International Journal of Advancements in Research & Technology, Volume 2, Issue 4, April-2013
ISSN 2278 - 7763

A Quantitative Approach for Textural Image Segmentation with Median Filter

Dr. D. Pugazhenthi [1] , Priya K [2]

[1]Department of Computer Science, Quaid-E-Millath Govt. College for Women, Chennai, India
Email: pugazh006@gmail.com
[2]Research Scholar, Department of Computer Applications, St. Peter's University, Chennai, India
Email: kpriyaa@gmail.com

ABSTRACT
Retrieving images from large and varied collections using image content as a key is a challenging and important problem. The success of the approach depends on the definition of a comprehensive set of goals for the computation of edge points. In this paper an approach is proposed which segment the image by separating the texture. The proposed method, nonlinear filters, removes only corrupted pixel by the median value or by its neighboring pixel value. As a result, the proposed method removes the noise effectively even at noise level as high and preserves the edges.

Keywords - 2D Median filter, Texture segmentation, Noise removal, Gray scale.

1 INTRODUCTION

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. In a (8-bit) gray scale image each picture element has an assigned intensity that ranges from 0 to 255. Image segmentation often serves as a crucial initial step before performing high-level tasks such as object recognition[1]. The texture has different intensities in the same region while it has the similar behavior. For the segmentation of intensity images, there are four main approaches [2] namely, threshold techniques, boundary based methods. Threshold techniques are based on the postulate that all pixels whose value (gray level, color value) lie within a certain range. The proposed algorithm in this paper also greatly focuses on how to effectively detect the salt and pepper noise and efficiently restore the image. Conventional median filtering approaches apply the median operation to each pixel unconditionally whether it is uncorrupted or corrupted. As a result, even the uncorrupted pixels are filtered and this causes degradation of image quality.

2 RELATED WORK

2.1 TEXTURE

Texture is a well-researched property of image regions, and many texture descriptors have been proposed, including multi-orientation filter banks [3] and the second-moment matrix [4]. In this paper, elaborate on the classical approaches to texture segmentation and classification, both of which are challenging and well-studied tasks. Rather, we introduce a new perspective related to texture descriptors and texture grouping motivated by the content-based retrieval task.

International Journal of Advancements in Research & Technology, Volume 2, Issue 4, April-2013 2
ISSN 2278 - 7763

2.2 TEXTURE SEGMENTATION

The goal of texture segmentation is to partition an image into homogeneous regions and identify the boundaries which separate regions of different textures. Segmentation is obtained either by considering a gradient in the texture feature space[5] or by unsupervised clustering[6] or by texture classification[7]. Segmentation by labeling often suffers from a poor localization performance because of the conflicting requirements of region labeling and boundary localization in terms of observation window. Unsupervised clustering/segmentation requires an initial estimate of the number of the regions in the image, which is obtained mostly by setting a threshold in the feature clustering algorithm.

2.3 MEDIAN FILTER

The median filter is a nonlinear digital filtering technique, often used to remove noise. The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. Consider median filter with three point window of W={-1, 0, 1} and the inputs are [x(n-1), x(n), x(n+1) = {-3,-6,8}. To formulate the function[10]

$$F(\theta) = \sum_{k=-1}^{1} | \theta - x_{n+k} |$$

3 PROPOSED ALGORITHM

The wavelet decomposition using Gabor functions has an important physical interpretation. The complex Gabor function has an even-symmetric (cosine) real part and an odd-symmetric (sine) imaginary part, which respond maximally to line edges (or bars) and step edges (of appropriate sizes and orientations), respectively, in the image. This wavelet decomposition can be viewed as obtaining a primal sketch of the raw intensity data by detecting perceptually significant features at different scales. These features can be detected at the local maxima in their energy[9]. If R_i and I_i represent the response from the even and odd symmetric feature detectors at a position i, then the local energy E_i at i is given by

$$E_i = \sqrt{R_i^2 + I_i^2}$$

This algorithm is proposed for gray level images, any color image can also be segmented by converting to gray level image. Gray level images can also have some noise or much difference in same texture. For example the texture of soil can have stones or some other types of sand that makes difference in pixel intensities and approaches to be a new texture. It makes necessary to remove these noise and ignore to be a different texture. These factors make difficult the segmentation task. So we used 2D median filtering. Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. 2D Median filtering [8] is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges. Median filtering is performed on the block of the matrix A in two dimensions (M*N). Each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image. The median filter reduces the variance of the intensities in the image. It preserves certain edge shapes. This algorithm gives minimum complexity as O (M*N), where input image size is M*N. This is minimum complexity to process any image.

International Journal of Advancements in Research & Technology, Volume 2, Issue 4, April-2013
ISSN 2278 - 7763

STEPS:

Step 1: Select 2-D window of size 3 x 3. Assume that the pixel being processed is E_{ij}.

Step 2: If $0 < E_{ij} < 255$then E_{ij} is an uncorrupted pixel and its value is left unchanged.

