Int. J. Electron. Commun. (AEÜ) 69 (2015) 307–313

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

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

Salt and pepper noise removal with image inpainting

Xianquan Zhang [a,b,c,*], Feng Ding [a], Zhenjun Tang [a,c], Chunqiang Yu [a,b,c]

[a] Department of Computer Science, Guangxi Normal University, Guilin 541004, PR China
[b] Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin 541004, PR China
[c] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, PR China

ARTICLE INFO

Article history:
Received 9 October 2013
Accepted 27 September 2014

Keywords:
Salt and pepper noise
Noise removal
Image denoising
Image inpainting
Image restoration

ABSTRACT

Salt and pepper noise removal is an important task in image processing. In this paper, we propose a simple and efficient restoration algorithm with the theory of image inpainting. Our algorithm takes noisy pixels as missing data for inpainting, adaptively selects convolution mask in terms of details of local regions, and achieves restoration by iterative convolutions. Many experiments are conducted to validate efficiency of our algorithm. The results show that our algorithm can efficiently remove noise while preserving image details even at a high noise density, and outperforms some well-known algorithms in terms of peak signal-to-noise ratio.

1. Introduction

Digital images are easily corrupted by salt and pepper noise during the transmission, which greatly degrades visual quality of the images. Therefore, developing efficient techniques for removing salt and pepper noise and simultaneously preserving image details becomes an important task in image processing. In the past years, various useful algorithms have been reported. A conventional well-known algorithm is the standard median filter (SMF) [1]. The SMF can reach good performances in noise removal and speed, but applies convolution operation to all pixels no matter whether or not they are noisy pixels. This strategy produces blurred images around edge regions. To overcome this weakness, researchers have proposed some improved techniques combining noise detection with image filtering. For example, Wang and Zhang [2] proposed a progressive switching median filter (PSMF), which exploits noise detector and noise filter in iterative manners to remove noises. Zhang and Karim [3] presented a new impulse noise detection technique for switching median filters based on one-dimensional Laplacian operators. However, as the noise density increases, visual qualities of the restored images significantly degrade. Chan et al. [4] gave a scheme with adaptive median filter and regularization method for removing salt and pepper noise. This scheme can preserve image details, but its computational cost is high. Chang et al. [5] introduced an adaptive median filter (AMF), which is better than SMF, but is unsuitable for corrupted images with high noise density and those images with rich textures. In Ref. [6], Wang and Wu proposed a noise detection and filtering algorithm, which can remove a wide range impulse noise while preserving image details. In another work [7], Kang and Wang modified the switching median filter by adding one more noise detector to improve capability of noise removal. In Ref. [8], Fabijańska and Sankowski introduced another modified switching median filter by analyzing local intensity extrema. Besides the above SMF based methods, other non-SMF based techniques have been also reported. For example, Awad and Man [9] proposed a noise removal approach based on a similar neighbor criterion. Mélange et al. [10] presented a method based on fuzzy logic for corrupted image sequences. Wu and Tang [11] introduced a scheme based on an impulse noise detector and the edge-preserving total variation inpainting (TVI) model. This TVI-based filter (TVIF) can preserve image details well, but its computational cost is also high.

Although many algorithms for removing salt and pepper noise are reported, there are still some problems in practice. For example, visual qualities of restored images are not good enough under high noise density. In this paper, we propose an efficient method with the theory of image inpainting (II) [12]. Our method takes those corrupted pixels as missing data for inpainting, adaptively selects convolution mask in terms of their local characteristics, and finally achieves restoration by iterative convolutions, which can deliver reliable information to the corrupted pixels. We conduct many experiments to validate efficiency of our method. The results show that, even though digital images are corrupted by salt and pepper noises with high densities, our method can restore them with good

* Corresponding author at: Department of Computer Science, Guangxi Normal University, No. 15 YuCai Road, Guilin 541004, PR China. Tel.: +86 18078337062.
E-mail addresses: zxq6622@163.com (X. Zhang), 648489640@qq.com (F. Ding), tangzj230@163.com (Z. Tang), 397578853@qq.com (C. Yu).

http://dx.doi.org/10.1016/j.aeue.2014.09.018
1434-8411/

visual quality. The rest of this paper is organized as follows. Section 2 introduces our algorithm. Section 3 discusses experimental results. Conclusions are finally drawn in Section 4.

2. Proposed algorithm

Our method firstly detects noisy pixels from the corrupted image. During the restoration, our method keeps those noise-free pixels unchanged, and inpaints those noisy pixels by iterative convolutions. Section 2.1 introduces noisy pixel identification, Section 2.2 presents selection of convolution mask, and Section 2.3 explains calculation of convolution masks.

