

Edge Enhancement and Its Applications

Kanika Gupta¹, Harjeet Singh²

^{1,2}(Electronics & Communication Engineering, Punjab Technical University, India)

Abstract: We describe an automated edge enhancement procedure that operates in conjunction with clustered-dot dithering, edge weighted unsharp masking, compound anisotropic diffusion etc. The goal is to make weak edges more noticeable and to give the strong edges in the half toned image a sharper and more natural appearance. Unsharp masking that is widely used for image enhancement amplifies image contrast by adding the high frequency component that is obtained by a linear high pass filter from an input image. To preserve image features, the anisotropic diffusion is governed by a local co-ordinate transformation. Experimental results show that edge enhancement is successful in improving the appearance of the text, line art and natural scenes. Also real ultrasonic images indicate effective preservation of edges and local details. Edge enhancement enhances edges in image with natural transition and sharp visual quality.

Keywords – Anisotropic diffusion, augmented Vision, Dyadic wavelet transform, Image edge enhancement, spatial phase filter

I. Introduction

Clustered-dot half toning is the default rendering mode or an available mode on most modern laser and ink-jet printers. However, clustered-dot techniques have a well known disadvantage in that they do not reproduce sharp edges in the image. Our intention is not to make weak edge areas more noticeable, but rather to give the strong edges a sharper and more natural appearance. One of the challenges we face is that clustered-dot dithering produces a regular and periodic dot pattern, and the eye is extremely sensitive to any change in that pattern. If we begin to modify the dots carelessly, we will end up with an “enhanced” image that looks worse than the one we started with.

We introduce a wavelet transform to the edge weighted unsharp masking to improve the noise suppression property. We examine the edge enhancement property and the noise amplification property of the proposed method. By this examination, we propose a design method for the weighting function. The weighting function is optimized to minimize the noise amplification while maintaining the desired edge property.

In this paper, we find a sufficient condition for constructing spline dyadic wavelets based on box-spline functions, where the fourier transforms of scaling and wavelet functions and their dual functions play an important role. Under the sufficient condition, we design several spline dyadic wavelet filters using box-spline functions. In simulation, we apply them to enhancement and edge detection of some medical images, which is required by medical doctors to diagnose illness.

In this paper, we adopted the generalised radial Hilbert transform theory to design spatial phase filters for edge-directed enhancement. The generalised Radial Hilbert transform (GRHLT), which is the generalisation of the separable 2-D HLT and the RHLT. The most important two advantages are noise immunity and that ramp edges can be detected successfully.

People with visual impairment are unable to be mobile without some form of assistance. In a recent large scale survey of providers of low vision services, we found that for elderly patients with low vision, they need an assessment while watching television, mobility and independent travel, and hobbies. In this paper, we used an advanced edge enhancing approach for those of low vision patients, which enhances only the major boundaries while keeping irrelevant ones intact.

II. Clustered-Dot Dithering

Any edge enhancement algorithm must first find the edges, and then manipulate the nearby pixels in a way that improves the appearance of the image. The following sections briefly describe how our algorithm accomplishes these two tasks.

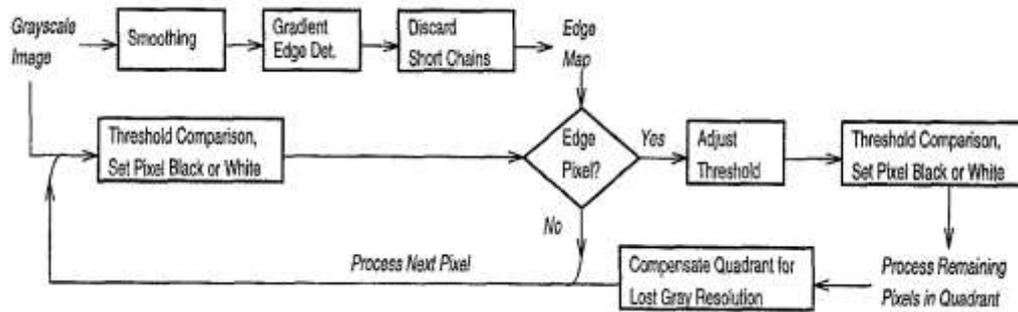


Figure 1: Block diagram of edge enhancement procedure

1. Generating the edge map

We have found that for most images, the edge maps generated by the Sobel algorithm are unsatisfactory for use in our edge enhancement scheme unless some additional processing is done. The trouble is due to the difficulty of finding a “good” threshold

