Available online at www.sciencedirect.com
SCIENCE DIRECT®
Image and Vision Computing 25 (2007) 61–69
www.elsevier.com/locate/imavis

Image restoration using digital inpainting and noise removal

Celia A. Zorzo Barcelos [a], Marcos Aurélio Batista [b,*]

[a] Faculty of Mathematics, Federal University of Uberlândia, Caixa Postal 593, CEP 38.400-902, Uberlândia, MG, Brazil.
[b] Department of Computing, Federal University of Goiás, Campus Catalão, Caixa Postal 56, CEP 75704-020, Catalão, GO, Brazil.

Received 26 May 2004; received in revised form 4 August 2005; accepted 18 December 2005

Abstract

Inpainting and denoising are two important tasks in the field of image processing with broad applications in image and vision analysis. In this paper, we present a new approach for image restoration. Our method simultaneously fills in missing, corrupted, or undesirable information while it removes noise. The denoising is performed by the smoothing equation working inside and outside of the inpainting domain but in completely different ways. Inside the inpainting domain, the smoothing is carried out by the Mean Curvature Flow, while the smoothing of the outside of the inpainting domain is carried out in a way as to encourage smoothing within a region and discourage smoothing across boundaries. Besides smoothing, the approach here presented permits the transportation of available information from the outside towards the inside of the inpainting domain. This combination permits the simultaneous use of filling-in and differentiated smoothing of different regions of an image. The experimental results show the effective performance of the combination of these two procedures in restoring scratched photos, disocclusion (or removal of entire objects from the image) in vision analysis and text removal from images.

Keywords: Inpaint; Image processing; Noise removal; Edge detection; Diffusion equation; Transport equation

1. Introduction

Inpainting is a practice carried out by artists when modifying a picture, in such a way so that an observer is unable to detect any changes. The terminology for digital inpainting was first introduced by Bertalmío, Sapiro, Caseles and Ballester [7] where these authors introduced a partial differential equation model based on the transport theory. The inpainting process consists of, in general, the filling-in of missing information within a domain D or to replace this domain with a different kind of information, based upon the image information available outside of the domain D. This domain is referred to as the inpainting domain and is where the original image has been damaged due to age action or also the region that we desire to add or remove information. The filling-in of missing information and the removal of noise are two very important topics in image processing, with several applications such as image coding and wireless image transmission (e.g. recovering lost blocks), special effects (e.g. removal of objects) and image restoration (e.g. fold lines, scratches and noise removal). The basic idea of inpainting algorithms is to fill-in regions with available information from their surroundings. In most cases, the available data of the original image is noisy which makes it necessary to eliminate the noise and fill-in the blank spaces (those without information). The basic idea of our algorithm is to complete these spaces which hold no information and eliminate noise (if exists) while preserving the edges, and the goal of this work is to recover the entire clean image $u(x)$ from a given incomplete noisy image $I(x)$ observed only outside of an inpainting domain D, performing the inpainting and denoising action simultaneously. The fill-in and the denoising procedures are carried out automatically, and are independent of the topology of the objects that compose the image or of the inpainting domain D, which could be a union of several disconnected regions.

In Section 2, we briefly describe the inpainting scheme given in [6,7] and the Mean Curvature Flow model. The proposed model for inpainting and denoising is described in Section 3. The Euler discretization of the proposed model and the numerical implementation are discussed in Section 4 as well as showing the application of our model in various examples, including denoising and restoring scratched pictures, texts and object removal from images. A note emphasizing the necessity of the diffusion process is presented

* Corresponding author. Address: Department of Computing, Federal University of Goiás, Campus Catalão, Caixa Postal 56, CEP 75704-020, Catalão, GO, Brazil. Tel.: +55 64 34411500

E-mail addresses: celiazb@ufu.br (C.A.Z. Barcelos), marcos.4d@uol.com.br (M.A. Batista).

0262-8856/$ - see front matter
doi:10.1016/j.imavis.2005.12.008

in Section 5 and the concluding remarks are presented in Section 6.

Throughout the paper, Ω denotes the entire domain of the image, D the inpainting domain, D^C the available part of the image I on Ω and u the restored image.

2. Image inpainting and image denoising

Inpainting is the art of rebuilding the basics of visual art and consists of filling-in unknown data in a known region of an image, with the principal objective of restoring harmony to the given damaged picture with parts worn by time, overexposed objects or objects we want to remove from the image.

To fill-in the inpainting domain, in a given picture, professionals follow these basic rules:

(1) Observe the image as a whole.
(2) Paint the inpainting domain in the same tones found in the surrounding background regions.
(3) Using, as a basis, similar regions of the same image or known images of the same theme to carry out repairs in damaged regions and then add texture.

