J. Vis. Commun. Image R. 25 (2014) 748–754

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Salt and pepper noise removal in binary images using image block prior probabilities

Stanislav Pyatykh *, Jürgen Hesser

University Medical Center Mannheim, Heidelberg University, Theodor-Kutzer-Ufer 1-3, D-68167 Mannheim, Germany

ARTICLE INFO

Article history:
Received 18 February 2013
Accepted 31 January 2014
Available online 15 February 2014

Keywords:
Salt and pepper noise
Image denoising
Binary images
Bayesian inference
Training methods
Impulse noise
Image restoration
Image filtering

ABSTRACT

During scanning and transmission, images can be corrupted by salt and pepper noise, which negatively affects the quality of subsequent graphic vectorization or text recognition. In this paper, we present a new algorithm for salt and pepper noise suppression in binary images. The algorithm consists of the computation of block prior probabilities from training noise-free images; noise level estimation; and the maximum a posteriori probability estimation of each image block. Our experiments show that the proposed method performs significantly better than the state of the art techniques.

1. Introduction

Image scanning and transmission often results in corruption of the image by salt and pepper noise [1–3], which appears in binary images as randomly distributed inverted pixels. Removal of this type of noise plays an important role for further processing steps, such as graphic vectorization and text recognition.

Salt and pepper noise suppression algorithms for binary images are essentially different from those for grayscale images. In a binary image, image objects and corrupted pixels have the same set of values, namely, $\{0, 1\}$, so that the detection of corrupted pixels becomes a nontrivial task. However, once a pixel is detected as corrupted, its value can be simply inverted. In a grayscale image, corrupted pixels can be easily identified, because their values are significantly different from the values of the neighbor pixels, but the estimation of their true values is difficult.

For binary images, several denoising algorithms have been proposed in the last two decades:

1. The median filter and the center weighted median filter [4]. The median filter results in distortion of thin lines and corners. The center weighted median filter avoids this drawback by replicating the value of the central pixel several times before the calculation of the median, i.e. by giving more weight to the central pixel.
2. The kFill algorithm [5] and the Enhanced kFill algorithm [6]. In these algorithms, the $k \times k$ neighborhood of each pixel is analyzed; and the decision to invert the pixel value is based on the number of white pixels in the neighborhood, the number of connected components in the neighborhood, and the number of white pixels in the corners of the neighborhood. The disadvantage of these approaches is shortening one-pixel-wide lines from their end points [3].
3. Morphological operators [7], which remove salt and pepper noise using a sequence of erosions and dilations. Such filters tend to remove thin lines and join close objects, which limits their accuracy.
4. Filters utilizing connected component labeling [8,9]. These filters are based on the idea that corrupted pixels usually form connected components of small size. Therefore, removing small connected components can suppress the noise. Nevertheless, corrupted pixels connected to objects cannot be restored in this way.
5. Line tracking algorithms [10,3,11], which are developed specifically for denoising of engineering drawings. These algorithms track thin lines in order to identify if they are caused by the noise or belong to an object. These methods can be integrated into other algorithms, such as kFill [10], or they can be used as a post-processing step [11].

* Corresponding author.
E-mail addresses: stanislav.pyatykh@medma.uni-heidelberg.de (S. Pyatykh), juergen.hesser@medma.uni-heidelberg.de (J. Hesser).

http://dx.doi.org/10.1016/j.jvcir.2014.02.001
1047-3203/

6. The discrete universal denoiser (DUDE) [12], which counts the occurrences of each pixel neighborhood in the whole image and estimates the value of a pixel using the occurrence count of its neighborhood.

The drawback of the currently existing methods is that only the input noisy image is used; and the prior knowledge about the objects, which are typical for the processed image, is not utilized. At the same time, this knowledge has been successfully used for grayscale image restoration [13,14].

In this article, we propose a new method for salt and pepper noise suppression in binary images, which is based on the maximum a posteriori probability (MAP) estimation of each image block. The algorithm utilizes training noise-free images to compute the prior probability of the occurrence of each image block. A noise level estimation procedure has been developed as well.

The article is organized as follows. The notation is introduced in Section 2. Section 3 focuses on the noise model. The noise level estimation algorithm is presented in Section 4. We present the noise suppression method in Section 5 and discuss its implementation in Section 6. The experiments and the discussion are given in Sections 7 and 8 respectively. We conclude in Section 9.

2. Notation

Further in this paper, images and image blocks are considered as matrices; images are denoted by capital letters and image blocks are denoted by bold capital letters. All matrices are numbered using the superscript, e.g. $A^{(1)}, A^{(2)}, \ldots$; and the (i,j) element of matrix A is denoted by A_{ij}. The set of all $M \times M$ matrices with elements from domain $\{0,1\}$ is denoted by $\{0,1\}^{M\times M}$. Let $S(A)$ be the sum of the absolute values of elements of matrix A. Then $S(A-B)$ is the number of different elements in matrices $A, B \in \{0,1\}^{M\times M}$. The binomial distribution is denoted by $B(n,p)$; and $\binom{n}{k}$ is the binomial coefficient.

Let us also associate 0 with black and 1 with white.

3. Noise model

Let U be the original, noise-free image; p be the noise level, i.e. the probability of a pixel value flip; and V be the corresponding noisy image. That means, V_{xy} has the following probability mass function for each pixel (x,y):

$$f(t;p,U_{xy}) = \begin{cases} 1-p & \text{if } t = U_{xy} \\ p & \text{if } t = 1-U_{xy} \end{cases}. \tag{1}$$

Without loss of generality, we can assume that $p \leqslant 0.5$. Indeed, if $p > 0.5$, the matrix with elements $1-U_{xy}$ can be considered as the original image (see (1)).