Value T that simultaneously preserves true edges and rejects false edges caused by quantization noise. Since it is easier to remove noise from the edge map than it is to fill in missing edge information, we have solved the thresholding problem by setting T to relatively low values and then cleaning up the resulting edge map in a separate post-processing step: we keep only the longer contiguous chains of edge pixels, and treat all others as non-edge pixels. Effectively, we introduce another threshold, γ , which specifies the smallest number of edge pixels in a contiguous chain such that the chain will be accepted as an edge. An additional processing step—pre-smoothing the grayscale image—further improves the edge map for our purposes. At first glance this may seem curious, since smoothing is in some sense the opposite of the edge enhancement that we have set out to accomplish. However, modest smoothing has little impact on the strong edges that we want to enhance, while it greatly reduces the undesirable effects of quantization noise. Many false edges are eliminated, as are weak edges that would not be well suited for enhancement anyway.

II. Edge Enhancement Algorithm

After the edge map has been computed, the edge enhancement algorithm proceeds according to a “selective two-pass” approach. On the first pass, the grayscale image is half toned using the clustered-dot threshold matrix in the usual way. A “selective” second pass occurs only in the vicinity of edges. The key elements of the second-pass processing are:

- local adjustment of the dithering threshold level to give finer spatial resolution near edges in the image
- tone compensation in the excessively bright and dark areas that result from using the modified threshold values.

2.1. Adjustments to the Dithering Threshold

The dithering threshold level is modified on a pixel-by-pixel basis for three different types of pixels near edges in the image. A Type 1 (T1) pixel is an edge pixel with at least six neighboring edge pixels. A Type 2 (T2) pixel is any edge pixel that does not qualify as a T1 pixel. A Type 3 (T3) pixel is one that is not identified as an edge pixel in the edge map, but has at least six neighbors that are edge pixels. A threshold adjustment is made at all T1 pixel sites, but at only some of the T2 and T3 sites. In all cases, the adjustment seeks to move the threshold level so that it lies between the original grayscale intensity values on either side of the edge. This ensures sufficient contrast at the edge locations in the resulting halftone. The goal is to trade some gray-level resolution (in the half toned image) for better spatial resolution near edges. At T1 sites, the new dithering threshold is computed in the following manner. We search the edge map for the nearest non-edge pixels to the left, to the right, above, and below. The search continues for only a few pixels in each direction, so that only the local image context will affect the choice of threshold value. If non-edge pixels are found within that distance in all four directions, then the intensity values in the original grayscale image are sampled at the four non-edge sites; however, if the search for non-edge pixels fails in one or more directions, then the grayscale image is sampled at the greatest allowable distance in those directions. (We have used 8 or 16 pixels as the limit.) Next, the new threshold value is computed as an average of the grayscale samples. If exactly two of the sample values are greater than 127 and two are less than 127, we consider the samples to be “balanced” and simply set the threshold equal to the mean of the four values. In the “unbalanced” case, we instead use the average of the maximum and minimum sample values. If the computed average, p , is significantly less than 127, our experiments have shown that it is better to set the threshold to a higher value, between p and 127, rather than directly using the value p . The reason is that low thresholds often result in isolated clumps of white pixels, and the eye is more sensitive to small areas that are too bright than to small areas that are too dark. We have

obtained good results by setting the threshold to the value given by $p + 0.75(127 - p)$ when p is less than 127. At T2 and T3 pixel sites, an additional test is performed to determine whether or not any threshold adjustment should take place. Whereas T1 pixels are in the “interior of the edge,” where they are substantially surrounded by other edge pixels, T2 and T3 sites occur on the “edge of the edge,” where they interact with several non-edge pixels. Because of the eye’s sensitivity to any irregularity in the clustered-dot screen pattern, it is very important to avoid certain kinds of threshold modifications where the edge and non-edge regions meet. Careless modifications invariably lead to artifacts that are perhaps best described as “gaps,” LLblobs,” and “beating” (Moire) along the edge. For this reason, the test that determines whether modification is allowed involves measuring the distance from the T2/T3 site to a neighboring non-edge cluster. Note that no distinction is made between T2 and T3 sites for the purpose of threshold adjustment.

