

Direct feature extraction from compressed images[†]

Bo Shen and Ishwar K. Sethi
Vision & Neural Networks Laboratory
Department of Computer Science
Wayne State University
Detroit, MI 48202
{bos, sethi}@cs.wayne.edu

ABSTRACT

This paper examines the issue of direct extraction of low level features from compressed images. Specifically, we consider the detection of areas of interest and edges in images compressed using the discrete cosine transform (DCT). For interest areas, we show how a measure based on certain DCT coefficients of a block can provide an indication of underlying activity. For edges, we show using an ideal edge model how the relative values of different DCT coefficients of a block can be used to estimate the strength and orientation of an edge. Our experimental results indicate that coarse edge information from compressed images can be extracted up to 20 times faster than conventional edge detectors.

Keywords: Image compression, DCT domain, interest area extraction, edge detection, feature extraction

1. INTRODUCTION

The ability to process and transmit visual information quickly is the current driving force in computer and telecommunication hardware and software developments. The present view of computing environment is that of a huge information management system in which the information from real world is digitized, synthesized, indexed, and manipulated. The content-based retrieval of non-textual information is an integral component of this computing environment. As most multimedia information is in compressed form for efficient storage and transmission, it is important to treat compressed multimedia information as a first class data type, i.e. the compressed data should be operated upon directly instead of its decompressed version.

Extracting information from compressed data is not a new concept; it goes back to the work of Hsu et al¹⁷ who used Mandala transform as a way of automatic target recognition in compressed images. In recent years, however, it has received increasing attention due to an explosive growth in multimedia. The recent work on processing of compressed data has been either with respect to performing image manipulation⁶⁻⁹ or with respect to video cut detection¹⁰. All these reported works are based on properties of the discrete cosine transform (DCT), which is at the heart of current image and video compression standards such as JPEG¹⁴, MPEG¹³, H.261 and HDTV. It appears that the issue of extraction of low level features directly from compressed data has not yet been sufficiently examined. Since the extraction of such features is a fundamental step towards the characterization of image and video content for content-based retrieval, it is important to examine the possibility of low level feature extraction directly from compressed images for speedier characterization of content in the context of an extremely large volume of image data. The work described in this paper is an attempt in this direction. Using DCT coefficients directly, we show in this paper the extraction of areas of interest and edges in a coded image. The proposed methodology is extremely simple and fast; up to 20 times faster processing compared to conventional feature extraction is achieved. Although the quality of the resulting edge information is not very good, the proposed methodology can be used as a fast step to quickly locate low level features in a coded image.

2. JPEG/DCT MODEL OF COMPRESSION

[†]. Also submitted as Technical Report to Computer Science Department, Wayne State University, Jan. 1996.

Before describing how to extract low level feature information directly from DCT compressed images, we briefly describe the JPEG/DCT model of compression¹⁴. The original image is transformed using forward DCT to frequency domain on a block by block basis, the standard block size is 8x8. After forward DCT, quantization is performed on transform coefficients to discard visually unimportant ones. This is a lossy process and some compression is achieved in this step. The quantized DCT block becomes a sparse matrix. Then a lossless entropy coding (in most implementations, Huffman coding) is performed to get further compression. For decompression of JPEG file, the reverse processes are performed to reconstruct the image from compressed stream. In the decompression procedure, we call the coefficients obtained directly after Huffman decoding quantized DCT coefficients, and coefficients after dequantization dequantized DCT coefficients. Discrete cosine transform (DCT) is the heart of this compression scheme which is defined as

$$F_{uv} = \frac{C_u C_v}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i,j) \quad (1)$$

where

$$C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \text{ DC} \\ 1 & \text{otherwise AC} \end{cases}$$

One simple observation is that each DCT coefficient F_{uv} is a linear combination of all pixel values within the block. Our approach for direct extraction of low level features is based upon the relationship between the pixels' values in a block and its DCT coefficients. For example, the forward DCT transform tells us that the coefficient in the upper left corner of a DCT encoded block is the DC coefficient and it represents the average luminance of the block. The remaining coefficients are all called AC coefficients and the value of each AC coefficient reflects variations in gray level values in certain direction at certain rate. To see this relationship, consider the coefficient F_{10} . From the definition of discrete cosine transform,

$$F_{10} = \frac{c_1 c_0}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)\pi}{16} f(i,j) = \frac{c_1 c_0}{4} \sum_{i=0}^7 \cos \frac{(2i+1)\pi}{16} \sum_{j=0}^7 f(i,j) \quad (2)$$

