DE-NOISING OF 24 BIT TRUE COLOR IMAGES USING ROTATING FILTERS

K V RAMPRASAD 1*

1. *Professor, Dept of ECE, KALLAM HARANADHA REDDY INSTITUTE OF TECHNOLOGY, GUNTUR, AP, INDIA.*
Email: kvrp1976@gmail.com

ABSTARCT:

Most of the images acquired from imaging sensors, need some amount of pre-processing before they can be compressed and archived. The pre-processing step usually involves smoothing of image data, noise cleaning, contrast enhancement and other related steps. In medical imaging, the primary criterion for a good image is the isolation of features that the investigating doctor desires. In pattern recognition systems and assembly robots, the region of interest is well-defined contours and edges based on which pattern matching algorithms can be applied. Some of the most notable milestones in the area of image processing include Computerized Axial Tomography, Emission Tomography, seismic activity studies and geophysical prospecting[1].

The objective of this paper is to remove the noise present in 24 bit true color images using rotating filters.

Keywords: **DATA, COMPRESSION, ROTATING FILTERS**

I. INTRODUCTION

Image Processing is a heavily used process in many applications such as entertainment industry, medical imaging, satellite imaging, and material testing in industries. Radars tracking systems, astronomy etc. No imaging system is ever perfect. Undesirable qualities creep into the image due to various causes such as limitations of the imaging system in terms of response to incident light, noise introduced into image sampling due to internal and external causes, atmospheric conditions, motion of the object or the imaging sensor during image acquisition, out-of focus conditions and latency of the imaging hardware. It is therefore imperative that some amount of processing be done on the acquired image so that the image possesses the desired properties. The desired properties in an image depend on the type of application at hand. In images relating to people and natural scenes, the desired qualities are well focused appearance, good contrast and clarity of details. Image processing is an extremely vast and complex domain and a considerable amount of research and development has taken place in the past two decades. In spite of the vast amount of developments in the recent past, there are still a large number of application areas that have not been exploited till now. There is a tremendous amount of opportunity and scope to discover newer and more efficient algorithms[2].

II. RELATED WORK

For removal of salt and pepper or random noise, generally averaging filters of 3 X 3, 5 X 5, 7 X 7 or median filters are suitable. But, they tend to blur the edges when the averaging operation takes place near the edge boundaries. Hence, we should go for rotating filters, in which averaging of pixel with neighboring pixels is avoided, instead of that another method, which is described below[3].

Here, one more statistical/ stochastic distribution filter based on a rotating grid is explained. This rotating filter, in some situations performs better and doesn't require the operation of extracting the edges and adding them to the smoothened image.

The filter is implemented over a 5x5 grid with a 3x3 sub-grid within. Assuming the pixel of interest to be centered in the 5x5 grid, there are 9 possible 3x3 sub-grids containing the pixel of interest. This is depicted in the diagram below:

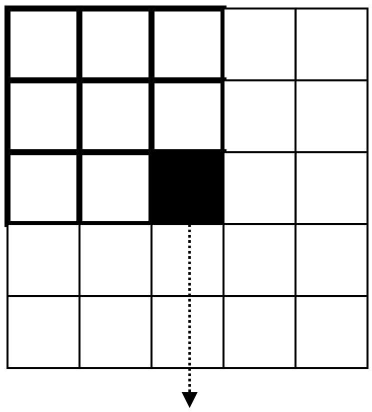


Fig. 1: Grid Structure

The 5x5 grid above is moved across the image left to right one pixel at a time and down by one line in order to cover the entire area. For each position of the grid, 9 sub-grids of size 3x3 are extracted and the dispersion coefficient within the 3x3 grids is calculated[4,5]. Thus we end up with 9 dispersion coefficients for each position of the 5x5 grid. The 3x3 grid with the least dispersion coefficient is chosen and the averaged value of this grid is substituted at the central point (x, y). The dispersion coefficient for a 3x3 sub-grid is defined as:

$$\frac{1}{9}\left[\sum_{x=0}^{2}\sum_{y=0}^{2}(p_{xy})^2 - \frac{\left(\sum_{x=0}^{2}\sum_{y=0}^{2}p_{xy}\right)^2}{9}\right]$$

Fig. 2: Formula for calculating dispersion coefficient

where p_{xy} is the pixel at (x, y). It is possible to possible to have multiple runs of the mask across the image. The process reaches a steady state when the value used to update the central pixel no longer changes. The following represent the result of applying the above dispersion filter:[6,9].

III. COLOR SPACE FUNDAMENTALS

From the color cube model[4], it is evident that the grayscale values along the diagonal of the cube connecting (0,0,0) and (1,1,1). The values of red, blue and green are equal along this axis. Therefore, equal values of red, blue and green should produce a grayscale. While this scheme does produce a grayscale, minor adjustments to red, green and blue based on certain standards will produce better grayscale images – images well balanced in grayscale intensities.

A color image consists out of pixels with red, green and blue (RGB) color variations. E.g. the color value of a pixel can be one out of 256 (= 2^8) variations of red, 256 variations of green and 256 variations of blue. So in this example the color value of 1 pixel can be stored in 3 x 8 bits = 24 bits.

In most programming languages it is possible to read the color value of a pixel as one number. This number could be in a *Long*, or a *Hex* notation. In case the color value is a *long*, it has a value between 0 and 256^3 (=16777216). If it is in the *hex* notation, you can have a value between 000000 and FFFFFF.

