Improve Edge Detection Technique using Adaptive Filter

Che Ming Chia
Department of Electrical Computer Engineer
University of California, San Diego
9500 Gilman Dr, La Jolla, CA
chchia@ucsd.edu

Abstract

Edges are one of the important feature in biomedical imaging for the problem such as hair crack detection and tissue damage detection. General noise reduction method such as Gaussian blur removes noise but during the noise removal process it also deform the edges that are important in recognition task. In this paper an improvement in adaptive filtering has been proposed that preserve more edges without much lose in Peak signal to noise ratio(PSNR). The algorithm cleans up noise in homogeneous background but preserve thin edges in the objects.

1. Introduction

Edge detection is one of the most common used technique in image analysis. It is also a critical element in image processing since edge contains lots of useful information. An edge can be describe as as a boundary between two regions separated by distinction in strong intensity values of the pixels. Edge detection can be define as a process of finding and tracing sharp discontinuities in the image. It has been widely used in object recognition, feature extraction, and image analysis. For example, most features extracted by algorithms are based on corners which have high correlation with edges. There are several kinds of edge detection methods such as Sobel [8], Perwitt [7], and the most popular one – Canny detector [1]. However, all of the among detectors have encountered artifacts that caused by noise. Therefore, noise reduction becomes an important issue in edge detection.

2. Related Work

In day to day edges in digital images have be an vital element in medical diagnosis. Much prior works relate to edge detection have been proposed to get better edges. However, the noise in the digital image has become an fatal concern since it can greatly affect the edge detection performance. Therefore, some methods have been provided to perform noise reduction such as Gaussian smooth filtering, Mean filter, and Median filtering. Although all of the above methods can efficiently reduce the noise, they lost valuable information in the edge which can lead the Doctor to make false diagnosis. P.Heinonen et al. [5] proposed an algorithm to reduce noise instead of blurring the edge by using median filter. However this algorithm rely on the distribution of neighbor pixels. No matter how people improve the median filter, there is a possibility that the algorithm will preserve impulse noise. Some other useful technique is by using wavelet transform [2] and sparse analysis regularization(SAR)[9]. This paper proposed an simple kernel-based solution that adapt to the image through iterations. Remove the noise and reduce the color levels in the same time which increase the the performance of gradient-based edge detection algorithms.

3. Dataset

In this paper, we performed our algorithm on Color BSD68 dataset(CBSD68). This benchmark dataset is widely used for measuring image denoising algorithms performance. It includes the original .jpg files, converted to lossless .png, and noisy with Additive White Gaussian Noise of different levels [3].

4. Proposed Algorithm (Description of Project)

The proposed method is a variant of the Sobel edge detection technique. In Sobel's method[8] the first step is to smooth the image to eliminate the noise in the image. For this task Gaussian smoothing filter is adopted which does not preserves the edges. So we adopt Adaptive filter which preserves the edges of the objects while smoothing the image.

4.1. Noise Type

In this paper, we evaluate out methods with Gaussian White Noise (GWN) on several level since it is the most common one.

White Gaussian noise, in which the values at any pair of times are identically distributed and statistically independent. Principal sources of Gaussian noise in digital images arise during acquisition. For example, sensor noise caused by poor illumination and high temperature.

4.2. Noise Reduction

The Adaptive smoothing is a class of typical nonlinear smoothing technique. The edge preserve smoothing algorithm is applied independently to every image pixel using different coefficients. To calculate the coefficients of the convolution mask for every pixels, Manhattan color distances d_i , i = 1,.,.,.,8 are extracted between the central pixel and the eight neighboring pixels in a 3x3 sliding window, which are normalized in the range [0,1].

That is,

$$d_i = \frac{|R_c - R_i| + |G_c - G_i| + |B_c - B_i|}{3}, d_i = [0, 1] \quad (1)$$

where R_c, G_c, B_c is the central pixel value in the current sliding window.

After calculating the color distance between its neighborhoods, we can evaluate the weights based on these values(kernel coefficients). The following equation is used:

$$w_i = (1 - d_i)^p, where \quad p \geq 1 \quad (2)$$

In words, w_i receives larger values for smaller color distance so pixels having small color distance from the central pixel receive large weights. This concludes to the following convolution mask:

$$\frac{1}{\sum_{i=1}^{8} w_i} \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & 0 & w_5 \\ w_6 & w_7 & w_8 \end{bmatrix} \quad (3)$$

The filtering of the image is achieved by applying the above convolution mask on RGB channels respectively. Factor p in equation(2) scales exponentially the color distance which means that it controls the blur effect on the edges.[4][6]

The next step is filter the image iteratively. In each iteration, the image is filtered by a convolution mask with different coefficients which means the filter process is based on image itself. That's why we called it adaptive filtering.