Step 3: If $E_{ij} = 0$ or $E_{ij} = 255$ then E_{ij} is a corrupted pixel then two cases are possible

Case i): If the selected window contain all the elements as 0's and 255's. Then replace E_{ij} with the mean of the element of window.

Case ii): If the selected window contains not all elements as 0's and 255's. Then eliminate 255's and 0's and find the median value of the remaining elements. Replace E_{ij} with the median value.

Step 4: Repeat steps 1 to 3 until all the pixels in the entire image are processed.

Each and every pixel of the image is checked for the presence of salt and pepper noise. If the selected window contains salt or pepper noise as processing pixel (i.e., 255/0 pixel value) and neighboring pixel values contains some pixels that adds salt (i.e., 255 pixel value) and pepper noise to the image.

78	**90**	**0**
120	**0**	**255**
95	**255**	**73**

where "0" is processing pixel. Now eliminate the salt and pepper noise from the selected window. That is, elimination of 0's and 255's. The 1-D array of the above matrix is [78 90 0 120 0 255 97 255 73]. After elimination of 0's and 255's the pixel values in the selected window will be [78 90 120 97 73]. Here the median value is 90. Hence replace the processing pixel by 90.

The threshold value is dependent on the image and the noise density. So, to restore different images we need to check for a range of threshold values and find out the best one. The threshold parameter is calculated using the detailed coefficient matrix. The detailed coefficient d(i,j) is calculated by calculating the absolute difference between the current pixel h(i,j) value and the mean of good pixels (around the current pixel) p(i,j). Now the algorithm intelligently decides that the threshold is

$$\mathbf{t = min(d(t,j))}$$

4 RESULT AND DISCUSSIONS

This proposed algorithm is implemented in MATLAB and a image is taken to perform this algorithm (Figure 1). The performance of the proposed algorithm is tested with different gray scale images. We expect to give better results for all the textured images while using the flexibilities as discussed. The usefulness of the proposed scheme for segmentation lies in the fact that very little parameter tuning or selection is needed. The parameters controlling segmentation are the preferred image scale and the approximate number of regions. The experimental results indicate that the proposed approach results in visually acceptable segmentation on this diverse image collection (Figure 2). For qualitative analysis, performances of the filters are tested at different levels of noise densities.

ISSN 2278 - 7763

Figure 1: Input Images

Figure 2 : Segmented Images

5 CONCLUSION

This paper represents a texture segmentation algorithm in term of object detection. This algorithm removes the texture part from the input image and shows the object part with some hard boundary if exists. This proposed algorithm is very much helpful to detect the obstacle in any image and provide very much efficiency and minimum complexity.
This algorithm is not able to suggest the coordinate of the obstacle appearing in the image and fails to describe the proper clarity of the object as well as number of object. Segmented results of the image are shown in binary image only. This paper uses a small 3 x 3 window having only neighbors of the corrupted pixel that have higher correlation; this provides more edge details, leading to better edge preservation. The proposed filter also shows consistent and stable performance across a wide range of noise densities. Computation time is also reduced. Effective noise removal can be observed even up to maximum noise density level.

6 ACKNOWLEDGMENT

It is with immense gratitude that I acknowledge the support and help of my Professor. I sincerely thank my parents for the great support. I wish to thank all my friends, who are supported for the outcome of the work.

7 REFERENCES

[1] W.Y.Ma and B.S. Majunath, “Edge flow: a framework of boundary detection and image segmentation”, Proc. of CVPR, 1997, pp744-749.
[2] R.M. Haralick and L.G. Shapiro, “Image segmentation techniques,” Comput. Vis. Graph. Image Process., vol.29, pp. 100-132, 1985.
[3] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. J. Opt. Soc. Am. A, 7(5):923-932,1990.
[4] W. Forstner. A framework for low level feature extraction. In *Proc. Eur Conj Comp.* Vis., pages 383-394,1994.
[5] B.S. Manjunath and R. Chellappa, “A Unified approach to boundary detection,” IEEE Trans. Neural Networks, Vol.4, No.1, pp.96-108, Jan. 1993.
[6] D. Dunn, W.E. Higgins, and J.Wakeley, “Texture segmentation using 2-D Gabor elementary functions,” IEEE Trans. Pattern Anal. And Machine Intell., vol. 16, pp. 130-149, Feb. 1994.
[7] J. Mao and A.K. Jain, “Texture classification and segmentation using multiresolution simultaneous autoregressive

International Journal of Advancements in Research & Technology, Volume 2, Issue 4, April-2013
ISSN 2278 - 7763

models," Pattern Recognition, Vol.25, No.2, pp.173-188, 1992.
[8] Lim, Jae S., Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ, Prentice Hall, 1990, pp. 469-476.
[9] M.C. Morrone and D.C. Burr, "Feature detection in human vision: a phase dependent energy model," in Proc. Royal Society of London (B), vol. 235, 1988, pp. 221-245.
[10] C. A. Bouman: Digital Image Processing - January 9, 2012.