Using the fact that $\cos(\pi - \theta) = -\cos \theta$, the above equation can be expanded as

$$F_{10} = \frac{c_1 c_0}{4} \left[\cos \frac{\pi}{16} \left(\sum_{i=0}^7 f(0,j) - \sum_{i=7}^7 f(7,j) \right) + \cos \frac{3\pi}{16} \left(\sum_{i=0}^7 f(1,j) - \sum_{i=6}^7 f(6,j) \right) + \right. \\ \left. \cos \frac{5\pi}{16} \left(\sum_{i=0}^7 f(2,j) - \sum_{i=5}^7 f(5,j) \right) + \cos \frac{7\pi}{16} \left(\sum_{i=0}^7 f(3,j) - \sum_{i=4}^7 f(4,j) \right) \right] \quad (3)$$

upper row 0,1,2,3
"- " bottom row 7,6,5,4

This representation means that the value of F_{10} essentially depends upon intensity difference in the vertical direction between the upper and lower parts of the input block. This is shown pictorially in Fig. 1 along with similar representations for four other AC coefficients. From this pictorial representation, it can be seen that different AC coefficients encode in some suitable way intensity changes within a block along different directions at different scales.

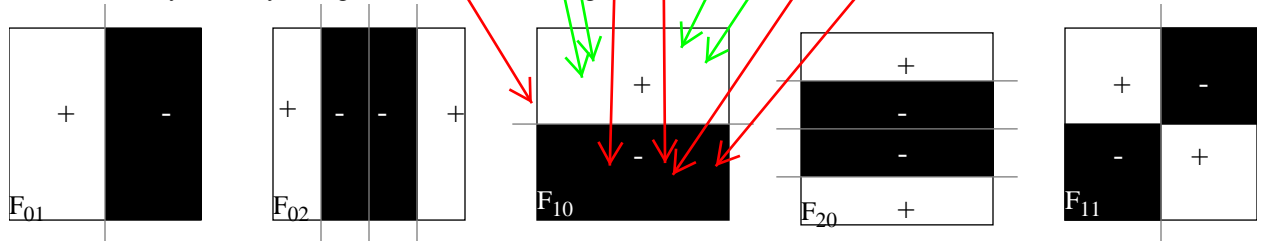


Fig. 1. Physical meanings of 5 AC coefficients

3. LOCATING AREAS OF INTEREST

Using AC Coefficients, one of the first task in many feature extraction schemes is to locate areas of interest, i.e. those parts

of an image which show sufficient intensity changes. The detection of such areas of interest is generally done by computing the following measure defined by Moravec¹⁵ which is actually the variance of a window of size $(a+1)$ by $(b+1)$ centered at pixel (x, y) in an image $f(x, y)$.

$$V(x, y) = \sqrt{\sum_{k=-\frac{a}{2}}^{\frac{a}{2}} \sum_{l=-\frac{b}{2}}^{\frac{b}{2}} [f(x, y) - f(x-k, y-l)]^2} \quad (4)$$

Since the variance measure of Eqn. (4) captures intensity changes within a window in all directions at all scales, it is possible to capture similar information within an encoded block by measuring its AC energy according to the following relationship,

$$A = \sum_{u=0}^7 \sum_{v=0}^7 F_{uv}^2 \quad (u, v) \neq (0, 0) \quad (5)$$

where F_{uv} is the AC coefficients. Instead of using the squared summation, it is also possible to use the following relationship with less computation.

$$A' = \sum_{u=0}^7 \sum_{v=0}^7 |F_{uv}| \quad (u, v) \neq (0, 0) \quad (6)$$

To provide an idea of how closely the relationship of Eqn. (6) captures the underlying intensity changes, we show in Fig. 2 the results of direct variance computation using a window size of 7×7 to match the JPEG block size of 8×8 , and its estimate through Eqn. (6). It is seen that the measure based on DCT coefficients is effectively able to locate all areas of high activity without a need for decompression.

