An Adaptive Edge Detection Algorithm for Images Corrupted with Impulse Noise

Krishna Kant Singh
Earthquake Engineering,
IIT Roorkee, Roorkee
India

Akansha Mehrotra
Earthquake Engineering
IIT Roorkee, Roorkee
India

Kirat Pal
Earthquake Engineering
IIT Roorkee, Roorkee
India

M. J. Nigam
E & C Engineering
IIT Roorkee, Roorkee
India

ABSTRACT

Edge detection is one of the most important tasks in the field of image processing. Detection of edges from noisy images is of greater importance as most images obtained are corrupted by impulse noise due to communication and transmission errors. In the proposed work a novel adaptive algorithm for finding edges of noisy images is proposed. One of the major problems with noisy images is that the noise pixels are also detected as edges, so in the proposed algorithm a threshold is used to distinguish between edge pixels and noise pixels. The value of threshold is inversely proportional to the level of details whose edges are detected and makes the algorithm adaptive. A sliding window is taken and the difference of the center pixel with all the pixels of the window whose value is not equal to zero or one is taken. But if the center pixel itself is zero or one then the difference from a noise free median is calculated. The average of the differences is checked against a threshold value. If the average value is above the threshold then it is a edge point otherwise it is noise based on this a binary image showing the edges is obtained.

Keywords

impulse noise, edge detection, adaptive.

1. INTRODUCTION

Edge detection is an important work for object recognition and is also an essential pre-processing step in image segmentation [1],[20]. Because edges represent important contour features in the corresponding image. Edges in images occur where change in grey level values is abrupt, i.e., places where some physical properties, such as illumination, geometry and reflectance of objects. The key issue of edge detection is to filter more noises while obtaining more edge details [2]. The traditional edge detection methods based on first-order gradient or second-order differentiation such as Sobel, Canny and Log are hard to efficiently extract the complex edge [3]. These edge detection methods are built for extracting edges of noise free images. In case of noisy images these algorithms perform poorly. Thus, the edge extraction from noisy images is gaining impetus. A edge detection algorithms that extract edges of noisy images is proposed in [4] which uses a separate wavelet based noise reduction block.

Noise is introduced into images during acquisition, signal amplification and transmission [5]. An important problem of image processing is to effectively remove noise from an image while keeping its features. Impulse noise is characterized by replacing a portion of an image pixels with noise values, leaving the remainder unchanged [6],[18],[19]. Such noise is introduced due to acquisition or transmission errors [7], [6], [8]. Nonlinear filters have been developed for removing impulse noise such as the traditional median filter [6]. Extensions of the median filter [9-14] are proposed to meet various criteria, e.g., robustness, preservation of edge, or preservation of details. But most of these suffer from the problem of being complex and also at high noise densities they do not give good results. In this paper a novel adaptive edge detection algorithm is proposed, the algorithm not only removes the noise but also finds edges efficiently. A threshold value is used to distinguish between edges and noise points. The algorithm finds the average of the differences of the pixels (whose value lies in between 0 and 1) of the window from the center pixel and if the center pixel is 0 or 1 then the difference is taken from a noise free median which is calculated iteratively by increasing the window. This average value is compared with a threshold value and if it is greater than the threshold then the center pixel is set to 1 otherwise 0, this way a binary image showing the edges of the image is formed.

2. NOISE MODEL

Impulse noise is one common noise type in communications [15,16,17]. It is also known as salt and pepper noise. In this type of noise the noise has either maximum (I_max) or minimum value (I_min) of the image intensity range. An image X corrupted with impulse noise is formulated as

\[ x_{i,j} = \begin{cases} n_{i,j}, \text{ with probability } p \\ s_{i,j}, \text{ with probability } 1-p \end{cases} \]

where \( n_{i,j} \in [I_{min},I_{max}] \) is the noisy impulse at the location \((i,j)\), \( p \) represents the probability of noisy pixel, \( n_{i,j}, s_{i,j} \) is the noise free pixel with probability \( 1-p \). Impulse noise appears as salt and pepper granules randomly distributed over the image.

3. PROPOSED WORK

This paper proposes an efficient and novel edge detection algorithm for images that are corrupted with impulse noise. Edges correspond to high frequency regions of an image, but due to the property of impulse noise that it is the minimum or maximum value poses a challenge to differentiate noisy and edge pixels from each other. To overcome this problem and find the edges of an image the proposed algorithm works as follows. For a sliding window \( W_M \) of size \( M \times M \), where \( M = 2L + 1 \) centered at \( x_i \), the algorithm first finds the similarity of the center pixel with its \((2L+1) \times (2L+1)-1\) neighbors, that is if \( L = 1 \), i.e., for a 3x3 window there will be 8 neighbors.

