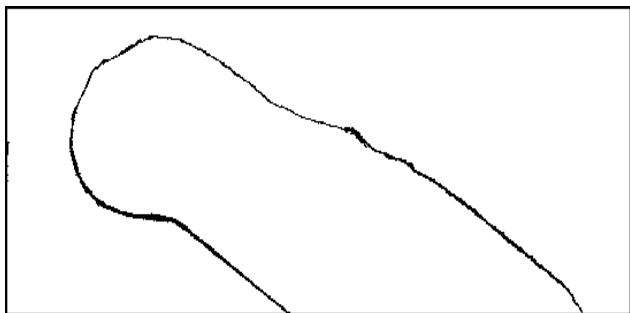
6. Result of the inpainting technique of Bertalmio et al. [1].



9. Reference image.

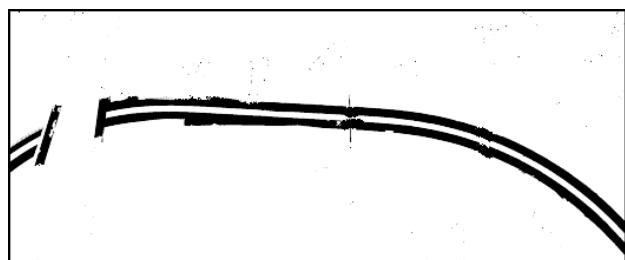


10. Result of the inpainting technique of Bertalmio et al. [1].

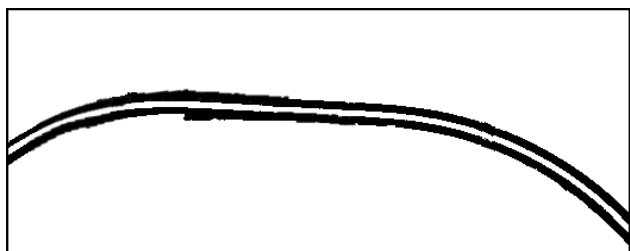


11. Result of the inpainting technique of [2].

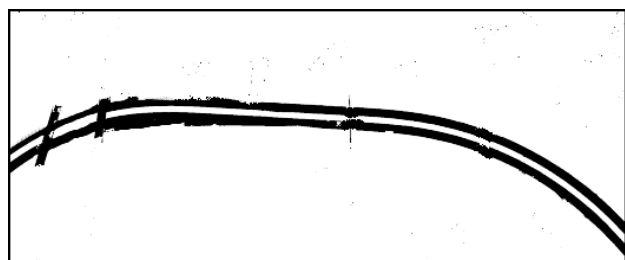
Figure 12 shows the test image 3, figure 13, the reference image, and in figure 14 and 15, the results obtained with the implementation of the inpainting algorithm of Bertalmio et al. [1] and Deng et al.[2] respectively.



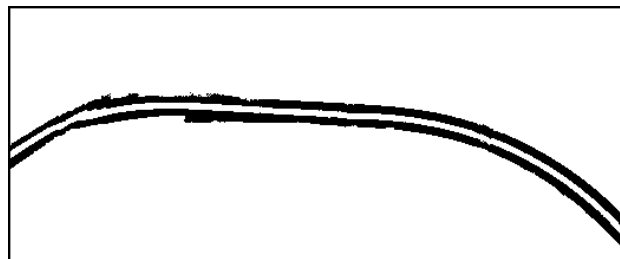
12. Test image 3.



13. Reference image.



14. Result of the inpainting technique of Bertalmio et al. [1].



15. Result of the inpainting technique of [2].

In order to facilitate the visualization of the results obtained with the application of the technique of inpainting, table I presents all the results obtained, and Table II, the processing time of each technique.

Table I. SSIM comparison between original images and results of the inpainting techniques.

Test Image	SSIM 1(%)	SSIM 2 (%)	
		[1]	[2]
1	77.21	78.94	96.11
2	88.77	88.91	98.67
3	93.02	93.75	97.30
Mean	86.33	87.20	97.36

SSIM 1 refers to the comparison between the entry image and the reference image, while SSIM 2, between the reference image and the result images of the corresponding techniques.

Table II. Processing time of each image for both techniques

Test Image	Processing time(s)	Processing time (s)
	Bertalmio, 2000	[2]
1	0.82	3.87
2	1.53	4.58
3	1.22	3.56

### III. CONCLUSIONS

Observing the final analysis of the results obtained with the techniques proposed by [1] and [2] in the entry images, we can understand that the results were highly satisfactory.

The technique of [1] showed an improvement of 0.87% compared to the extracted feature. While the technique of [2] obtained an improvement of 11.03%. This difference may happen due to the first technique not fully allow removal of occlusions and incorrectly detected features. This type of removal is frequent in most study cases, as we

can see in the test images presented, which can be considered a weak point for [1] technique.

However, the processing time for each algorithm differs significantly considering the amount of regions to be inpainted, image's dimensions, and other attributes that may interfere in the restoration process. [1] technique proved to be faster than [2]. Also, [1] algorithm seems to be as useful as [2] to fill features with missing regions.

Therefore, we can say that the inpainting algorithm of [1] is more appropriate to restore images as long as there are few occlusions and incorrectly detected features, otherwise [2] for the removal process.

At any rate, the good results obtained regarding the improvement of the quality of the process of extraction of partially detected cartographic features can be used in the area of cartography, by supporting processes that update cartographic products.

### ACKNOWLEDGEMENTS

The authors thank the Foundation for Research Support of the State of São Paulo - FAPESP (Process 2017/13029-8) and the National Council of Scientific Research - CNPq.

### REFERENCES

- [1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, *Image inpainting*. Proc. Siggraph Computer Graphics, 2000.
- [2] L-J.Deng, T-Z.Huang, X-L.Zhao, *Exemplar-Based Image Inpainting Using a Modified Priority Definition*. PLoS ONE, 2015.
- [3] A. C.Kokaram, *Detection of Missing Data in Image Sequences*. IEEE Transactions on Image Processing, 1995.
- [4] O.Feier, A.Ioana, *Digital Inpainting for Artwork Restoration: Algorithms and Evaluation*, 97f, 2012.
- [5] O. L.Meur, Guillemot C, *Image Inpainting: Overview and Recent Advances*. IEEE Signal Processing Magazine. 2014.
- [6] T. T.Dang, A.Beghdadi, M. C.Larabi, *Inpainted image quality assessment*. European Workshop on Visual Information Processing (EUVIP), 2013.
- [7] N.Peter, K. Hisakazu, Y. Masahiro, W. Hidenori, M. Shogo, *SSIM image quality metric for denoised images*. International Conference on Visualization, Imaging and Simulation – Proceedings, 2010.
- [8] W. Maged, E. Rasha, *Quality Assessment of Visual Cryptography Images*. Australian Journal of Basic and Applied Sciences, 2016.
- [9] B. Furht et al., *Multimedia Encryption and Watermarking*. Springer US, 2005.
- [10] C. Zhang, Z. Qiu, *Simple quality assessment for binary images*. Journal of Electronics (China), 2007.
- [11] W. Zhou, *Objective Image Quality Assessment: Facing The Real-World Challenges*. Electronic Imaging, 2016.