

A Block-Edge-Pattern-Based Content Descriptor in DCT Domain

Jianmin Jiang, Kaijin Qiu, and Guoqiang Xiao

Abstract—In this correspondence, we describe a robust and effective content descriptor based on block-edge patterns extracted in discrete cosine transform domain, which is suitable for applications in JPEG or MPEG compressed images and videos. This content descriptor is constructed by a run-length edge-block histogram with three patterns including horizontal edge, vertical edge and no edge. In comparison with existing descriptors, the proposed features: 1) low-cost computing suitable for real-time implementation and high-speed processing of compressed videos; 2) robust to orientation changes such as rotation, noise, reverse, etc.; 3) operates in compressed domain. Extensive experiments support that the proposed content descriptor is effective in describing visual content, and achieves superior performances in terms of retrieval precision and recall rates.

Index Terms—Compressed-domain video analysis, content descriptors, MPEG.

I. INTRODUCTION

SINCE many images and videos distributed or stored inside computers are now represented in compressed formats at the source, research on compressed-domain image or video processing becomes increasingly important. Representative work in this area can be briefly summarized as: 1) texture feature extraction via statistical measures of discrete cosine transform (DCT) coefficients [1], [4]; 2) reorganization of DCT coefficients to extract color and shape features [2]; 3) statistical feature extraction from DCT or wavelets via mean and standard deviation [3], [5]; and 4) optical flow estimation in MPEG-2 [6].

Extensive research on image processing for the past decades illustrate that edge is a strong feature for characterizing visual content. While most techniques reported in the literature extract edge features in pixel domain, including those widely reported for image segmentations [7], recent trend on image processing is characterized by detecting edges and classifying their patterns at a block level either in pixel domain or compressed domain [8], [9]. Compressed-domain edge extraction is mainly represented by using: 1) the polarities of the projection of DCT coefficients [10]; 2) adaptive edge operator selection based on the mean, variance, and entropy of DCT coefficients [7]; and 3) pattern analysis of DCT coefficients to extract edge strength and orientations [11]–[13].

Manuscript received June 26, 2007; revised October 19, 2007. This work was supported by the EU IST FP6 Programme under the IP project: Live staging of media events under Contract IST-4-027312. This paper was recommended by Associate Editor P. L. Correia.

J. Jiang is with the Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China, and also with the School of Informatics, University of Bradford, Bradford BD7 1DP, U.K. (e-mail: j.jiang1@bradford.ac.uk).

K. Qiu and G. Xiao are with the Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: qkjqkj@swu.edu.cn, gqxiao@swu.edu.cn).

Digital Object Identifier 10.1109/TCSVT.2008.924106

Based on the analysis of the existing work as described above, we propose a block-edge-pattern (BEP)—based content descriptor, in which only three BEPs—no edge, horizontal edge, and vertical edge, are extracted directly inside DCT domain. In comparison with the existing work, our contributions and their novelty can be highlighted as: 1) a new run-length (RL) histogram to exploit the strength of the multidimensional histograms across all dimensions, yet avoid the high dimensional problems with conventional histograms; 2) extraction of three-edge patterns rather than five to further improve the processing speed and save computing cost without compromising the performance of content description.

The rest of the article is organized into four further sections, where Section II briefly revisit the work reported in [12] and present our argument that extraction of three-edge patterns rather than five is sufficient for the proposed content descriptor, Section III describes our proposed RL histogram to construct the content descriptor. Finally, experimental results and their analysis are given in Section IV to support the proposed design, and conclusions are given in Section V.

II. EXTRACTION OF BEPs IN DCT DOMAIN

In 2005, a technique of extracting five-edge patterns is proposed [12]. This technique operates in compressed domain by just analyzing the value of the first few DCT coefficients and thus only limited decoding operations are required. Such block edge pattern classification is based on the principle that, when a block of 8×8 pixels is divided into four sub-blocks of 4×4 pixels, comparison of their average pixel value decides whether there exists an edge pattern inside the block or not. As an example, if the average pixel of the two sub-blocks on the left is significantly larger than the average pixel of the two sub-blocks on the right or vice versa, the block can be classified as a vertical edge pattern block. While the principle has been applied by many other researchers [13], the contribution of [12] specifies that such edge pattern classification can be carried out by extracting edge measure values in DCT domain, which are summarized in Table I, where $X(1, 0)$, $X(0, 1)$ and $X(1, 1)$ are the first three DCT coefficients (excluding the DC) along the zig-zag scanning order.