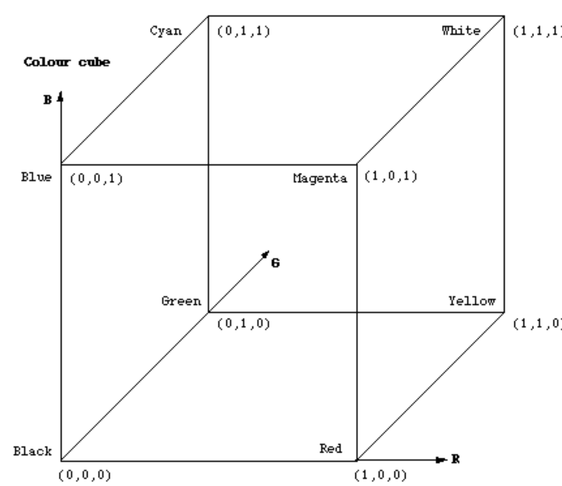


Fig 2: Color Cube Model

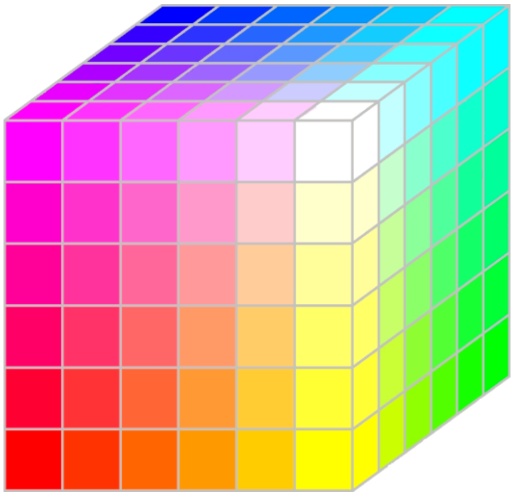


Fig 3: RGB Color Cube

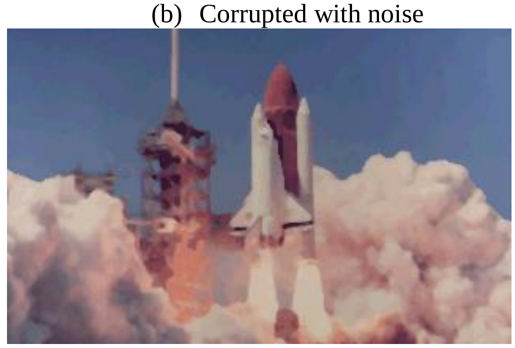
The figure shown represents a color cube in a 3-D rectangular Cartesian coordinate system. The origin (0, 0, 0) represents black color. The coordinate (1, 1, 1) denotes white color. The diagonal connecting (0, 0, 0) and (1, 1, 1) represents the grayscale line. All the points on this diagonal have equal values for x, y, z and depict a grayscale lying somewhere between black and white. The basic rule is that equal percentages of red, blue and green produce a grayscale that lies between black and white.

IV. RESULTS & DISCUSSIONS

In the pre-processing stage, the noise in the images is eliminated using rotating filters and the results are compared with median filters.

(a) Original Image

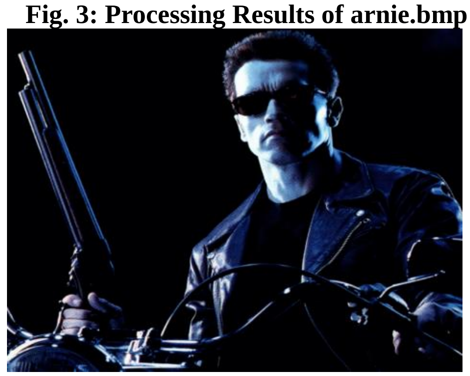
(b) Corrupted with noise



(c) Smoothened with Median Filter

(c) Smoothened with Rotating Filter

Fig. 3: Processing Results of arnie.bmp



(a) Original Image


(b) Corrupted with noise


(c) Smoothened with Median Filter


(d) Smoothened with Rotating Filter

Fig. 4: Processing Results of arnie.bmp

V. CONCLUSIONS

In the processing stage, the averaging operations on pixel data, some times tend to blur the edges, which are nothing but the sharp transitions in the image, so it is observed that the non-linear filters like median and rotating filters are more appropriate and the edges need not be extracted separately and preserved.

VI. REFERENCES

[1]Abramson A., Introduction Theory and Coding, McGraw-Hill, New York, 1963.
[2]Antomini M., "Image Coding Using Wavelet Transform" IEEE Trans. On Image Proc. 1: 205-220, 1992
[3]Elias P.M., (1994). A new data structure for cumulative probability tables. Software-Practice and Experience, 24(3): 327-336. Errata published in 24(7): 667, July 1994.
[4])Jacquin A., "Image Coding Based On A Fractal Theory of Integrated Constructive Image Transformation" IEEE Trans. On Image Proc., 1: 18-30,1992
[5]Mandelbrot B.B., The Fractal Geometry of The Nature, Freeman, New York, 1982.
[6] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: Theory and design," IEEE Trans. Signal Proc., vol. 40, no. 4, pp. 882–893, April 1992.
[7] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," IEEE Trans. Commun., vol. 31, no. 4, pp. 532–540, April 1983.
[8] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," IEEE Trans. Image Process., vol. 14, no. 12, pp. 2091–2106, Dec. 2005.
[9] A. L. Da Cunha, J. Zhou, and M. N. Do, "The Nonsubsampled Contourlet Transform: Theory, Design, and Applications," IEEE Trans. Image Process., vol. 15, no. 10, pp.3089–3101, Oct. 2006.
[10] Contourlet based image watermarking using optimum detector in the noisy environ ment S.M.E. Sahraeian, M.A. Akhaee, S.A. Hejazi, F. Marvasti