In [6,7], Bertalmío et al. introduced the digital inpainting concept and proposed the Partial Differential Equations (PDE's) model to simulate the digital work of the artisan's art of inpainting. As the work of these artisans, is highly subjective, they limit the digital simulation to the automatization of the item (2), and for this, they proposed to transport the information from the surrounding inpainting domain into its interior always following the direction of the isophotes, arriving at ∂D. In order to make the transport of information follow this proposal, Bertalmio uses the inner product between two vectors, one indicates the inpainting's normal direction ($\nabla(\Delta u)$) and the other indicates the isophote direction ($\nabla^{\perp} u$). Following this procedure, the inpainting domain receives information and thus decreases.

The goal of the inpainting model introduced in [6,7] is to transport as smoothly as possible (along the isophotes) the information from the surrounding inpainting domain. The transport of information is performed by solving the following PDE

$$u_t = \nabla(\Delta u)\cdot\nabla^{\perp} u, \quad x \in D, \quad t > 0 \tag{1}$$

where t is a scale space parameter as in [2,3,12]. Note that this evolution equation runs only inside the region to be inpainted D. In Eq. (1), $\nabla^{\perp} u$ (Fig. 1a), is a vector which indicates the direction of $u(x)$ variation of least intensity. The absolute value of $\nabla^{\perp} u$ is numerically equal to the instant variation rate of $u(x)$ along the isophotes. The vector $\nabla(\Delta u)$ indicates the direction where the value of $u(x)$ varies abruptly (indicating a border or edge) and its absolute value is numerically equal to the Laplacian instant variation rate in this direction (Fig. 1b). The inner product of these two vectors (Fig. 1c) gives us the value to be transported to the pixel in question. We observe that in the case where the two vectors are perpendicular the transport will not be carried out. In this case, we have a strong indication that the pixel in question is localized on a border. In other cases where the transport does not happen $\nabla(\Delta u)$, or $\nabla^{\perp} u$, is a void vector. For more details of this approach, see [6,7].

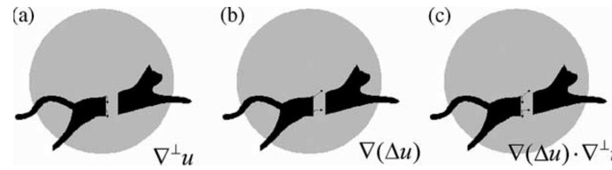

Fig. 1. (a) The perpendicular vector to ∇u, (b) the $\nabla(\Delta u)$ vector and (c) the two vectors showing where the inpainting will be performed, one notices that from the lowest lying pixel there will not be any transport of information once the vector $\nabla(\Delta u)$ and $\nabla^{\perp} u$ are perpendicular.

The inpainting method proposed in [6,7] consists of intercalating the Eq. (1) with a diffusion equation whose objective is 'to ensure a correct evolution of the direction field' [6]. They used the following anisotropic diffusion equation

$$u_t = g_\varepsilon K|\nabla u|, \quad x \in D^\varepsilon \tag{2}$$

where D^ε is a dilation of D with radius ε, K is the Euclidean curvature of u, and g_ε is a smooth function in D^ε such that $g_\varepsilon = 0$ in ∂D^ε and $g_\varepsilon = 1$ in D^ε.[1] We will refer to this model as the BSCB model.

In this paper, we propose a modification of the BSCB model, which consists of using different diffusion processes at points whether they belong to the inpainting domain or not. The equation used in D^C makes a differentiated diffusion at points whether or not they belong to the edges of the objects that compose the image. In this way, the noise will be eliminated and the edges preserved. Such an equation enriches the inpainting process eliminating any eventual noise in the image, at the same time a correct evolution is possible in the direction of the transportation and stabilizes the process, eliminating the necessity to carry out a diffusion in D^ε which in practice creates the 'frame effect', this means around the vicinity of the inpainting domain, within the dilation of D, the data is smoothed.

Several PDE-based techniques have been proposed for the smoothing of an image, some obtained on the direct derivation of the evolution equations, and others from energy approaches such as the L^2 norm dependent model or from the total variation model of Rudin, Osher and Fatemi [16]. Here, we will smooth the data inside the inpainting domain using the Mean Curvature Flow (MCF) equation (see [1,2] for further details), which is given by

$$u_t = |\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

This equation diffuses u in the direction orthogonal to its gradient ∇u and does not diffuse in the direction of ∇u. The level sets of the solution of the MCF equation move in the

[1] In the numerical implementations, Eq. (2) was not implemented in D^ε, a non-linear scaling was used to stabilize the discrete equation (see [6], p. 66).