The test at T2/T3 sites proceeds as follows. By examining the pattern of other edge pixels adjacent to the T2/T3 site, we can usually approximate the direction normal to the edge. A look-up table with 256 entries can be constructed for this purpose. (There are

Next, the grayscale intensity is sampled at the location one pixel away from the T2/T3 site in the normal direction. This sample value is used to determine whether we are at the bright side or the dark side of the edge. Near the bright side, threshold modifications will tend to change black pixels into white ones, because for the most part, the grayscale values in the area will be greater than the computed threshold p . If such modifications occur at locations adjacent to an existing cluster of white, non-edge pixels in the halftone, the result will appear as an unnatural white blob or streak in the image. Therefore, the test is used to ensure that threshold modifications at the bright side of the edge occur only in close proximity to black, non-edge clusters, where the appearance will be that of changing the shapes of the existing clusters rather than creating a large spot of solid color. Similarly, on the dark side of the edge, the test allows more black pixels to be created only when a white, non-edge cluster is nearby. Note that in each case we are concerned with the distance to a “LLminoritycolor” cluster. The distance limit on both sides depends on the grayscale intensity as determined by the sample taken in the normal direction: if the sample pixel is very bright or very dark, threshold modifications are allowed at a greater distance from the minority cluster, because ordinary dithering produces clusters that are more widely separated in these areas anyway. A second look-up table with 256 entries may be used to convert the grayscale sample value to a distance limit (1-5 pixels in practice). We search in the approximate normal direction out to the maximum distance specified in the table. Only if a minority-color pixel is found does threshold modification take place at the T2/T3 site. At those sites where the normal direction is indeterminate, threshold modification is always allowed.

2.2. Compensation for Lost Gray Resolution

The threshold modifications described above will increase the number of white pixels near the bright side of the edge and increase the number of black pixels near the dark side. To some extent this is desirable, because, after all, we seek to enhance the edges. However, the loss of grayscale resolution can result in disturbing artifacts unless we take steps to rearrange the pixels in excessively bright and dark areas. Fortunately, the objectionable patterns tend to occur in predictable ways, allowing us to test and compensate for them using a relatively small set of rules. The rules we have developed are intended only as examples, because they have been tailored for a particular type of printer (the HP LaserJet 4M), a particular 8 x 8 clustered-dot matrix, and a particular printing resolution (600 dpi). However, it is very likely that with slight modifications, these rules would be useful with a variety of printers and dithering matrices. Although they are customized, our rules provide general insight for the two-fold problem of identifying and improving the “Lbad” pixel patterns that result from threshold modifications.

The compensation rules focus on three types of localized pixel patterns, taking into consideration the portions of the dithering matrix that overlay pixel sites where threshold modifications have taken place. Within a typical square threshold matrix, corresponding to a 45-degree screen angle, there are two “light” quadrants that tend to produce white clusters, and two “dark” quadrants that tend to produce black clusters. A black hole occurs in a light quadrant when threshold modifications create so many black pixels that the white cluster becomes invisible (largely because of toner/ink spreading). The black hole is eliminated by forcing several pixels near the center of the quadrant to be white. A white hole occurs when threshold modifications create too many white pixels in a dark quadrant. Because of the eye’s sensitivity to bright highlights, the rules for white holes are not simply color-reversed versions of those for black holes. White holes tend to appear at the corners of a dark quadrant; all four corners are checked, and if white holes are found, they are eliminated by flipping pixels to the black state in a pattern that depends on the number of affected corners. A lonely pixel results when threshold modifications leave a black or white pixel with no more than one neighbor of the same color. These isolated pixels usually are not rendered cleanly by the printer. Compensation consists of converting the lonely pixel to a 2 x 2 neighborhood of the same color.