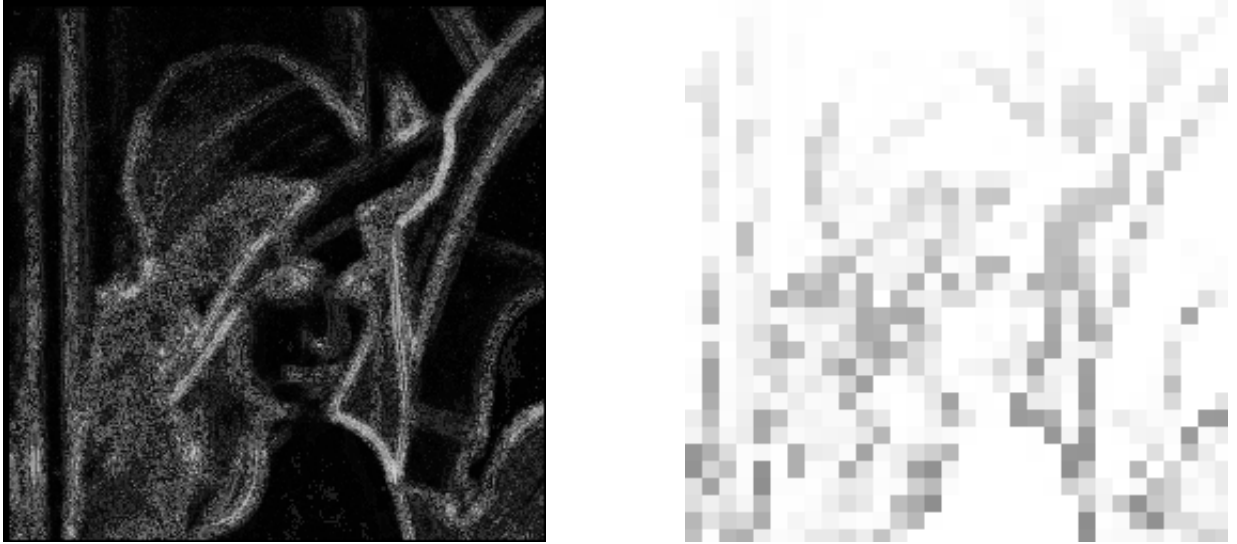


Fig. 2. Moravec interest operator vs. AC activity in DCT domain

4. FAST COARSE EDGE DETECTION

Edge is one of the most important features of visual information. Many content-based indexing and retrieving methods are based on the discrimination of edge information. As shown in the previous section, blocks containing edges can be extracted out by using ac measurement. One can decompress these blocks then perform edge detection operator on them to get the actual edges. This is already more efficient than decompressing the whole image and detecting edges in spatial domain. However, to use compressed data directly as feature extraction input, we try to obtain edge information without decompression which offers even better efficiency. This method eliminates the processes of decompression for any block

and convolution which is required for almost all spatial domain edge detection. Both of these are time-consuming processes.

There have been some preliminary feature extraction methods based on the classification of DCT coefficients. Some coarse features can be extracted out directly from transform coefficients¹⁻³. Edge blocks are decided by simply counting the number of non-zero coefficients in high and medium frequencies followed by decompress the block to get the actually edge³. For purpose of feature-based compression, there was work using generalized wavelet transform, which showed some effectiveness, but their purpose is for feature-based compression⁴. For purpose of block recovery, the edge orientation of the neighboring block of the lost block are estimated by using horizontal and vertical energy, then the edge orientation of the lost block is simulated based on some fuzzy logic operation⁵. But there is no unified, quantitative manner in compressed domain on the subject of edge detection. In the following, we examine the possibility of edge extraction from DCT coefficients using an ideal edge model.

4.1. Ideal edge model in DCT domain

In the model shown in Fig. 3, we consider an ideal step edge cutting through a block of size 8, which is the standard size for JPEG/DCT¹⁴. Based on the edge model, we are interested in relating three edge parameters, namely edge orientation (θ), edge strength (h) and edge offset from center (d) with the DCT coefficients of the block.

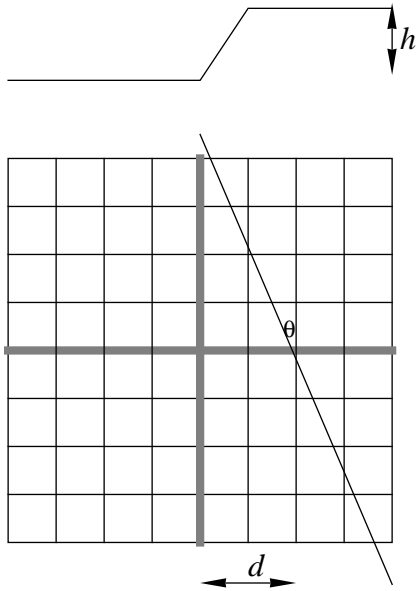


Fig. 3. Ideal step edge model

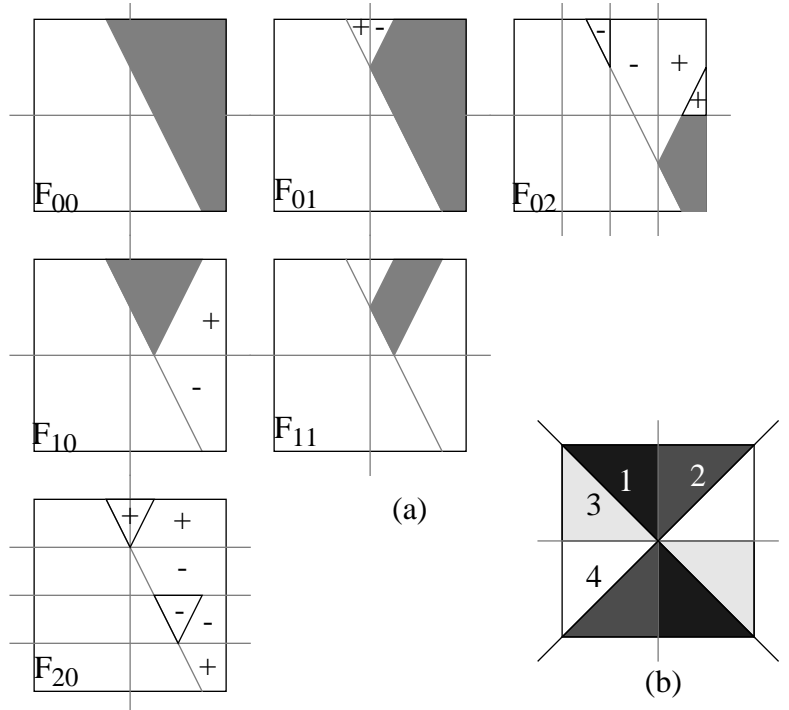










Fig. 4. Physical meanings of AC coefficients in a block which contains an ideal edge

