Edge Enhancement of Remote Sensing Image Data in the DCT Domain

Biao Chen, Shahram Latifi
Dept. of Electrical and Computer Engineering

Junichi Kanai
ISRI

University of Nevada, Las Vegas, Las Vegas, NV89154
{biaochen, latifi}@ee.unlv.edu, kanaij@isri.unlv.edu
(702)895-1428, (702)895-4016, (702)895-1070

November 14, 1997

1. Introduction

In image processing, processing speed, memory consumption, and computational complexity are people's most concerns. Manipulating compressed images is one efficient way to handle image data because most image data are usually archived or transmitted in a compressed form. Many data compression standards, such as JPEG, MPEG and H.261, use the Discrete Cosine Transform (DCT) because most coefficients in the DCT domain are small and quantized to zero. In other words, the data rate in the DCT domain is much lower than that in the spatial domain. Therefore, manipulating images by directly calculating only the non-zero DCT coefficients in the transformed domain without inverse transform can greatly reduce the storage requirement as well as computational time.

Much research was done on manipulating the compressed images base on the DCT properties. Smith and Rowe [17] presented a set of algorithms for algebraic operations including scalar addition/multiplication and pixel-wise addition/multiplication. Chitprasert and Rao [5], Ngan and Clarke [11], and Lee and Lee [8] proposed algorithms for image filtering in the DCT domain.

Based on the research mentioned above, Chang, Wang and Messerschmitt [3][4] developed algorithms for the image translation, scaling, overlapping(opaque and semi-transparent), linear filtering and pixel multiplication that can be applied to image subtitling, anti-aliasing, and masking. They further derived some new algorithms [1][2] for image shearing and rotation using multi-pass techniques and matrices multiplication, and techniques for the image/video indexing, editing, searching and matching in the DCT domain. Shen and Sethi [13][14] applied the above algebraic operations and developed their inner-block manipulation and scanline algorithms for compressed still/video images with more mathematical details. Other important operations, such as locating areas of interest and detecting coarse edge, were also described in [15].

The computational complexity of the DCT domain manipulations depends on the image compression rate and the sparsity of manipulation matrices. For block-wise operation, it can be reduced by about 65%, and speedup can be more than 100 to 1 without obvious quality degradation according to the experimental data [2][14][17].

Here, we develop some new algorithms for contrast manipulation and spatial feature manipulation including highpass filtering, contrast stretching, and edge enhancement in the DCT domain. Section 2 describes these algorithms in detail and shows the examples. In Section 3, the analysis and evaluation of the performance are addressed. The paper concludes in Section 4.

2. Image Enhancement in the DCT Domain

Enhancement of remote sensing image data involves highlighting the distinction between the features in the image so that its visual quality is improved. There are many enhancement techniques. Here, we discuss contrast manipulation and spatial feature manipulation including *Contrast Stretching, Highpass Filtering, and Edge Enhancement.* The Lena image is used in our experimentation to show the visual effects.

2.1 Contrast Manipulation

Low contrast makes it difficult to recognize the details of an image. The reason of this situation is that data in the image use only a small portion of possible gray levels of the display, while most display typically work over a range of 256 gray levels. Contrast stretching is one category of contrast manipulation, which intends to extend this narrow range of image data over a wider range such that the full 256 gray levels of the display can be used entirely.

Linear stretching is a uniform expansion of the gray scale of an image. It modifies each pixel in an image using the following equation:

$$output = (\frac{input - MIN}{MAX - MIN}) \times 255 \tag{1}$$

where *input* is the digital number of the pixel in the input image, *output* is the digital number assigned to pixel in the output image, *MIN* is the minimum gray level of the input image and *MAX* the maximum gray level of input image [9].

According to the linearity of the DCT, linear contrast stretching can be calculated directly in the DCT domain by

$$DCT(output) = \frac{255}{MAX - MIN}[DCT(input) - DCT(MINmatrix)] \tag{2}$$

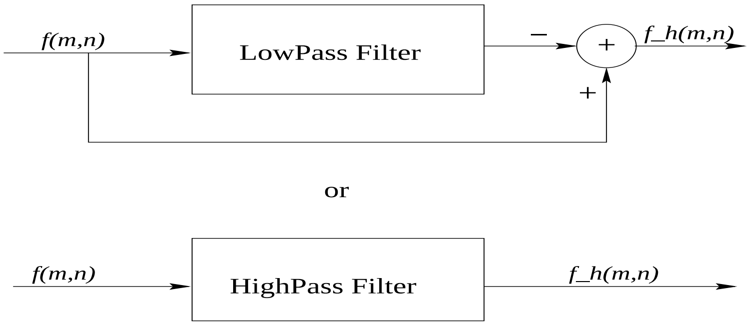
where *input* and *output* are the input and output image blocks, respectively, and MINmatrix is an 8-by-8 matrix with all elements equal to MIN. In reality, the subtraction must be performed only on the DC component. The range of gray levels of the image as well as the MIN value needs be available prior to the operation. Later in the section, we will use this operation as one of the steps needed to perform edge enhancement.