While it is much simpler to calculate the edge measure values in DCT domain, it still requires four additions and two multiplications to calculate $\delta_{\pi/4}$ and $\delta_{3\pi/4}$, respectively. Yet if we only use three-edge patterns, as seen from Table I, significant savings on computing cost could be achieved since calculation of δ_0 and $\delta_{\pi/2}$ does not incur any addition or multiplication and only two DCT coefficients instead of three are needed. In addition, our further observation and analysis on its content application verifies that image edge distribution can be characterized by three-edge patterns instead of five. In other words, we only need the edge patterns of no edge, vertical edge, and horizontal edge to formulate their content applications without compromising the quality of visual content description. This is because

TABLE I
MEASURE FOR FIVE-EDGE PATTERNS

Edges	Measure values
No edge	$\hat{\lambda}$ (set by user, $\hat{\lambda} = 50$ in our design)
0	$\delta_0 = X(1,0) $
$\pi/4$	$\delta_{\pi/4} = \frac{3}{4} \max \left\{ \begin{array}{l} \frac{1}{2}(X(0,1) + X(1,0) + X(1,1)), \\ \frac{1}{2}(X(0,1) + X(1,0) - X(1,1)) \end{array} \right\}$
$\pi/2$	$\delta_{\pi/2} = X(0,1) $
$3\pi/4$	$\delta_{3\pi/4} = \frac{3}{4} \max \left\{ \begin{array}{l} \frac{1}{2}(X(0,1) - X(1,0) + X(1,1)), \\ \frac{1}{2}(X(0,1) - X(1,0) - X(1,1)) \end{array} \right\}$

TABLE II
DISTORTION MEASURE OF EDGE-PATTERN IMAGES
AGAINST THEIR SOURCE IMAGE

Image Samples	DM _s (dB)	DM _i (dB)
Lena	59. 439892	59. 180862
Peppers	57. 447807	57. 687534
Clown	50. 144257	50. 109631
Camera	48. 590816	48. 425625
Boat	55. 761734	56. 058308
Bridge	47. 100334	47. 334343

that an arbitrary edge inside an image can be generally approximated by combinations of vertical and horizontal edges. To support the argument, we reconstruct an edge pattern image based on the extracted edge patterns and then apply a distortion measure (DM) by following the spirit of peak-signal-to-noise-ratio (PSNR) with respect to the original image [17], where the pixel value of the reconstructed edge image is defined as the corresponding average pixel values inside the sub-block, which are used to determine the block edge patterns. Summary of all the DM values for six popular image samples is provided in Table II.

Table II indicates that the three-edge pattern images only incur very small difference in DM values in comparison with that of five-edge pattern images. Such measurements support that, when the edge pattern extraction technique reported in [12] is further developed to construct content descriptors, it could be sufficient to only use the vertical edge pattern and the horizontal edge pattern.

Correspondingly, the three BEPs can be classified by

$$\text{BEP} = \begin{cases} \text{no_edge}, & \text{if } \max(\delta_0, \delta_{\pi/2}) < \lambda \\ \text{vertical_edge}, & \text{if } \delta_{\pi/2} \geq \delta_0 \\ \text{horizontal_edge}, & \text{otherwise.} \end{cases} \quad (1)$$

However, with five block edge patterns, their classification is done by

$$\text{BEP} = \begin{cases} \text{no_edge}, & \text{if } \max(\delta_0, \delta_{\pi/4}, \delta_{\pi/2}, \delta_{3\pi/4}) < \lambda \\ 0_edge, & \text{if } \delta_0 = \max(\delta_{\pi/4}, \delta_{\pi/2}, \delta_{3\pi/4}) \\ \frac{\pi}{4}_edge, & \text{if } \delta_{\pi/4} = \max(\delta_0, \delta_{\pi/2}, \delta_{3\pi/4}) \\ \frac{\pi}{2}_edge, & \text{if } \delta_{\pi/2} = \max(\delta_0, \delta_{\pi/4}, \delta_{3\pi/4}) \\ \frac{3\pi}{4}_edge, & \text{if } \delta_{3\pi/4} = \max(\delta_0, \delta_{\pi/4}, \delta_{\pi/2}). \end{cases} \quad (2)$$

As seen from (1) and (2), the three-edge pattern classification would not only achieve computing cost savings on edge measure calculation, but also on edge pattern classifications.

