SALT AND PEPPER NOISE DETECTION BASED ON NON-LOCAL MEANS

Carlos Junez-Ferreira, Fernando Velasco-Avalos
FIE, Universidad Michoacana de San Nicolás de Hidalgo, Ciudad Universitaria, Morelia, Mexico

Nelio Pastor-Gomez
FIC, Universidad Michoacana de San Nicolás de Hidalgo, Morelia, Mexico

Keywords: Image denoising, Salt and pepper noise, Non-local means, Adaptive median filter.

Abstract: Denoising is an important task inside the image processing area. In this paper, an algorithm for detecting and suppressing salt and pepper noise is presented. Firstly, the algorithm computes an estimation of the denoised image by using a variant of the Non-Local Means proposal. This estimation is segmented in order to detect corrupted pixels avoiding misclassifying pixels with extreme values that belong to objects on the uncorrupted image. Once pixels are classified, the algorithm performs a suppression step by using an adaptive median filter. Obtained results show that the implementation of this proposal gives good noise detection and suppression.

1 INTRODUCTION

Frequently, digital images are corrupted by undesirable random variations in intensity values, called noise. The presence of noise is due to several factors, among which are found the process of acquisition, compression and transmission of data. A common type of noise is impulsive noise, known as salt and pepper. This kind of noise randomly changes intensities of some pixels to the maximum ($v_{\max}$) or minimum ($v_{\min}$) values of the intensity range on the image (Chan et al., 2005).

The median filter has been widely used for impulsive noise suppression due to its ability to preserve edges in a better way than weighted averaging methods. It must be noticed that, median filter does not insure the edge preservation, mainly in images with high noise density. Numerous algorithms based on the median filter have been presented, as the proposed by Wang & Zhang (1999), Yuan & Tan (2006), and Chan et al. (2005).

2 IMPULSE NOISE DETECTION

Evidently, good noise detection is essential to carry out its suppression and to get a better restoration quality. One way to detect noise is through an adaptive median filter, as used by Chan et al. (2005), which is described below.

Let $\tilde{\boldsymbol{y}}$ the obtained image by applying an adaptive median filter to an image with salt and pepper noise $\boldsymbol{y}$. Moreover, according to the salt and pepper noise model, noisy pixels take its values from the set $\{v_{\min}, v_{\max}\}$. Then, it could be defined the set of corrupted pixels candidates as

$$N = \{y_{i,j} : \tilde{y}_{i,j} \neq y_{i,j}, y_{i,j} \in \{v_{\min}, v_{\max}\}\} \qquad (1)$$

for all $(i, j) \in A$.

Another proposal has been presented by Wang & Zhang (1999), where the following expression is used

$$N = \{y_{i,j} : |\tilde{y}_{i,j} - y_{i,j}| > T\} \qquad (2)$$

where $(i, j) \in A$ and T is a predefined threshold. Expression (2) is used iteratively in order to detect impulse noise.

3 NON-LOCAL MEANS

Buades et al. (2005) proposed the Non-Local Means

Junez-Ferreira C., Velasco-Avalos F. and Pastor-Gomez N. (2010).
SALT AND PEPPER NOISE DETECTION BASED ON NON-LOCAL MEANS.
In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pages 344-347
DOI: 10.5220/0002880703440347

algorithm, based on the idea that images contain repeated structures and that averaging these structures, the noise of an image can be reduced.

Given a discrete image with noise $\boldsymbol{y}$ the restored value $\hat{y}_{m,n}$, for the pixel at location $(m,n) \in A$, is computed as the weighted average of all pixels of the image,

$$\hat{y}_{m,n} = \sum_{(i,j)\in A} w_{i,j}^{m,n} y_{i,j} \tag{3}$$

where the family of weights $\{w_{i,j}^{m,n}\}$ depends on the similarity between pixels at positions $(m,n) \in A$ and $(i,j) \in A$, and it satisfies the conditions $0 \le w_{i,j}^{m,n} \le 1$ and $\sum_{(i,j)\in A} w_{i,j}^{m,n} = 1$.

Structural similarity between $y_{i,j}$ and $y_{m,n}$ depends on similarity between vectors $V(\Omega_{i,j})$ and $V(\Omega_{m,n})$, where $\Omega_{k,l}$ denotes a fixed size neighbourhood and centered at pixel $y_{k,l}$.

Similarity between above mentioned vectors is measured by a decreasing function of Euclidean distance, $d_{i,j}^{m,n} = \left\| V(\Omega_{m,n}) - V(\Omega_{i,j}) \right\|^2$. Pixels with similar neighbourhood to $V(\Omega_{m,n})$ will have large weights, which are defined as

$$w_{i,j}^{m,n} = \frac{1}{Z_{m,n}} e^{-\frac{d_{i,j}^{m,n}}{H^2}} \tag{4}$$

where $Z_{m,n}$ is a normalization constant, and the parameter H acts as a filtering degree, that is, it controls the decay of weights as a function of distances. For implementation purposes, a window of size W_1 is used to compute the average with a limited number of neighbours, instead of averaging all pixels of the image. Also, a window of size W_2 is used to define the structure of the neighbourhood and the size of vector $V(\Omega_{k,l})$.