2.2 Spatial Filtering and Filtering in the DCT Domain

A. Theory of Spatial Filtering

Spatial filtering attempts to modify the pixel values according to the gray levels of neighboring pixels. Here we only consider the spatial lowpass and highpass filtering. Spatial lowpass filtering

(a) Spatial LowPass Filter



(b) Spatial HighPass Filter

Figure 1: Spatial Filter. $f(m,n)$, $f_l(m,n)$ and $f_h(m,n)$ are the original image, lowpass filter output and highpass filter output, respectively

can be viewed as a spatial averaging operation which can smooth and blur the image, while spatial highpass filtering is useful in sharpening and extracting the edges of the image.

Spatial averaging, in an image, means replacing the current pixel with the average value of the pixels around it to emphasize the low frequency features. Figure 1(a) shows spatial lowpass filter and Figure 2(a) shows a sample lowpass filter kernel [10]. The filtering can be implemented in the spatial domain by convolving an image with the filter kernel. The filter kernel is an array with properly assigned coefficients of odd size, such as 3x3, 5x5 or 7x7. The convolution operation is to move such a kernel throughout the original image, and the output of the pixel value at the center of the kernel is calculated as the weighted sum of corresponding pixel values. In other words, if we apply a lowpass filter in Figure 2(a), each pixel in the output image is equal to the average of 8 pixels around it and itself in the input image.

On the other hand, the highpass filter emphasizes high frequency components in an image. Highpass filtered images can be obtained by either subtracting a lowpass filter output from the original image

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

(a) A lowpass filter kernel

1	-2	1
-2	4	-2
1	-2	1

(b) A highpass filter kernel

Figure 2: Filter Kernel

or convolving a highpass filter kernel with the original image directly. The highpass filtering theory and an example of highpass filter kernel are shown in Figures 1(b) and 2(b).

B. Linear Highpass Filtering in the DCT Domain

When a kernel is separable, the convolution operation can be replaced by matrix multiplication operation, which is linear. In other words, we can treat the two-dimensional filter as two one-dimensional filters, which can be applied first in one direction, then in the other. Since the DCT is an orthogonal, separable and linear transform, two-dimensional separable linear filtering can be done in the DCT domain by multiplication with the pre-matrix and post-matrix for the vertical and horizontal direction filtering [1][3][4]. These matrices must be generated from an appropriate kernel.

Two solutions for highpass filtering are presented. Let B_i, B_{Li}, and B_{Hi} be an 8-by-8 image sub-block and the corresponding sub-blocks after lowpass and highpass filtering, respectively. One solution for linear highpass filtering is to subtract a lowpass filtered image from the original. Considering the lowpass filter as spatial averaging, we implement the filter kernel shown in Figure 2(a) by arranging the pre- and post-matrix as follows:

Pre-matrix and Post-matrix:

$$V_l = H_l = \frac{1}{3} \times \begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix} \tag{3}$$

Since the matrix is symmetric, it can be a pre-matrix a post-matrix. Therefore, the lowpass (B_{Li}) and highpass (B_{Hi}) filtered image sub-blocks in the spatial domain can be calculated by

$$B_{Li} = V_l \times B_i \times H_l \tag{4}$$

$$B_{Hi} = B_i - B_{Li} \tag{5}$$

Since the DCT is linear and distributive to the matrix multiplication, the above operations can be obtained in the DCT domain by

$$DCT(B_{Li}) = DCT(V_l) \times DCT(B_i) \times DCT(H_l) \tag{6}$$

$$DCT(B_{Hi}) = DCT(B_i) - DCT(B_{Li}) \tag{7}$$

An alternative way is to use a linear highpass filter directly, i.e. take the difference among pixels, with the filter as follows:

Pre-matrix and post-matrix:

$$V_h = H_h = \frac{1}{3} \times \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \tag{8}$$

Highpass filtered image sub-blocks in the spatial and DCT domain can be obtained by the following formulae.

$$B_{Hi} = V_h \times B_i \times H_h \tag{9}$$

$$DCT(B_{Hi}) = DCT(V_h) \times DCT(B_i) \times DCT(H_h) \qquad (10)$$

2.3 Edge Enhancement

As we discussed, high frequency components only emphasize the local contrast of the original image but do not preserve the low frequency brightness information contained in the image. Edge enhancement tends to keep both edge and low frequency information. Usually in remote sensing data processing, edge enhancement is implemented in three steps [9]:

1. The high frequency component image is obtained by edge extraction using a highpss filter. The roughness of the image decides the filter kernel size. The 'rougher' the image, the smaller is the size of the filter kernel.
2. All, or a portion, of the gray levels of the original image is added back to the highpass filtered image, so that the local contrast of the image is exaggerated and spatial details are more clear than before.
3. Contrast stretching is applied to the composite image.

As mentioned before, all three steps can be directly done in the DCT domain. Step 1 can be done using any of two approaches mentioned in Section 2.2. We have used both approaches to generate edge enhanced images. Figures 3, 4 and 5 show the original Lena image and edge detection using lowpass and highpass filters, respectively. Figures 6 and 7 represent the enhanced Lena image using the mentioned two approaches.