4.2. Edge orientation









Fig. 4(a) shows in graphical form the contribution of the ideal edge to the first six DCT components of a block. Without loss of generality, we can assume the white area of intensity 0, and the shaded area of intensity h . It is seen from this figure that while the magnitude of F_{01} depends upon all the three edge parameters, its sign gives the edge step direction. The same applies to the remaining AC coefficients. To simplify, let us assume edge offset to be zero at first. As shown in Fig. 4(b), four sections can be defined for edge orientation. If the edge angle is within sections 1 and 2, the edge can be considered as a vertical-dominant edge. Similarly, it can be considered as a horizontal-dominant edge when the edge angle is within sections 3 and 4. For the zero edge offset, it is straight forward to prove the conclusions of Table 1. In fact it can be easily shown that relationships in Table 1 hold true even if the edge offset is not zero.

Table 1: Vertical-dominant and horizontal-dominant edges

Vertical-dominant	section 1	$ F01 > F10 $	$F01 \leq 0$	$F10 > 0$	
				$F10 < 0$	
	section 2		$F01 \geq 0$	$F10 > 0$	
				$F10 < 0$	
Horizontal-dominant	section 3	$ F01 < F10 $	$F10 \leq 0$	$F01 > 0$	
				$F01 < 0$	
	section 4		$F10 \geq 0$	$F01 > 0$	
				$F01 < 0$	

For special cases of true horizontal, vertical, or diagonal edges, again it is easy to show the relationships of Table 2.

Table 2: Vertical, horizontal and diagonal edges

Vertical	$F_{10} = 0$	$F_{01} > 0$	
		$F_{01} < 0$	
Horizontal	$F_{01} = 0$	$F_{10} > 0$	
		$F_{10} < 0$	
Diagonal	$ F_{01} = F_{10} $	$F_{01} < 0, F_{10} > 0$	
		$F_{01} > 0, F_{10} < 0$	
		$F_{01} > 0, F_{10} > 0$	
		$F_{01} < 0, F_{10} < 0$	

To get more accurate edge orientation information, one needs to look at more DCT coefficients. Based on a detailed