III. CONTENT DESCRIPTOR DESIGN

In this section, we follow the general route to design a content descriptor based on histograms for the edge block patterns extracted. Given blocks of 8×8 edge-block-histogram (EBH) pixels, their classified edge patterns are firstly transformed into a set of three elements, $\psi = \{\#, -, |\}$, where $\#$ = no-edge, $-$ = horizontal edge and $|$ = vertical edge. When a histogram is constructed by calculating the number of occurrences for each edge pattern, the histogram will only have three elements, which is often referred to as the 1-D EBH. As the number of bins in such 1-D is too small to characterize the image content, the occurrence of states can be counted instead by considering combination of edge patterns, leading to construction of high dimensional histograms. As an example, if we examine two edge patterns together as one state, the total number of elements inside the histogram can be increased to 9 (a 2-D EBH). When N edge patterns are considered together as one state, the resulting histogram will have 3^N bins or elements.

Low-dimensional EBH not only have small number of elements, but also contain little information for the location of those classified edge patterns inside the image, which are often useful to reflect the visual content. On the other hand, 5-D EBH and 6-D EBH incurs large number of states, which illustrate significant potential for high computing cost, especially when millions of compressed images need to be addressed via such content descriptors. To achieve an appropriate balance, we propose a RL histogram following the spirit of RL coding in the area of lossless data compression. Specifically, we raft scan all block edge patterns on both row and column basis to count the number of same patterns inside one single run. Every time a different edge block pattern is encountered, it is regarded as a broken run and thus a new run will be started. The output code for each run can be represented as

$$R_i = \{\alpha_i, \eta_i\} \quad (3)$$

where $\alpha_i \in \{\#, -, |\}$, and η_i stands for the number of α_i inside this single run.

R_i essentially represents the state of RLEBH. To reduce the statistical modeling cost [14], we group those long RL states together into one single state

$$R_i = \begin{cases} \{\alpha_i, \eta_i\}, & \text{if } \eta_i \leq T \\ \{\alpha_i, T\}, & \text{when } \eta_i > T. \end{cases} \quad (4)$$

In our implementation, we selected $T = 4$. As a result, the histogram can be represented as follows:

$$H_{\text{run}} = \frac{\{C_1, C_2, \dots, C_M\}}{M} \quad (5)$$

where C_i counts the number of the state R_i occurred inside the image after the RL coding is finished, M is 28, including 12 states in RL on row-based scanning, 12 states in RL on column-based scanning, and two special states “-|” and “|_” on both row and column basis.

The RL histogram given in (5) not only resolves the problem that histograms with high dimension increase their size in exponential order, but also provides non-redundant statistical analysis of the edge pattern distribution. This is achieved by the nature of RL, where same edge patterns are grouped together to formulate states rather than individually fixed patterns. When the histogram is constructed by such states, additional advantage can be gained as such that grouped edge patterns are capable of describing connected edges rather than individual patterns. Such connectivity is described by components of: 1) horizontal RL; 2) vertical RL; and 3) individual RL (run-length = 1). In this way, the proposed RL histogram not only describes the visual content in terms of edge pattern distribution, but also in terms of these connected edges, providing further potential information for shape, contour or internal texture of visual objects. To this end, the proposed content description is different from those edge-based descriptors [8], [9], where the technique proposed in [8] relies on Canny edge detector to extract salient points and connect them into edge curves via a water-filling filter, and the technique in [9] extracts edge map to formulate global features through fuzzy compactness and combination of shape information for image retrieval.

In addition, the proposed RL histogram is capable of preserving the strength of conventional histograms, as well as characterizing the connectivity of edges and the continuity of regional edge patterns. As seen in (5), the first three states with a run of 1 single edge pattern are exactly the same as those elements inside one dimensional histogram. These parts of RL histogram will make the same contribution in describing the visual content as 1-D conventional histogram. Further on, the states with one run of two edge patterns will function in the same principle as that of 2-D conventional histogram. The only difference is that among all the possible different states inside conventional 2-D histograms, only the states, where the two edge patterns are the same, are selected to construct bins inside the RL histogram to describe the visual content. Such selection is based on the ground that: 1) due to the fact that the content descriptor is mainly based on edges, their connectivity and continuity is more important than others in describing the visual content; 2) other states with different edge patterns are less likely to make useful contributions since most of them will be rejected in practical image content processing. One typical example is image segmentation, where detected edges need to be further processed to form elements of meaningful segmentations such as contours, etc; 3) the role of such states with two different edge patterns is partially played by the previous three states with one single edge pattern. For other states with three different edge patterns or more, the principle is the same. Therefore, the proposed RL histogram is essentially extracting those useful states from conventional histograms across all dimensions and consequently combining all their strengths together to describe the visual content, yet such a design overcomes the problem that high dimensional histograms generate high number of bins inside the histogram, making it computing intensive and non-manageable.