3. Results Analysis

By applying two kinds of filters to perform edge enhancement, we can see that the approach using lowpass filter leads to a better edge enhancement result in the DCT domain as compared to that using highpass filter. The approaches in both the spatial and DCT domain prove to be comparable

Figure 3: Original Lena image.

Figure 4: Highpass filtered Lena image through V_l and H_l in the DCT domain.

Figure 5: Highpass filtered Lena image using V_h and H_h in the DCT domain.

Figure 6: Enhanced Lena image using lowpass filter in the DCT domain with contrast stretching: 100% gray levels of the original image is added back.

Figure 7: Enhanced Lena image using highpass filter in the DCT domain with contrast stretching: 100% gray levels of original image is added back.

after examining the Mean Square Error(MSE) using the following equation.

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2 \tag{11}$$

where σ^2 is MSE, N is the total number of pixels in an image, x_n is the nth pixel value of the original image and y_n is the nth pixel value of the processed image. The result in the Table 1 shows that there is not much difference in image qualities obtained through the spatial and DCT domain operations.

Table 1. MSE values of processed image vs. original image and MSE of processed imagea in the DCT domain vs. that in the spatial domain.

	MSE(spatial domain)	MSE(DCT domain)	MSE(spatial vs. DCT)
Enhancement with LPF	54.158600	52.306549	2.794586
Enhancement with HPF	10.317596	11.907883	0.539871
Horizontal First Diff.	18865.115234	18940.140625	0.355395
Vertical First Diff.	18220.728516	18298.572266	0.381891

4. Conclusion

Image enhancement algorithms in the DCT domain have been presented. Contrast stretching, highpass filtering, and edge enhancement were implementable in the DCT domain by only using the matrix multiplication based on the properties of the DCT. Experimental results show that the quality of images generated by the new algorithms are comparable with that of images generated by the corresponding methods in the spatial domain. Our future work will focus on further studying the other manipulations in the DCT domain, which are useful to remote sensing data analysis.

5. Acknowledgment

This work was supported by NASA Research Grant NAG5-3994 and by NSF Research Grant IRI-9616206.

References:

[1] S.F. Chang, *"Some New Algorithms for Processing Images in the Transform-Compressed Domain"*, SPIE Symposium on Visual Communications and Image Processing, May 1995.

[2] S.F. Chang, *"Compressed-Domain Techniques for Image/Video Indexing and Manipulation"*, IEEE International Conference on Image Processing(ICIP95), Special Session on Digital Library and Video on Demand, Washington DC, Oct, 1995.

[3] S.F. Chang, W.L. Chen and D.G. Messerschmitt, *"Video Compositing in the DCT Domain"*, IEEE Workshop on Visual Signal Processing and Communications, Rayleigh, North Carolina, pp.138-143, Sept. 1992.

[4] S.F. Chang and D.G. Messerschmitt, *"Manipulation and Compositing of MC-DCT Compressed Video"*, IEEE Journal on Selected Areas in Communications, Vol 13, No. 1, Jan. 1995.

[5] B. Chiptrasert and K.R. Rao, *"Discrete Cosine Transform Filtering"*, Signal Processing, Vol. 19, No. 3, PP. 233-245, Mar. 1990.

[6] A.K. Jain, *"Fundamentals of Digital Image Processing"*, Prentice-Hall, 1989.

[7] R. Klette and P. Zamperoni, *"Handbook of Image Processing Operators"*, John Wiley & Sons, 1996.

[8] J.B. Lee and B.G. Lee, *"Transform Domain Filtering Based on Pipelining Structure"*, IEEE Transactions on Signal Processing, Vol. 40, No. 8, pp. 2061-4, Aug. 1992.

[9] T.M. Lillesand and R.W. Kiefer, *"Remote Sensing and Image Interpretation", second edition*, John Wiley. 1987

[10] J.S. Lim *"Two-dimensional Signal and Image Processing"*, Prentice-Hall, 1990.

[11] K.N. Ngan and R.J. Clarke, *"Lowpass Filtering in the Cosine Transform Domain"*, Int. Conf. Communication, Seattle, WA., pp. 31.7.1-5, June 1980.

[12] K. Sayood, *"Introduction to Data Compression"*, Morgan Kaufmann, 1996.

[13] B. Shen and I.K. Sethi, *"Inner-Block Operations On Compressed Images"*, Proc. of ACM Multimedia'95, pp. 490-499, San Francisco, CA, Nov. 1995.

[14] B. Shen and I.K. Sethi, *"Scanline Algorithm in Compressed Domain"*, Proc. SPIE vol.2668, Digital Video Compression Algorithms and Technologies 1996.

[15] B. Shen and I.K. Sethi, *"Direct Feature Extraction from Compressed Images"*, Proc. SPIE vol. 2670, Storage & Retrieval for Image and Video Databases IV 1996.

[16] M.A. Sid-ahmed, *"Image Processing - Theory, Algorithms, and Architectures"*, McGraw-Hill, 1994.

[17] B.C. Smith and L.A. Rowe, *"Algorithms for Manipulating Compressed Images"*, IEEE Computer Graphics and Applications, PP. 34-42, Sep. 1993.