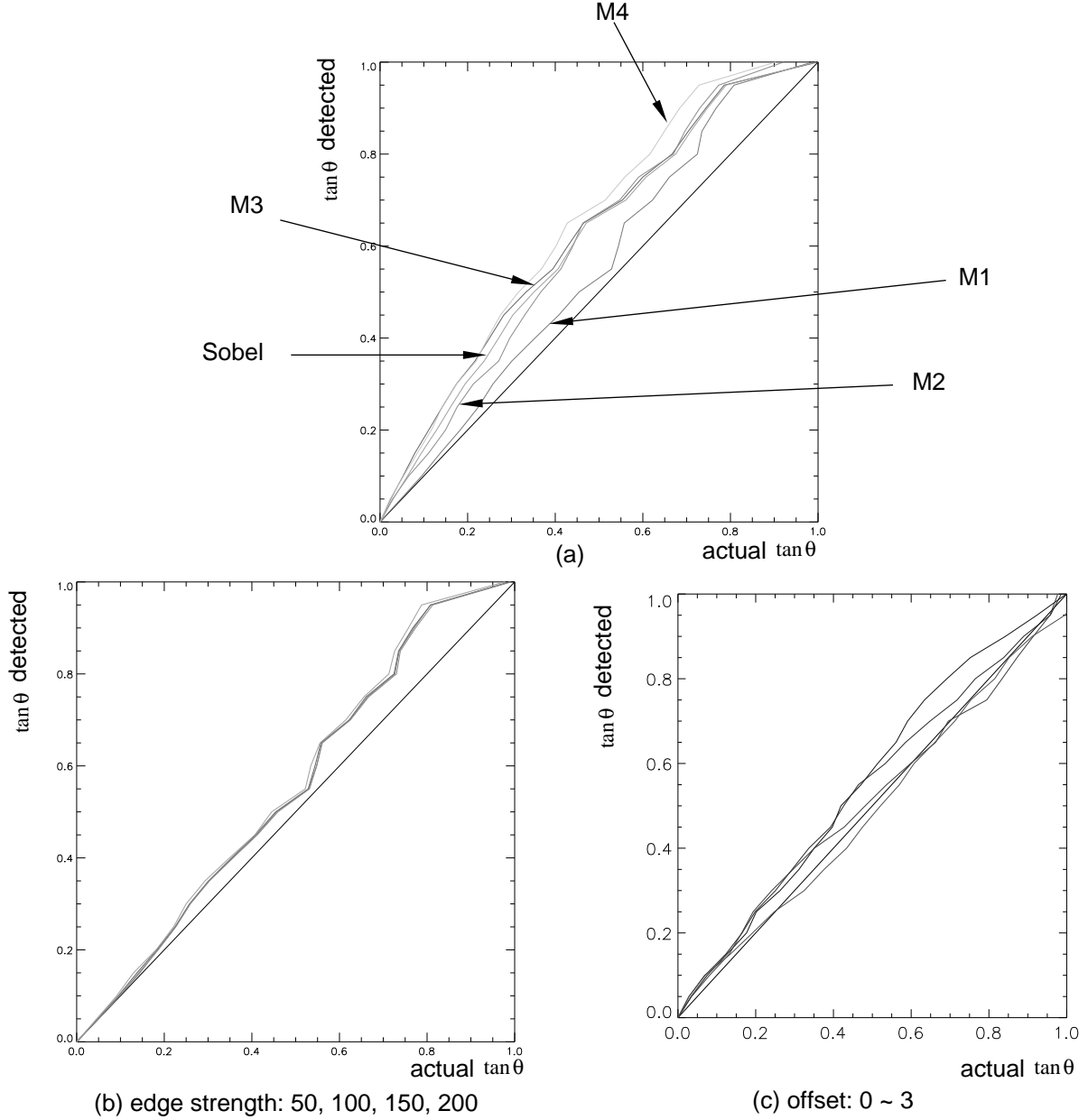


Fig. 5. The detection of edge orientation. (a) Actual vs. detected angle plots
(b) Performance of $M1$ metric against edge strength, (c) Performance of $M1$ metric against edge offset

analysis, we suggest the following four different metrics of DCT coefficients to obtain accurate edge orientation information:

$$M1 \quad \tan \theta = \left(\sum_{v=1}^7 F_{0v} \right) / \left(\sum_{u=1}^7 F_{u0} \right)$$

$$M2 \quad \tan \theta = \left(\sum_{v=1}^7 D_{0v} \right) / \left(\sum_{u=1}^7 D_{u0} \right)$$

$$M3 \quad \tan \theta = F_{01}/F_{10}$$

$$M4 \quad \tan \theta = D_{01}/D_{10}$$

The quantity D_{uv} in the expressions above stands for the coefficient obtained from compressed stream directly after Huffman decoding, and F_{uv} represents the dequantized DCT coefficient. To see how these metrics perform, we ran a simulation experiment simulating an ideal edge of different orientations. The results of this experiment are shown in Fig. 5(a). It is seen that the metric $M1$ provides the best edge orientation estimate. For comparison purposes, we also show the edge orientation estimate due to the Sobel edge operator. To see how well the metric $M1$ is affected by the other edge parameters, strength and offset, we repeated our simulation experiment by varying these parameters. These results are shown in Fig 5(b) and (c). As we see from these two figures, the metric $M1$ is fairly robust and has little to do with edge strength, and is not affected by edge offset. We also tried other metrics such as the one mentioned in Lee⁵ which uses horizontal and vertical energy, they did not provide satisfactory results.

4.3. Edge offset from center

For estimating edge offset, let us refer again to Fig. 4 showing the calculations of different DCT coefficients in a graphical form. It is easy to see that when an edge cuts through the block center, F_{11} must be zero. The reverse is not necessarily true, because if either F_{10} or F_{01} is zero then F_{11} must also be zero. However, F_{10} or F_{01} being zero implies that the edge is either a true vertical edge or a horizontal edge. The edge offset then can be computed by using F_{00} and F_{10} or F_{01} .

Also, F_{02} and F_{20} can be used to decide edge offset to a finer level. From Fig. 4- F_{02} , one can see that the sign of the coefficient can be used to decide whether the edge is to the right or to the left of the block center. The edge offset can be decided to two-pixel precision by the sign of F_{02} for vertical-dominated edges (or F_{20} for horizontal-dominant edges). In this fashion, even higher frequency coefficients can be used to decide edge offset to one-pixel precision. But since higher frequency coefficients are generally more severely quantized, more information loss is expected for these coefficients, they are less likely to be used independently to judge edge offset.

In our algorithm, we only use simple methods introduced above to get coarse offset information. One can derive better methods by considering more coefficients based on this DCT edge model. For instance, a look up table can be constructed for the relationship between the magnitude of F_{11} and edge offset.

4.4. Edge strength

To see how an estimate about edge strength can be made, consider the first row of pictures of Fig. 6 showing the representations of coefficients F_{10} while a vertical-dominant edge moving from left to right. The shaded area in each picture corresponds to the magnitude of coefficient F_{10} . In the central three picture blocks when the upper and lower bound of the edge are within the block, the shaded area is actually the same; it can be calculated as following.

