J. Vis. Commun. Image R. 48 (2017) 43–53

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Non-texture image inpainting using histogram of oriented gradients ☆

Vahid K. Alilou [a], Farzin Yaghmaee [b,*]

[a] Department of Electrical and Computer Engineering, Semnan University, Semnan, Iran
[b] Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran

ARTICLE INFO

Article history:
Received 4 September 2016
Revised 4 April 2017
Accepted 9 June 2017
Available online 11 June 2017

Keywords:
Image inpainting
Histogram of oriented gradients
Dominant orientation
Local gradients

ABSTRACT

This paper presents a novel and efficient algorithm for non-texture inpainting of images based on using the dominant orientation of local gradients. It first introduces the concept of a new matrix called *orientation matrix* and then uses it for faster and better inpainting. The process of propagating information is carried out by using a new formulation which leads to much more efficient computations than the previous methods. The gain is both in terms of computational complexity and visual quality. The promising results in contexts of text, scratch, and block loss inpainting demonstrate the effectiveness of the proposed method.

1. Introduction

Image inpainting is the problem of repairing/recovering parts of an image that have been damaged or partially occluded by undesired objects [1]. This problem has attracted the interest of the scientific community in the last decade, mainly due to the growth of important applications such as image editing [2,3], image replacement [4], noise removal [5,6], image restoration [7,8], compression [9–11], disocclusion [12], zooming and super-resolution [13], error concealment [14,15], and video inpainting [16–18].

Digital image inpainting is a technology lying at the intersection of computer graphics, image, and signal processing. The basic idea in this technology is to develop such an algorithm that imitates the way professional art repairers fix damages in paintings or sculptures.

Formally, the problem of inpainting can be defined as follows: "Given an input image $\boldsymbol{I}$ with a target region Ω try to fill-in all pixels of the target region by using the information of the known areas $\Phi = I - \Omega$" [19].

The research in this field was pioneered by Bertalmio et al. [1]. They introduced an inpainting approach based on linear partial differential equations. After that, many inpainting methods were proposed to address this problem [20–22]. The inpainting methods found in the literature can be categorized into two major categories: *diffusion-based* and *exemplar-based*.

The first category is diffusion-based. The key idea behind these methods is the smooth propagation of information from the boundary toward the interior of the missing areas along the direction of isophote lines (lines of equal gray values). It is the same way as professional painting restorators work on damaged artworks [23].

The second category is exemplar-based in which the propagation of information is carried out through a copy-and-paste texture synthesis procedure. This terminology refers to the methods that synthesize entire patches by learning from patches in the known part of the image. This idea was initially proposed by the work of Efros et al. [24]. Afterward Criminisi et al. [25] designed a revolutionary algorithm for removing large objects based on texture synthesis with giving higher priority to linear structures. After that many other exemplar-based methods have been proposed based on this work. For example, the idea of using non-local means of a set of candidate patches was proposed by Wong et al. [26]. Xu and Sun [27] proposed a method in which the priorities of patches are calculated based on the sparseness of the patch's non-zero similarities to its neighboring patches. Kumar et al. [28] added a new "edge length" term in the priority equation of [25] which leads to propagation of linear structures in a better way. In their very recent work [29], they formulated the exemplar-based image inpainting as a metric labeling problem and solved it using simulated annealing algorithm. Many other exemplar-based approaches [30–33] can be found in the literature.

The methods proposed in both categories are interesting and efficient in different situations. The patch-based texture synthesis technique of the exemplar-based methods provides significant advantages which make it possible to inpaint large holes in images.

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail address: f_yaghmaee@semnan.ac.ir (F. Yaghmaee).

http://dx.doi.org/10.1016/j.jvcir.2017.06.003
1047-3203/

Diffusion-based methods on the other hand propose different formulations which enable the algorithm to add some new information to the image rather than performing only copy-and-paste procedure.

The diffusion-based methods are well founded on the theory of partial differential equation (PDE) and variational formulation. They, in general, avoid having unconnected edges that are perceptually annoying and can achieve convincingly excellent results for filling non-textured or relatively small missing regions. However, they suffer from blur close to edges and contours. Some new works have been proposed to solve this problem. For example, Benzarti and Amiri [34] proposed a new method based on using non-linear diffusion tensor. This technique is more effective than the use of a simple diffusion gradient due to the reliable information it can give in the presence of complex structures. Another work is proposed by Bornemann and März [35]. They propose to match the high-level quality of the methods Bertalmio et al. [1] and Tschumperl [36] to develop a very fast inpainting algorithm. Their algorithm is established based on two observations. The first is about the celebrated method of Bertalmio et al. [1] which is, neglecting some anisotropic diffusion steps that are interleaved for stabilization. The second is about discrete, linear, and single-pass inpainting method of Telea [37]. As a result, they successfully designed a fast inpainting algorithm which behaves strongly diffusive and creates peculiar transport patterns.