In this paper a 3x3 window is chosen. But when the similarity is calculated it is to be ensured that the center pixel itself is not noise and also only those neighbors are to be considered that are noise free. For an image with grey levels in the range \((0,1)\) a pixel is considered is impulse noise if it has the value
either 0 or 1. Similarly for an image with grey level in the range (0, 255) impulse noise has the value either 0 or 255. To resolve this problem two cases arise based on which the similarity of the window denoted by \( W_{av} \) is calculated, as follows

CASE I: If \( x_{ij} \neq 0 \) and \( x_{ij} \neq 1 \), this implies that the center pixel is noise free and the average value of the window is calculated by taking only those neighbors that do not have the value 0 or 1 as the neighbors \( N_r \) might also be noisy and they are not to be considered.

\[
W_{av} = \frac{1}{n} \sum_{r=0}^{n} |x_{ij} - N_r| \text{ where } 0 < N_r < 1 \quad ... (2)
\]

In equation (2), \( n \) is the number of neighbors \( N_r \) whose value lies in the range zero and one.

CASE II: If \( x_{ij} = 0 \) or \( x_{ij} = 1 \), this implies that the center pixel is noisy hence it cannot be utilized to calculate the window average. So the average is calculated using the median of the window, but again there is a possibility that this median comes out to be noise. Therefore, the median is calculated by increasing the window size by 2 each time until a noise free median \( (M_{nf}) \) is obtained. This noise free median is now used to calculate the window average similar to equation (1), as follows

\[
W_{av} = \frac{1}{n} \sum_{r=0}^{n} |M_{nf} - N_r| \text{ where } 0 < N_r < 1 \quad ... (3)
\]

In equation (3), \( n \) is the number of neighbors \( N_r \) whose value lies in the range zero and one.

The window average calculated above is used to detect edges by checking it against a threshold value. If the average value is above the threshold \( (\delta) \) this implies that the pixel belongs to edge otherwise it is a non-edge point. So, finally the resultant image is obtained as

\[
y_{ij} = \begin{cases} 
1 & \text{if } W_{av} > \delta \\
0 & \text{otherwise} 
\end{cases} \quad ... (4)
\]

In equation (4), \( \delta \) is the adaptive threshold, with the value of threshold the level of edges can be controlled. The value of threshold is inversely proportional to the level of details that will be detected. The image comprised of the pixels \( y_{ij} \) shows the edges of the input image. The proposed algorithm is shown below.

1. Input noisy image.
2. Take a sliding window \( W_M \) centered at \( x_{ij} \)
3. If \( x_{ij} = 0 \) or \( x_{ij} = 1 \) calculate
   \[ W_{av} = \frac{1}{n} \sum_{r=0}^{n} |x_{ij} - N_r| \text{ where } 0 < N_r < 1 \]
4. If \( x_{ij} = 0 \) or \( x_{ij} = 1 \) calculate
   \[ W_{av} = \frac{1}{n} \sum_{r=0}^{n} |M_{nf} - N_r| \text{ where } 0 < N_r < 1 \]
5. Select a threshold \( \delta \).
6. If \( W_{av} > \delta \), then the output pixel \( y_{ij} = 1 \).
7. If \( W_{av} \leq \delta \), then the output pixel \( y_{ij} = 0 \).

### 4. EXPERIMENTAL RESULTS

The proposed algorithm was compared with several existing edge detectors to evaluate its performance. The images have been contaminated with salt and pepper noise. The results were compared on the basis of the efficiency with which the true edges are detected. Standard Cameraman grey image was taken as the test image as shown in figure 1. The proposed algorithm (PA) with the value of \( \delta = 0.03 \) was compared with Sobel operator (SO) and Canny edge detector (CD). Noise of different densities was added ranging from 10% to 50%. It can be seen in figure 2(b) and (c) that with 5% noise sobel operator and canny edge detector detects noise also as edges while it can be seen that the proposed algorithm finds accurate edges as seen in figure 2(c). With the noise percentage increases to 30% and 50% with Sobel and canny edge detector the edges cannot be identified at all while the proposed algorithm works well even with such high noise. It can be seen that with the proposed algorithm all the edges are detected like the fingers of the cameraman and the building behind is also detected which remains undetected with the other filters.

![Fig 1: Test image cameraman](image1)

![Fig 2: Edge detection using different filters](image2)
5. CONCLUSION

An Adaptive Edge Detection Algorithm for Images Corrupted with Impulse Noise is proposed in this paper. The algorithm uses a threshold value to detect the edges of an image. The proposed algorithm finds the average of the differences of all those pixels whose value lies between zero and one with the center pixel and if the center pixel itself is zero or one then the difference from the adaptive median is calculated. The average is compared with the threshold to find the edges. The results also show that the proposed algorithm works better than the other algorithms.

6. REFERENCES