In general, the Non-Local Means algorithm gives good results in terms of noise reduction, however, this does not always happen, especially in images with high salt and pepper noise level.

4 SALT AND PEPPER NOISE DETECTION AND SUPPRESION PROPOSAL

In this work, the Non-Local Means algorithm will be used to provide a preliminary estimate of the restored image, with the aim of detecting salt and pepper noise, even with high density.

In order to consider the presence of objects whose pixels intensity values are equal to the maximum or minimum values on image, in other words, black or white objects in the image without noise, we propose a segmentation, which is performed by grouping neighbouring pixels with similar intensity values, based on a threshold *TI*.

The purpose of making this segmentation is to find a partition $\boldsymbol{S}$ of an image $\hat{\boldsymbol{y}}$ on a set of regions, in such a way that

$$R_{big} \cup R_{small} = \hat{\boldsymbol{y}} \tag{5}$$

where

$$R_{big} = \{R_s : h(R_s) > P\}, \text{ and} \tag{6}$$

and,

$$R_{small} = \{R_s : h(R_s) \le P\}, \tag{7}$$

R_{big} is the set of regions considered relevant objects, that is, regions whose number of pixels represent a percentage of image greater than a threshold *P*. Obviously, R_{small} is the set of regions that are, by their size, regarded as details or noise. The function $h(R_s)$ computes the percentage of the image that corresponds to a region R_s. Thus, one can discriminate between pixels that can be considered corrupt and those that belong to an object, although in both cases the pixels intensities are extreme value.

Considering the above exposed, our proposal to detect salt and pepper noise can be described, in general, through the following steps:

```
ŷ =Compute_Estimation(y,TW,PAR)
S=Segmentation(ŷ,TI)
α =Pixels_Classification(y,S)
x̂ =Noise_Suppression(ŷ,α,Wmin,Wmax)
```

where `PAR` is a vector containing the parameters *H*, W_1 and W_2.

The preliminary estimate, described in Figure 1, is performed by calculating the weights of the pixels in the neighbourhood for a corrupt pixel candidate according to expression (4), and its estimated value will be the median value only of intensity values of pixels with weights greater than a threshold *TW*. In this way, only pixels whose structure is similar to the pixel in question are involved. The median value

```
Compute_Estimation(y,TW,PAR1)
  for each pixel at position (i,j)
    if (y_{i,j}=v_min) or (y_{i,j}=v_max)
      Compute family of weights w^{i,j}
      for each weight w^{i,j}_{m,n}
        if w^{i,j}_{m,n} ≥ TW
          VW ← VW ∪ y_{m,n}
        end
      end
      ŷ_{i,j} ← MEDIAN(VW)
    else
      ŷ_{i,j} ← y_{i,j}
    end
  end
  return(ŷ)
```

Figure 1: Pseudo-code of estimation computing.

```
Pixels_Clasification(y,S)
  for each pixel at position (i,j)
    if (y_{i,j}=v_min) or (y_{i,j}=v_max)
      if (i,j) ∈ R_small
        α_{i,j} ← 1
      else
        if (y_{i,j} ≠ ŷi,j)
          α_{i,j} ← 1
        else
          α_{i,j} ← 0
        end
      end
    else
      α_{i,j} ← 0
    end
  end
  return(α)
```

Figure 2: Pseudo-code of classification step.

is used instead of weighted averaging because its better results with salt and pepper noise.

Subsequently, this estimation will be segmented, as described above, then, a pixel is considered corrupt if its intensity value is an extreme value and the pixel belongs to a region considered irrelevant. If the pixel belongs to a relevant object and its value has changed at estimation, this pixel is also considered corrupted. Figure 2, shows the classification procedure, where if $\alpha_{i,j}=0$, the pixel at position (i,j) is regarded as corrupt and $\alpha_{i,j}=1$ if not.

Once pixels are classified, the algorithm proceeds to noise reduction, by applying an adaptive median filter. This kind of filtering considers geometric closeness in restoration. First, a neighbourhood for corrupt pixels is defined, then uncorrupted pixels belonging to the neighbourhood are stored in a vector, and the central pixel is replaced by median value of this vector. If the mentioned vector is empty, the window size is increased, repeating the procedure. If the size increases to reach a maximum size of window, the central pixel value is replaced by the median value of all pixels in neighbourhood. This procedure is described in Figure 3.