Thanks to advances in image processing, new methods have been proposed for automatic detection of inpainting areas. Bertalmio et al. [20] introduced a class of automated methods for digital image/video inpainting. Venkatesh et al. [38] proposed an efficient video inpainting algorithm capable of addressing inpainting under the stationary camera and moving cameras. The stationary background region is filled by a combination of adaptive background replacement and image inpainting technique. Recently, Mosleh et al. [17] presented a two-stage framework for automatic video text removal in order to detect and remove embedded video texts and fill-in their remaining regions by appropriate data.

The contribution of this paper is to develop a new practical non-texture image inpainting approach with the aim of recovering scratches, texts, or block losses from an image with a very low computational burden. This approach makes it possible to build a practical image editing software that can be used in various devices such as mobile phones and tablets. An extension of this work can also be used for the application of automatic video inpainting. The remainder of this paper is organized as follows: Section 2 describes the proposed method. In Section 3 we perform experiments and show the results and next we conclude this work in Section 4.

2. Proposed approach

The key idea behind our algorithm is to use a new matrix we call it *orientation matrix*. The overall architecture of the proposed inpainting system is shown in Fig. 1. At the first step the algorithm calculates the gradients of the input image in both x and y directions and then computes our orientation matrix using these gradients.

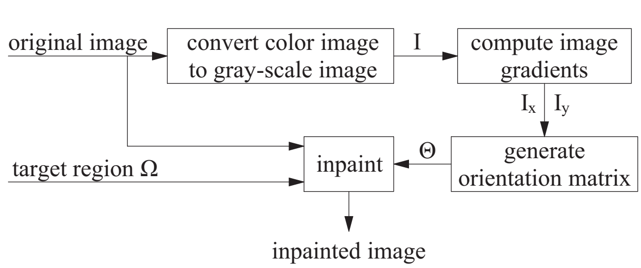

Fig. 1. The overall architecture of the proposed system.

2.1. Orientation matrix

For a given input image $\boldsymbol{I}$, we introduce the following equation for calculating our orientation matrix $\boldsymbol{\Theta}$:

$$\boldsymbol{\Theta}(x,y) = round\left(\left(\frac{\arctan\left(\frac{\boldsymbol{I}_x(x,y)}{\boldsymbol{I}_y(x,y)}\right)}{\pi} + \frac{1}{2}\right) \times n\right) \quad (1)$$

in which $\boldsymbol{I}_x$ and $\boldsymbol{I}_y$ are gradients in x and y directions. This formula generates a matrix containing integer values ranging from 0 to n. Each value corresponds to an angle in the range of $-\frac{\pi}{2}$ to $+\frac{\pi}{2}$. In our implementations, n is considered to be 16. However, one may choose a different value for the quantization of the gradients. Fig. 2 shows these values and their corresponding angles. Since $\frac{-\pi}{2}$ and $\frac{\pi}{2}$ are at the same direction, we use:

$$\boldsymbol{\Theta}(x,y) = \begin{cases} \boldsymbol{\Theta}(x,y), & \boldsymbol{\Theta}(x,y) > 0 \\ n, & \text{otherwise} \end{cases} \quad (2)$$

This means we limit our quantization over exactly n different integer values (each corresponds to a specific orientation).

Now, we have a matrix of the same size as the original image $\boldsymbol{I}$ in which each pixel (x,y) contains an integer value in the range of 1–n. Each value represents the orientation of the gradient of the image at pixel (x,y).

To better show the characteristics of this matrix, we took the well-known *Cameraman* image and then calculated the orientation matrix for this image. Then we expanded it in n different binary images which have been shown in Fig. 3. As you can see in the images, each different value (from 1 to 16) captures the oriented gradients of the original image in the direction of its corresponding orientation.

2.2. Filling process

Here, we describe the filling process of the unknown areas using our pre-calculated orientation matrix. But, before we proceed with the description of this stage, we provide the important notations we will use in this paper (see Table 1). Fig. 4 demonstrates our notation diagram. This figure illustrates an input image $\boldsymbol{I}$ with a target region Ω. The goal is to estimate the color components of all pixels of the target region using the information exists in the source region Φ.

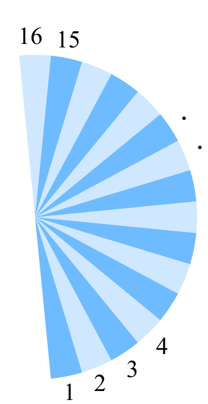

Fig. 2. Orientation angles and their corresponding values θ ($n := 16$).