When the adaptive filter is applied to boundary pixels of an object, the color distances d_i take large values for the neighboring pixels that do not belong to the object, and hence the colors of these neighboring pixels have a small weights on the final color received by the central pixel. For this reason, the proposed adaptive filter can be considered as edge preserving filter(Figure 1). In other words, you can also consider the convolution kernel as a process of checking whether the central pixel belongs to its neighboring pixels.

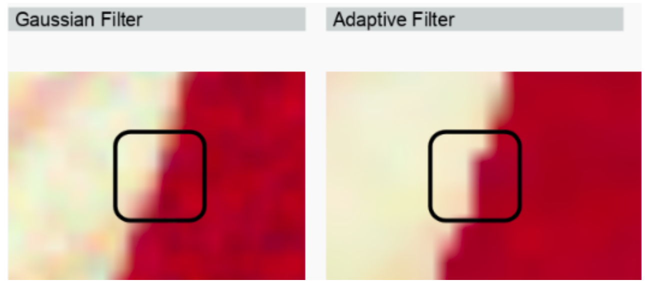

Figure 1. Show how the Edge Preserve Smoothing Filter (EPSF) works.

4.3. p-degree Parameter Tuning

p plays an important role in this algorithm since it controls how blurry should the edge becomes. If p = 1, the filter is approximate to the box kernel which means that the blurry effect is very large and the information on edges can be ignored. As p gets larger, coefficients with small color distance from the central pixel increase their relative value difference from coefficients with large color distance, so the blurring effect decreases. A fixed value p = 10 is used for all of our experiments because this resulted in very good performance. The central pixel of the convolution mask is set to zero to remove impulsive noise.

4.4. Number of Iterations Selection

Numbers of iteration is another critical factor. It reduces number of color levels in the image. As the it gets larger, the clear edge will be enhanced but implicit edge(subtle features) in the object will be ignored Figure 3. Therefore, the more the number of iteration does not means to get better SNR (Figure 4). To solve this problem, we terminate the


Figure 2. Image for evaluating the effect of iterations

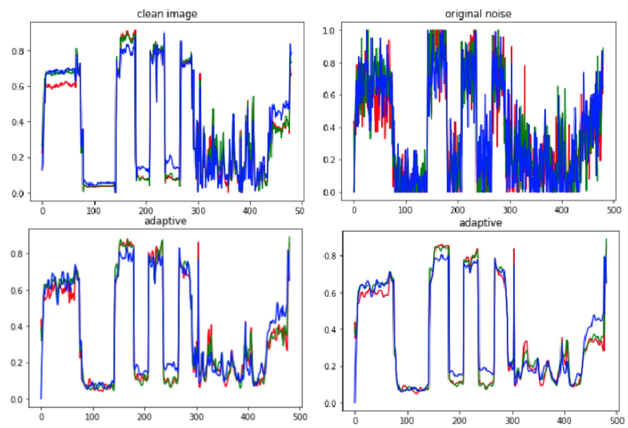

Figure 3. Left-top image is variation in row = 150 in clean Figure 2. Right-top image is variation in row = 150 in noisy Figure 2. Left-bottom image is variation in row = 150 in noisy Figure 2 after doing 3 times iteration(**Target image**). Right-bottom image is variation in row = 150 in noisy Figure 2 after doing 10 times iteration.

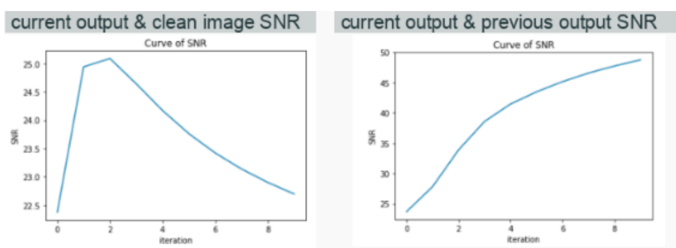

Figure 4. Left image is the trend of SNR through each iteration. Right image is the SNR between current output and previous output which represent the variation in current iteration

iteration when the variation between two iterations are sufficient small. Although we cannot use gradient descent to calculate loss function since we do not include clean image in our algorithm, we can still compare the SNR between current output and previous to see whether the variation is small enough.