Let A_0 to A_3 be the shaded areas along each row as marked in Fig. 6. Assuming unit edge step, we can compute

$$\begin{aligned} A_0 &= 2 \times \frac{4 \tan \theta + 3 \tan \theta}{2} \times 1 = 7 \tan \theta \\ A_1 &= 2 \times \frac{3 \tan \theta + 2 \tan \theta}{2} \times 1 = 5 \tan \theta \\ A_2 &= 2 \times \frac{2 \tan \theta + \tan \theta}{2} \times 1 = 3 \tan \theta \\ A_3 &= 2 \times \frac{\tan \theta}{2} \times 1 = \tan \theta \end{aligned}$$

According to Eqn. (3),

$$F_{10} = \frac{h}{\sqrt{2}} \times (A_0 \cos \frac{\pi}{16} + A_1 \cos \frac{3\pi}{16} + A_2 \cos \frac{5\pi}{16} + A_3 \cos \frac{7\pi}{16}) = 9.111 h \tan \theta$$

The above equations do not hold when the edge boundary is outside the block as shown in Fig. 6(a) and (e). This means

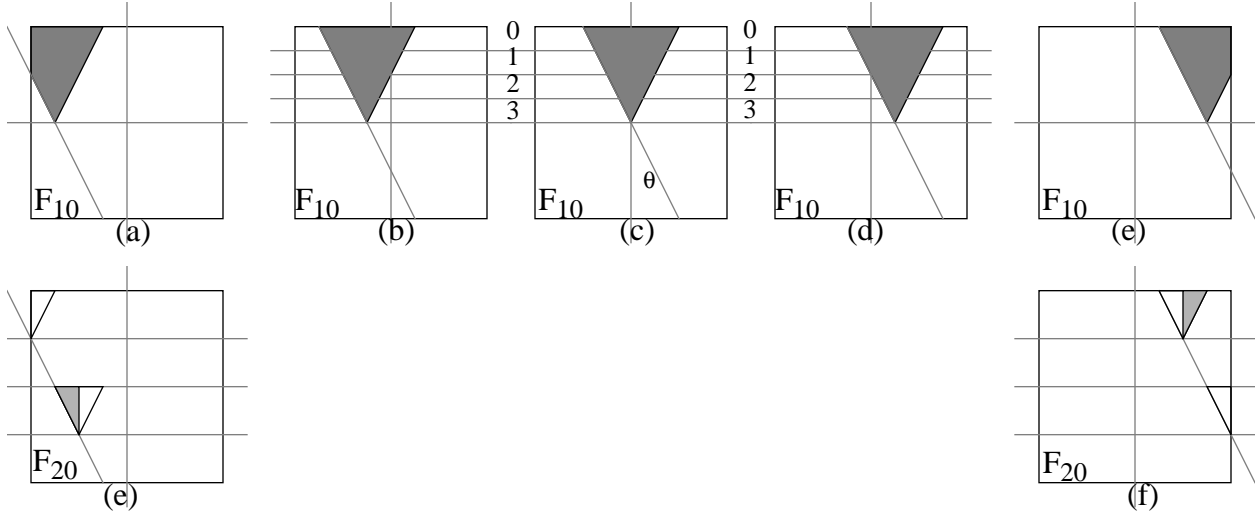


Fig. 6. Edge offset from center

that the relationship is not so true if the offset is larger than certain amount. However, we can compensate this by using the value of F_{20} . As shown in Fig. 6(e) and (f), the outside part of F_{10} have the same area as F_{20} represented, for actual computing, a weight α has to be used to adjust the value of F_{20} because F_{20} has different cosine value weights. Approximately thus

$$F_{10} + \alpha F_{20} = 9.111h \tan \theta$$

This gives us the relationship between edge strength and angle. Because we can find out the edge angle roughly from Section 4.2 the edge strength can be calculate as:

$$h = \frac{F_{10} + \alpha F_{20}}{9.11 \tan \theta} \quad (7)$$

If it is a horizontal-dominant edge, following the same derivation, we have,

$$h = \frac{F_{01} + \alpha F_{02}}{9.11 \tan \theta} \quad (8)$$

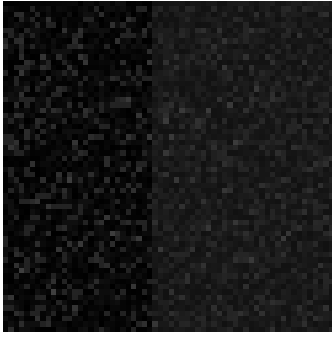
Finding more equations using other coefficients, we can use constrains to achieve better (more robust) results. Noise may affect some coefficients but not all.