It can be determined that, the proposed algorithm is $O(N_D)$, where N_D is the number of data.

```
Noise_Suppression(ŷ,α,W_min,W_max)
  for each pixel at position (i,j)
    if α_{i,j} = 1
      W ← W_min, witness ← 0
      do
        VN ← φ
        for each element in window Ω_{i,j}
        of size W^2
          if α^{i,j}_{m,n} = 0
            VN ← VN ∪ ŷ_{m,n}
          end
        end
        if VN ≠ φ
          x̂_{i,j} ← MEDIAN(VN)
          witness ← 1
        else
          W ← W + 2
        end
      while (W ≤ W_max) & (witness = 0)
      if (W > W_max) & (witness = 0)
        x̂_{i,j} ← MEDIAN(Ω_{i,j} of size W^2_max)
      end
    else
      x̂_{i,j} ← ŷ_{i,j}
    end
  end
  return(x̂)
```

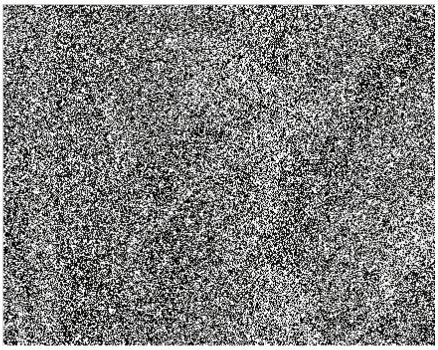
Figure 3: Pseudo-code of noise suppression.

5 EXPERIMENTAL RESULTS

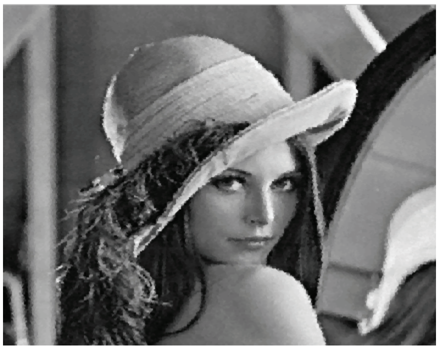
Experiments show that results obtained by applying the described algorithm are very good. The implementation of the code was made using Matlab 6.5 R13 on a PC 2.20 GHz Core Duo CPU.

The test image (Figure 4) was corrupted with salt and pepper noise with different densities through *imnoise* function of Matlab.

In order to compute the estimation, the values $W_1=3$, $W_2=3$, $TW=0.10$ and $H=10$ were used. For segmentation $T=0.001$ (the intensity values range of image is [0, 1]), and the value of P depends on the



(a)



(b)

Figure 4: Image 1, (a) Noisy image (80%), (b) Denoised image.

maximum size of a region that can be considered irrelevant, we use *P=0.001*. For noise reduction we used $W_{min}=3$ and $W_{max}=10$.

In order to quantify the performance of the detector, we used the values (Yuan & Tan, 2006) $\Gamma_1 = \gamma_g / \gamma_d$ and $\Gamma_2 = \gamma_g / \gamma_n$, where γ_d is the number of pixels detected with the proposed algorithm, γ_n is the number of pixels that are really corrupted and γ_g is the number of detected pixels that are really corrupted. Tables 1 and 2 show the performance of our proposal, which is identified as 1, for different noise densities on Lena image, compared with algorithm 2 (Chan et al., 2005) and 3 (Wang & Zhang, 1999). Table 3 shows the performance of the algorithm for noise reduction measured by PSNR and UIQI (Wang & Bovik, 2002).

Table 1: Performance of different detectors for Lena image with a noise density of 20%.

Algorithm	Γ_1	Γ_2	Time (s)
1	0.9988	1	17.59
2	0.7922	1	11.66
3	0.9332	0.9529	35.35

Table 2: Performance of different detectors for Lena image with a noise density of 80%.

Algorithm	Γ_1	Γ_2	Time (s)
1	0.9998	1	39.29
2	0.9994	1	26.31
3	0.9293	0.9731	35.20

Table 3: Performance of our noise suppression proposal for Lena image.

Noise (%)	PSNR (dB)	UIQI	Time (s)
20	41.7821	0.9735	21.06
50	34.9502	0.9105	36.45
80	29.4583	0.7786	52.10

6 CONCLUSIONS

This paper has presented an algorithm for detection and suppression of salt and pepper noise in digital images. The obtained results have shown that the proposed algorithm provides good results in acceptable time. In future work, we can consider adapting this algorithm for the suppression of different types of noise and using it as part of an image pre-processing step, in order to perform tasks such as segmentation and object recognition in a robust way.

REFERENCES

Buades, A., Coll, B., Morel, J. M., 2005. A non-local algorithm for image denoising. In *proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2, 60-65.

Chan, R. H., Ho, C. W., Nikolova, M., 2005. Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization. *IEEE Transactions on Image Processing*. Vol. 14(10), 1479-1485.

Wang, Z., Bovik, A. C., 2002. A universal image quality index. *IEEE Signal Processing Letters*. Vol. 9(3), 81-84.

Wang, Z., Zhang, D., 1999. Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*. Vol. 46(1), 78-80.

Yuan, S. Q., Tan, Y. H., 2006. Difference-Type noise detector for adaptive median filter. *IEEE Electronic Letters*. Vol. 42(8).