III. Edge Enhancement Property Of Edge Weighted Unsharp Masking

To consider the relationship between the blurred edges that appear in the input and the weighting function, we define the ratio between the squared sums of the wavelet transforms at the different scales as:

$$r = \frac{\sqrt{(Wf_2^H)^2 + (Wf_2^V)^2}}{\sqrt{(Wf_1^H)^2 + (Wf_1^V)^2}} \tag{1}$$

and is used to indicate the degree of smoothness of an edge. Fig. 3 shows some examples of the squared wavelet transforms for one-dimensional edges that are blurred by the Gaussian filters of which kernels is at different variances. In this example, the wavelet transform at scale $j=2$ is derived with the low-pass filter $\{1/8, 3/8, 3/8, 1/8\}$ as the scaling function. The ratio r is equal to one for a step edge that is not blurred. The ratio increases along the variance of the Gaussian filter. In Fig. 2, the relationship between the variance of the Gaussian kernel of the filter and the ratio r is shown. The ratio r is proportional to the logarithmic function of the variance of the Gaussian kernel. Next, we represent the weighting function with the ratio r . Since the wavelet transforms at $j=1$ and $j=2$ are correlated around the image edges, we assume that the ratio between the vertical and horizontal wavelet transform is almost constant for $j=1, j=2$ as follow:

$$\frac{Wf_1^V}{Wf_1^H} \approx \frac{Wf_2^V}{Wf_2^H} \tag{2}$$

We get the weighting function with the ratio r as the form of:

$$e_x(m, n) \approx Wf_1^H (\alpha + \beta r) \tag{3}$$

$$e_y(m, n) \approx Wf_1^V (\alpha + \beta r) \tag{4}$$

The edges that appear in the input image under the enhancement process are blurred at any degree. The ratio r will be larger than one at the edge locations. To guarantee the enhancement effect for the blurred edges, Eqn. (3) and (4) have to be larger than zero for $r > 1$.

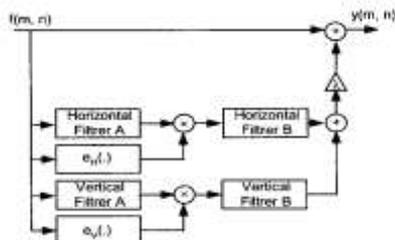


Figure 2: The proposed edge weighted unsharp masking

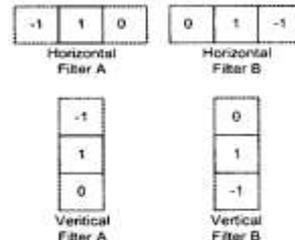


Figure 3: Four Filters used in Filter 2

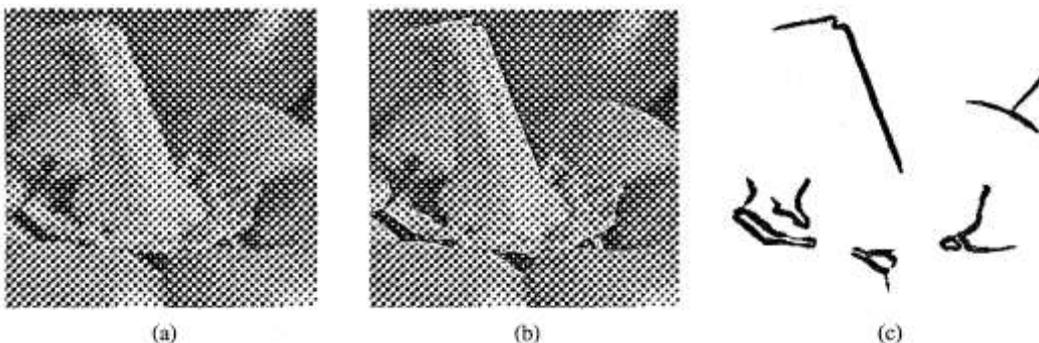


Figure 4: Central portion of the “peppers” image, printed at 150 dpi to show the dot structure. (a) Unmodified; (b) enhanced; (c) edge map used to generate enhanced image.

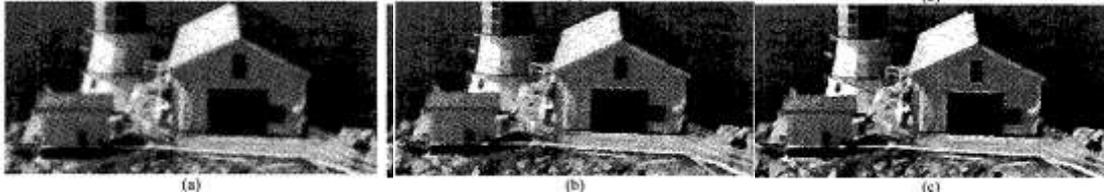


Figure 5(a) Input Image, (b) enhancement result by the cubic unsharp masking, (c) the proposed method