Let W be the image width and H be the image height. Each of images U and V contains $N_B = (W-M+1)(H-M+1)$ blocks of size $M \times M$. Let us denote these blocks by $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ respectively. $\mathbf{V}^{(k)}_{ij}$ has the following probability mass function:

$$f\left(t;p,\mathbf{U}^{(k)}_{ij}\right) = \begin{cases} 1-p & \text{if } t = \mathbf{U}^{(k)}_{ij} \\ p & \text{if } t = 1-\mathbf{U}^{(k)}_{ij} \end{cases} \quad i,j = 1,\ldots,M. \tag{2}$$

4. Noise level estimation

Let us denote the histogram of sums $S(\mathbf{V}^{(k)})$ by $h(t)$:

$$h(t) = \sum_{k=1}^{N_B} [S(\mathbf{V}^{(k)}) = t] \quad t = 0,\ldots,M^2 \tag{3}$$

where $[S(\mathbf{V}^{(k)}) = t]$ equals 1 if $S(\mathbf{V}^{(k)}) = t$ and 0 otherwise.

Assume that image U contains N_0 black homogeneous blocks $\mathbf{U}^{(k)}$. Since these blocks contain only zeros, the sums $S(\mathbf{V}^{(k)})$ corresponding to these blocks follow the binomial distribution $B(M^2,p)$. Therefore, the expected value of the histogram of these sums is $N_0 f_B(t;M^2,p)$, where $f_B(t;M^2,p)$ is the probability mass function of the binomial distribution $B(M^2,p)$. Hence, in order to estimate N_0 and p, the distance between $h(t)$ and $N_0 f_B(t;M^2,p)$ is minimized:

$$\widehat{N}_0, \hat{p}_0 = \arg\min_{N_0,p} \sum_{t=0}^{M^2} \left(h(t) - N_0 f_B(t;M^2,p)\right)^2 \tag{4}$$

The minimization is presented in Algorithm 1. $\hat{N}_0$ is initialized with the total number of blocks N_B and i_{max} optimization iterations are done then. At each iteration, $\hat{p}_0$ is found by the enumeration of p from 0.0001 to 0.5 with step 0.0001 (line 4). Then, $\hat{N}_0$ is computed as zero of the derivative of the objective function (line 5). In our experiments, no more than 7 iterations were required for convergence.

Algorithm 1

Input: image V corrupted with salt and pepper noise
Output: black block count estimate $\hat{N}_0$, noise level estimate $\hat{p}_0$
1: $\hat{N}_0 \leftarrow N_B$
2: $\hat{p}_0 \leftarrow 0$
3: **for** $i = 1$ **to** i_{max} **do**
4: $\quad \hat{p}_0 \leftarrow \arg\min_p \sum_{t=0}^{M^2} \left(h(t) - \hat{N}_0 f_B(t;M^2,p)\right)^2$
5: $\quad \hat{N}_0 \leftarrow \sum_{t=0}^{M^2} h(t) f_B(t;M^2,\hat{p}_0) / \sum_{t=0}^{M^2} f_B^2(t;M^2,\hat{p}_0)$
6: **end for**

The same procedure is applied for white homogeneous blocks. Namely, the number N_1 of white homogeneous blocks and the noise level are estimated as

$$\widehat{N}_1, \hat{p}_1 = \arg\min_{N_1,p} \sum_{t=0}^{M^2} \left(h(t) - N_1 f_B(t;M^2,1-p)\right)^2. \tag{5}$$

Then, the final noise level estimate p_{est} is selected based on the predominance of black or white homogeneous blocks:

$$p_{est} = \begin{cases} \hat{p}_0, & \text{if } \hat{N}_0 \geqslant \hat{N}_1 \\ \hat{p}_1, & \text{if } \hat{N}_0 < \hat{N}_1 \end{cases} \tag{6}$$

5. Noise suppression

Let $Z^{(q)}, \ q = 1,\ldots,T$ be noise-free images of size $W_q \times H_q$, which contain information of the same kind as that contained in U. These images are used for training. Each image $Z^{(q)}$ contains $N_q = (W_q - M + 1)(H_q - M + 1)$ blocks of size $M \times M$, which are denoted by $\mathbf{Z}^{(q,n)}, \ n = 1,\ldots,N_q$. The prior block probabilities are computed as

$$P(\mathbf{A}) = \frac{1}{\sum_{q=1}^{T} N_q} \sum_{q=1}^{T} \sum_{n=1}^{N_q} [\mathbf{A} = \mathbf{Z}^{(q,n)}] \quad \mathbf{A} \in \{0,1\}^{M\times M} \tag{7}$$

where $[\mathbf{A} = \mathbf{Z}^{(q,n)}]$ equals 1 if $\mathbf{A} = \mathbf{Z}^{(q,n)}$ and 0 otherwise. That means, the prior probability for block $\mathbf{A}$ is proportional to the number of occurrences of this block in the training images.

For each block $\mathbf{V}^{(k)}$, we compute original block estimate $\hat{\mathbf{U}}^{(k)}$ as the MAP estimate:

$$\hat{\mathbf{U}}^{(k)} = \arg\max_{\mathbf{A}\in\{0,1\}^{M\times M}} P(\mathbf{V}^{(k)}|\mathbf{A}) P(\mathbf{A}) \quad k = 1,\ldots,N_B \tag{8}$$

where $P(\mathbf{V}^{(k)}|\mathbf{A})$ is the probability that block $\mathbf{A}$ will be equal to block $\mathbf{V}^{(k)}$ after corruption with salt and pepper noise of level p. This probability depends only on the number of different elements in blocks $\mathbf{V}^{(k)}$ and $\mathbf{A}$; and it can be computed as