5. PERFORMANCE EVALUATION

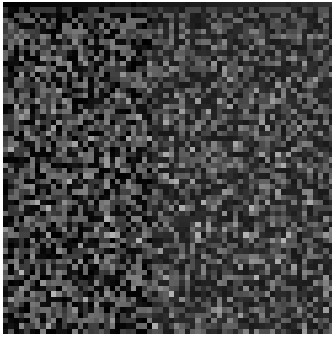
To show how the suggested use of DCT coefficients provides information about edges, we present in this section results from two computer simulation experiments. The first experiment is performed to show that the edge detection using DCT coefficients will hold in present of noise. For this, three noisy vertical edges were generated by adding independent Gaussian noise of standard deviation σ . These edges are shown in the first column of Fig. 7 along with the corresponding signal-to-noise ratio which is defined as $SNR=(h/\sigma)^2$, where h is the edge strength normalized to the range 0 to 1. The central column of Fig. 7 shows the detected edge due to Sobel operator. The DCT edge detection result is shown in the last column. For each case, the figure-of-merit of edge detection is calculated using the following measure defined in Pratt¹¹.

$$R = \frac{1}{I_N} \sum_{i=1}^{I_A} \frac{1}{1 + ad^2}$$

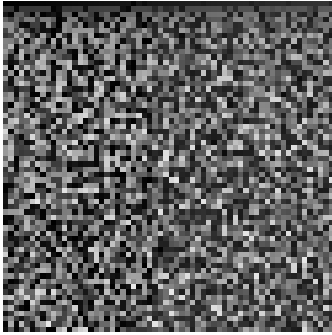
where $I_N = \text{MAX}(I_I, I_A)$ and I_I and I_A represent the number of ideal and actual edge map points, a is a scaling constant, and d is the separation distance of an actual edge point normal to a line of ideal edge points. The scaling factor a is used to



SNR=100



SNR=11



SNR=4

Gaussian edge

Sobel

DCT edge model based

Fig. 8. Edge detection with the presence of Gaussian noise

penalize edges that are localized but offset from the true position. We select $a=1/9$ in our experiment.

Table 3: Edge detection figure of merit

SNR	R (Sobel)	R (DCT edge model based)
100	99.8%	100.0%

Table 3: Edge detection figure of merit

SNR	R (Sobel)	R (DCT edge model based)
11	84.0%	95.4%
4	30.6%	82.4%

Table 3 summarizes the results of figure-of-merit computation. It is seen from this table as well as from Fig. 7 that DCT based edge detection provides a performance comparable to Sobel operator.

The second experiment was performed using two well-known real images of size 256x256. The experiments are conducted on a SUN SPARCstation-5 with 70MHz CPU. The images are gray scale images and compressed by using default quantization table. A public domain JPEG codec¹² is used in the traditional approach - decompress then apply gradient operator on the decompressed version to obtain the edge map. In DCT domain approach. This codec is also used in our algorithm only for Huffman decoding in the JPEG decompression processes to obtain the DCT coefficients.

Fig. 9 shows the comparison between two approaches. Fig. 9(b) is the simulated edge map using the edge parameters found by DCT domain approach. Since we compute these edge parameters on a block by block basis by using DCT coefficients directly for each block, the process is much faster but the result is not so precise as that of the spatial domain pixel-based convolution. Table 4 shows edge detection time (CPU time) for fruit image with different sizes. In the conventional

Table 4: Edge detection on fruit.jpg with different sizes

Size	Compression Ratio	Decompress then Sobel (sec.)	DCT domain (sec.)	Speedup
128x128	3.53	0.2+0.28	0.03	16.0
256x256	5.57	0.2+1.1	0.06	21.6
512x512	7.90	0.6+4.7	0.20	26.5

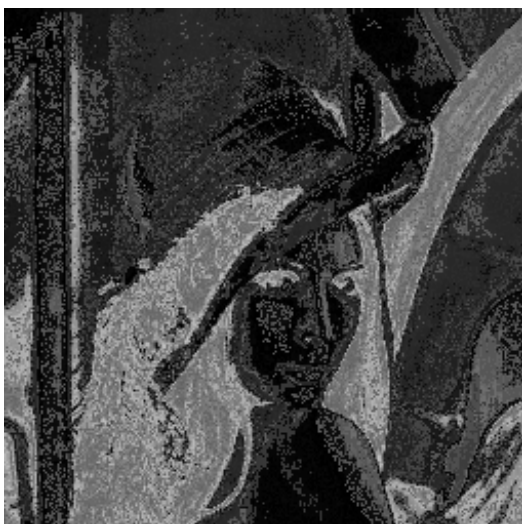
approach, the processing time includes decompression and spatial domain Sobel edge detection. With higher compression ratio, the speedup may be even larger. For video processing, the DCT domain approach can process at a frame rate around 16 fps (frames per second).