IV. Experimental Results

We have conducted experiments to test the operation of the edge enhancement algorithm. All of the work to date has been done using a LaserJet 4M printer and an 8 x 8 tone-corrected dithering matrix, with the output printed at 600 dpi. Before computing the edge map, we smoothed each test image using a uniform 5 x 5 convolution kernel. For generation of the edge map, we used parameter values of $T = 96$ (Sobel algorithm threshold) and $T = 250$ (minimum length of edge pixel chains). These two parameters can be varied to give more or less enhancement. Our experience suggests that reasonable values are $90 \leq T \leq 102$ and $100 \leq T \leq 500$. Figure 4 compares the unmodified and enhanced versions of a test image, and also shows the edge map. DV and BV have been exploited as the quantitative measures of the image enhancement in Ref. [1], 21. In this measure, the image is split up to the detail region and the background region in terms of the local variance of the noiseless original image. Increment of the average of the variances of background region (BV) is not desired in image enhancement. Degree of enhancement is controlled by h in both methods. The proposed method can reduce BV to about a half of the cubic unsharp masking. Especially, BV of the enhanced image is less than the input image of which noise variance is 150. The enhancement results from the cubic unsharp masking and the proposed method are shown in Fig. 5(b) and (c), respectively. In these results, DV is specified as 1100 with appropriate A. Comparing with the proposed method and the cubic unsharp masking, the proposed method can significantly reduce noises that appear in the homogeneous regions of the input image.

V. CONCLUSION

We have described an automated edge enhancement algorithm for clustered-dot halftones. The algorithm operates by trading off gray level resolution for finer spatial resolution in the immediate vicinity of edges in the image. Experimental results have demonstrated that the algorithm produces noticeably sharper edges in natural images, text, and line art, without introducing disturbing artifacts. Although the testing so far has been done with only one type of (laser) printer and one dithering matrix, we expect that with suitable modifications, the algorithm will also give good results with ink-jet printers and different matrices. Moreover, it is very likely that our techniques could be extended to the case of color half toning. In this paper, we propose an edge-weighted unsharp masking filter for enhancement of noisy images. The statistical analysis of the noise amplification leads the optimum parameters of the weighting function to minimize the variance of noise components appears in the enhancement result. The statistical analysis also shows that the proposed method can reduce the noise variance to about 1/5 to 1/10 of the weighted unsharp masking filter. The proposed method requires the computation of the wavelet transforms over two scales. The computational cost of this extension of the weighting function corresponds to the separable low pass filter for scaling of the wavelet transform. We show that the noise reduction capability can be obtained with relatively small computation costs for the edge-weighted unsharp masking.

References

- [1] H. Trontelj, J. E. Farrell, J. Wiseman, and J. Shu, "Optimal half toning algorithm depends on printing resolution," in SID Int'l. Symp. Dig. of Tech. Papers, vol. XXIII, pp. 749-752, Soc. for Info. Display, 1992.
- [2] J. Canny, "A computational approach to edge detection," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-8, pp. 679-698, Nov. 1986.
- [3] G. Ramponi, "A simple cubic operator for sharpening an image," Proc. IEEE Workshop on Nonlinear Signal and Image Processing, Halkidiki, Greece, pp. 963-966, June 1995.
- [4] G. Ramponi, "A cubic unsharp masking technique for contrast enhancement," Signal Processing, vol. 67, no. 2, pp. 211-222, June 1998.
- [5] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 14, no. 7, pp. 710-732, July 1992.
- [6] B. M. Sadler and A. Swami, "Analysis of multiscale products for step detection and estimation," IEEE Trans. on Information Theory, vol. 45, no. 3, pp. 1043-1051, April 1999.
- [7] T. Abdulkirim, S. Takano and K. Nijima, "The Dyadic lifting schemes and the denoising of digital images," International Journal of Wavelets, Multiresolution and Information Processing, Vol.6, No.3, pp:331-351, 2008.
- [8] Atabany, W., and Degenaar, P. "A Spatiotemporal Parallel Image Processing on FPGA for Augmented Vision System", Proceedings of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering. December 2007.