4.5. Edge Detection

Compute the gradient magnitude and direction of the edges in the image: After removing noise, we have to find the edge strength and direction by taking the gradient of the image. This is done by the Sobel operator by applying on the image. Then, the approximate absolute gradient magnitude at each pixel is found. This can be calculated by using the formula:

$$G = Gx + Gy \tag{4}$$

where G denotes the gradient of pixel
Gx denotes the gradient in x-direciton
Gy denote th gradient in y-direction

The Sobel gradient kernel is as follow:

$$S_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$S_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

5. Experiments and Results

Some of the results can be seen on Figure 5, Figure 6. It can clearly observe that the noise in the homogeneous part has been suppress but preserve some of subtle features inside the object. The thickness of edges obtain by using Gaussian smooth filtering noise reduction is proportional to the standard deviation given by the user. The higher std. make the noise reduction performance better but greatly decrease the edge preserve performance simultaneously. Thus, as shown in our results, the thickness of edges has been greatly improved which means it is precisely positioned.(Figure 7)

6. Conclusion

Hence from the results it is clear that by using the Adaptive filter for edge detection yields better result than the pure sobel edge detection which uses Gaussian filter for smoothing. For future improvement, we still need to find a robust method to predict the best number of iteration which implicitly finding the best SNR. In this paper, we only observe that there is a trend in the iteration and try to extract the one that has better SNR. However, we cannot precisely select the exact number of iteration to get output with best SNR.

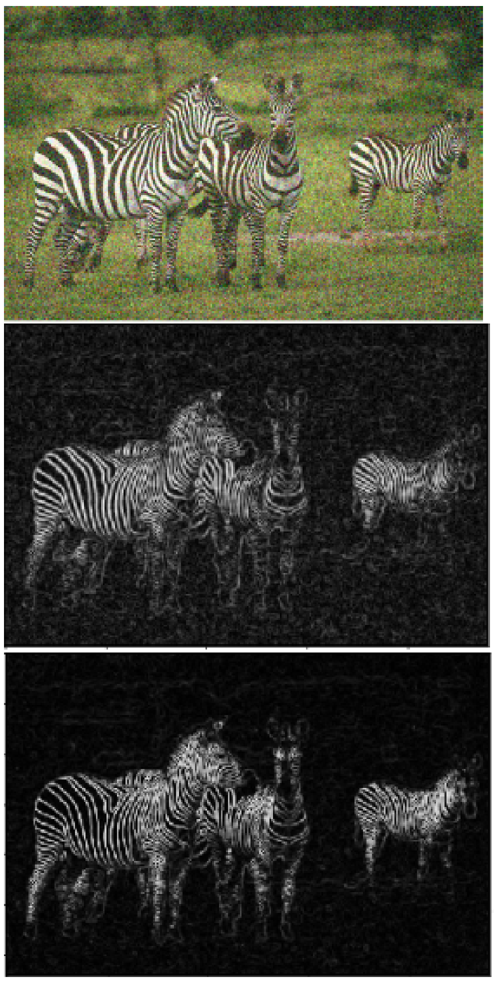

Figure 5. From top to bottom is the original noisy image, edge detection with Gassian smooth, and edge detection with adaptive filtering

Figure 6. From top to bottom is the original noisy image, edge detection with Gassian smooth, and edge detection with adaptive filtering

Figure 7. Left image is the Sobel edge detection using Gaussian smooth filtering with large standard deviation. Right image is the results of proposed algorithm.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[2] D. L. Donoho. De-noising by soft thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[3] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proc. 8th Int'l Conf. Computer Vision*, 2:416–423.

[4] N. Nikolaou and N. Papamarkos. Color reduction for complex document images. *Wiley Periodicals*, 2009.

[5] P.HeinonenI, DefeA.Nieminen, and Y.Neuvo. Median based algorithms for image enhancement and edge detection. *IFAC*, 19:33–39, 1986.

[6] Praneeth.CH, S. R. .V, and Srinivas.K. Image edge detection using adaptive filter. *IJCSIT*, 2(4):1581–1583, 2011.

[7] Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 1970.

[8] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. *Stanford Artificial Intelligence Project (SAIL)*, 1968.

[9] S. Vaiter, G. Peyre, C. Dossal, and J. Fadili. Robust sparse analysis regularization. *arXiv:1109.6222v5*, 2012.