2.1. Noisy pixel identification

Efficient noise detection plays an important role in noise removal algorithm. It can efficiently reduce false detection and missing detection. False detection mistakenly takes some noise-free pixels as noisy pixels. Consequently, convolution operations blur the edges around these wrong noisy pixels. Missing detection cannot find all noisy pixels and then visual quality of the restored image is restrained. In literature, many useful noise detection methods have been reported, such as [13–15]. Here, we exploit the method [15] to identify image noises. This detector [15] defines a 3×3 window, and exploits relationship between the maximum and minimum values of the pre-window and current widow to find noisy pixels. It is a simple and efficient strategy for noise detection, and thus makes our algorithm fast speed. In experiments, we took standard 256×256 Lena as test image, added salt and pepper noise with 0.9 density, marked the real noisy pixels and counted the noise pixels detected by [15]. We find that, the number of real noisy pixels is 58982 and the number of detection errors between the real noisy pixels and the detected noisy pixels is 93. Let $p_{i,j}$ (i = 1,2,. . ., M; j = 1,2,. . ., N) be the pixel in the i-th row and the j-th column of the corrupted image. Thus, we use $m_{i,j}$ (i = 1,2,. . ., M; j = 1,2,. . ., N) to indicate whether or not $p_{i,j}$ is a noisy pixel. The definition of $m_{i,j}$ is as follows.

$$m_{i,j} = \begin{cases} 0 & \text{If } p_{i,j} \text{ is a salt or pepper noise} \\ 1 & \text{Otherwise} \end{cases} \quad (1)$$

Clearly, $m_{i,j} = 1$ means that $p_{i,j}$ is a noise-free pixel, and $m_{i,j} = 0$ implies that $p_{i,j}$ is a noisy pixel.

2.2. Convolution mask selection

In this study, we define two kinds of convolution masks, i.e., 3×3 mask and directional mask. This section discusses selection of convolution mask. Detailed calculation of each convolution mask is presented in Section 2.3.

For those noisy pixels in smooth regions, we use a simple convolution window with a $k \times k$ mask to remove noise and keep local region smooth. However, for those noisy pixels in textural regions, if we still use a simple $k \times k$ convolution mask, the textural details cannot be preserved and the corresponding regions are blurred. In fact, textural regions have dominant directions which should be used in noise removal. Here, we propose convolution masks along four directions as shown in Fig. 1, and adaptively select a convolution mask according to the dominant direction of the region of noisy pixels. As region direction is considered in convolution operation, visual quality of restored image is efficiently improved.

During noise removal, we need to choose convolution masks for noisy pixels. This is done as follows. Take the noisy pixel as the center of convolution mask. If there are a few noise-free pixels in the convolution window, selection of directional convolution mask induces false edges. In this case, we calculate the numbers of noise-free pixels in the four directional convolution masks. If one of the four pixel numbers is smaller than a threshold R, we select a 3×3 convolution mask for the noisy pixel. Otherwise, we choose a directional mask according to the direction of local region.

Note that, for those pixels located at the image border, directional masks cannot be used when their distances to the border are smaller than 3. In this case, we choose the nearest eight pixels around the noisy pixel $p_{i,j}$ to construct a 3×3 convolution mask.

2.3. Noise removal with convolutions

Here, we restore corrupted images with the theory of image inpainting. For those noise-free pixels, we keep their values unchanged. For those noisy pixels, we adaptively select convolution mask according to its local details, and perform iterative filtering. Let p be a noisy pixel, and $\mathbf{R}_p$ be a set forming by other pixels in the convolution mask. Detailed calculations of the 3×3 convolution mask and directional convolution mask are as follows.

(1) 3×3 convolution mask

Suppose that r_p is a pixel in the convolution mask of p, i.e., $r_p \in \mathbf{R}_p$, $V^{(0)}(r_p)$ is the initial value of r_p (i.e., the original value in the corrupted image), $V^{(n-1)}(r_p)$ is the value of r_p after $n-1$ filtering iterations. Thus, the p value after n filtering iterations is defined as follows.

$$V^{(n)}(p) = \frac{1}{8} \sum_{r_p \in \mathbf{R}_p} V^{(n-1)}(r_p) \quad (2)$$

Clearly, all pixel values in the 3×3 mask are used to conduct iterative filtering, except the p value.

(2) Directional convolution masks

Let S_1, S_2, S_3, and S_4 be the directional convolution masks along vertical direction, horizontal direction, 45° direction, and 135° direction, respectively. We choose a directional mask S_t ($t \in \{1,2,3,4\}$) for p by the generalized standard deviation of noise-free pixels. Suppose that (i, j) are the coordinates of p, $\mathbf{C}_t$ is a set forming by those noise-free pixels in S_t, q is a noise-free pixel in S_t, and $I_t(q)$ is the q value, and K_t is the number of noise-free pixels in S_t. Thus, the mean of these noise-free pixels $\mu_t(i,j)$ is calculated by the following equation.

$$\mu_t(i,j) = \frac{1}{K_t} \sum_{q \in \mathbf{C}_t} I_t(q) \quad (t = 1, 2, 3, 4) \quad (3)$$

Therefore, the generalized standard deviation of noise-free pixels can be obtained as follows.

$$\sigma_t(i,j) = \sqrt{\frac{1}{K_t - 1} \sum_{q \in \mathbf{C}_t} (I_t(q) - \mu_t(i,j))^2} \quad (t = 1, 2, 3, 4) \quad (4)$$

We calculate all $\sigma_t(i,j)$ ($t \in \{1,2,3,4\}$), find the minimum value, and take the directional mask with minimum value S_{min} as the convolution window.

In our directional convolution masks, the distances from noise-free pixels to noisy pixel are different. In general, the smaller the distance, the more important the noise-free pixel is. This means that the importance of each noise-free pixel should be considered in convolution operation. Here, we use the distance of each noise-free pixel to construct its weight. Let (x, y) be the coordinates of the pixel r_p. Thus, the distance from r_p to p is measured by Euclidean distance.

$$d(r_p, p) = \sqrt{(i-x)^2 + (j-y)^2} \quad (5)$$