6. CONCLUSION

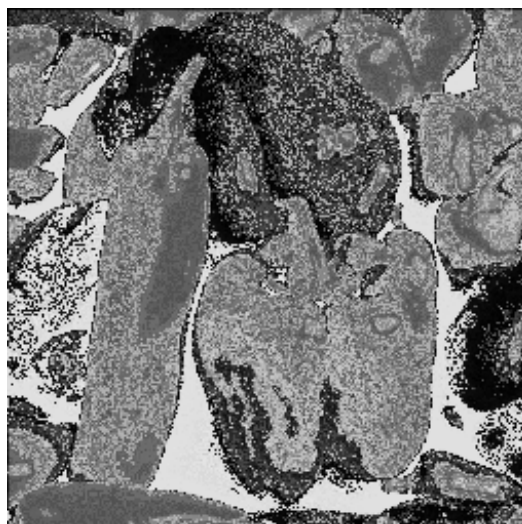
We showed that by using DCT coefficients directly, edge information can be extracted much more faster and to a fairly good extent. We suggest it is good enough for some coarse classification or feature based scene detection in video sequence¹⁶. In this kind of process, edge detection has to be applied on each (key) frame, faster process for each frame is desired. Also, in large scale image/video databases, tremendous amount of image/video needs to be processed for effective content-based indexing, fast processing approach is crucial for the efficiency of the whole system.

The edge information obtain in our algorithm is in model based format, it is useful for postprocess to get better edge map. For instance, edge strength parameter can be used for edge connection, false edge elimination. It is also easy for higher level or symbolic manipulation for extraction of more meaningful visual features. Future direction would be to use edge information extracted directly from compressed image for feature based classification which leads to fast and effective indexing of visual information. In case of classifications which need more accurate edge information, we are developing fast edge detection method using compressed domain convolution. It can provide even better edge map than traditional method with less processing time.

The concept of extracting low level features from compressed image or video can well be expanded to feature extraction directly from compressed audio. It is a desired technique for large digital audio databases. Some form of discrete cosine



(a)



(b)

(c)

Fig. 9. (a) Original (b) DCT domain (c) Traditional Sobel

transform is used in MPEG audio compression, extraction of low level features of audio can also help the classification of video clip.

REFERENCES

- [1] J. W. Kim, and S. U. Lee, "Discrete Cosine Transform - Classified VQ Technique for Image Coding", *Intl. Conf. on Acoust., Speech, and Signal Process*, pp. 1831-1834, Glasgow, Scotland, May 1989.
- [2] Y. Ho and A. Gersho, "Classified Transform coding of Images Using Vector Quantization", *Intl. Conf. on Acoust., Speech, and Signal Process*, pp. 1890-1893, Glasgow, Scotland, May 1989.
- [3] F. Arman, et al, "Image Processing on Compressed Data for Large Video Databases", *Proc. ACM Intl. Conf. Multimedia*, pp. 267-272, June 1993.
- [4] R. Wilson, et al, "Feature Extraction for Low Bit Rate Image coding Using a Generalized Wavelet Transform", *SPIE Advanced Image and Video Communications and Storage Technologies*, vol. 2451, pp. 249-257, Feb. 1995.
- [5] X. Lee, et al, "Information Loss recovery for Block-Based Image Coding Techniques - A Fuzzy Logic Approach", *IEEE Trans. Image Processing*, vol. 4, no. 3, Mar. 1995.
- [6] S. -F. Chang and D. G. Messerschmitt, "Video Compositing in the DCT domain", *IEEE Workshop on Visual Signal Processing and Communications*, pp. 138-143, Raleigh, NC, Sept. 1992.
- [7] S. -F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video", *IEEE JSAC Special Issue on Intelligent Signal Processing*, 1994.
- [8] B. C. Smith and L. Rowe, "Algorithms for Manipulating Compressed Images", *IEEE Computer Graphics and Applications*, pp. 34-42, Sept. 1993.
- [9] B. Shen and I. K. Sethi, "Inner-Block Operations On Compressed Images", *Proc. ACM Intl. Conf. Multimedia'95*, pp. 490-499, San Francisco, Nov. 1995.
- [10] I. K. Sethi and N. Patel, "A Statistical Approach to Scene Change Detection", *SPIE Proc. Storage & Retrieval for Image and Video Databases III*, vol. 2420, San Jose, CA, Feb. 1995.
- [11] W. K. Pratt, *Digital Image Processing*, second edition, Wiley-Interscience Publication, 1991.
- [12] T. Lane, *Independent JPEG Group Software Codec*, <ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6.tar.gz>
- [13] D. L. Gall, "MPEG: A Video Compression Standard for Multimedia Applications", *Communications of the ACM*, vol. 34, no. 4, Apr. 1991.
- [14] G. K. Wallace, "The JPEG Still Picture Compression Standard", *Communications of the ACM*, vol. 34, no. 4, Apr. 1991.
- [15] D. H. Ballard and C. M. Brown, *Computer Vision*, pp. 68-70, Prentice-Hall, Englewood Cliffs, New Jersey, 1982
- [16] R. Zabih, J. Miller and K. Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", *Proc. ACM Intl. Conf. Multimedia*, pp. 189-200, San Francisco, Nov. 1995.
- [17] Y. S. Hus et al, "Pattern Recognition Experiments in the Mandala/Cosine Domain", *IEEE Trans. Pattern analysis and Machine Intelligence*, vol. PAMI-5, no. 5, pp. 512-520, Sept. 1983.