Finally, to describe the edge connectivity in two dimensions, the RL histogram is constructed by scanning along rows as well as columns, which is commonly used in processing two dimensional signals (such as DCT). In this way, the edge connectivity can be described in two dimensions rather than in

one dimension. Therefore, global connectivity such as diagonal edge connectivity will be described by collective contributions from both row-based edge connectivity and column-based edge connectivity.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

To evaluate the proposed content descriptor, we design two phases of experiments. The first phase is to use such a content descriptor to run content-based image retrieval (CBIR) and measure its effectiveness in terms of retrieval precision. For this purpose, a database of 14787 JPEG compressed images is established, all of which are manually classified into ten categories in order to provide a ground truth for measuring the performances. The ten categories include cars, flowers, bike, building, animal, people, mountains, sky, sunset, and textured. The second phase is to subject the randomly selected images inside the database to a range of orientation changes and then add them into the database to see how many such changed images can be retrieved when the original image is taken as a query. This is designed to test the robustness of the proposed content descriptors.

To provide a comparison with the proposed content descriptor, we select two existing algorithms [3], [15] as our benchmark to compare the performances on the principle that the existing techniques are comparable and the relevant comparisons are fair in terms of computing cost and algorithm complexity. The first benchmarking technique [3] operates in compressed domain via direct extraction of mean and variance out of each block with 8×8 pixels. A histogram is then constructed from the space of mean and standard deviation by quantizing the mean into four regions and the standard deviation into seven regions. Therefore, this technique provides good comparability that it operates in compressed domain and constructs the content descriptor based on histograms. The second benchmark was drawn from an existing PicToSeek CBIR system [15], where color features invariant to viewpoints were described in pixel domain and the indexing key was constructed via histograms. The invariant color space is a direct result of the RGB space transformation, which is specified as follows:

$$\{c_4, c_5, c_6\} = \begin{cases} c_4 \{RGB\} = \frac{R-G}{R+G} \\ c_5 \{RGB\} = \frac{R-B}{R+B} \\ c_6 \{RGB\} = \frac{G-B}{G+B} \end{cases} \quad (6)$$

The histogram-based indexing key is constructed as follows after the invariant color space is quantized into four states indexed by k

$$H(c_4c_5c_6) = \frac{\eta\{(c_4 \in k) \cap (c_5 \in k) \cap (c_6 \in k)\}}{W \times H} \quad k \in [0, 3] \quad (7)$$

where $\eta(\cdot)$ represents a counting operation on the number of state occurrences.

To calculate the similarity between histograms, we adopt the simple L_1 distance, which is chosen on the ground that: 1) it is widely used by the CBIR community, such as the two benchmarks [3], [15]; and 2) it has lower computing cost than standard Euclidean distance, yet preserves its strength in similarity measurement.

TABLE III
SUMMARY OF THE FIRST-PHASE EXPERIMENTS
MEASURED IN PRECISION RATES

Image Categories	Bench-mark-1	Bench-mark-2	RLEBH (3)	RLEBH (5)
Cars	0.42	0.41	0.92	0.55
Flowers	0.41	0.60	0.67	0.79
Bike	0.62	0.33	0.60	0.51
Building	0.38	0.29	0.45	0.73
Animal	0.30	0.28	0.42	0.41
People	0.62	0.34	0.82	0.27
Mountains	0.28	0.31	0.43	0.43
Sky	0.39	0.53	0.47	0.40
Sunset	0.50	0.58	0.53	0.52
Textured	0.81	0.27	0.61	0.65
Average	0.47	0.39	0.59	0.53

For the first phase, we randomly choose 200 images from the database, 20 out of each category, as the query images to search the database. The retrieval performance was measured in terms of precision, which is the number of retrieved images deemed to be relevant divided by the first N retrieved images. In our experiments, we designed N to be 12 in line with the benchmarks [3], [15] and other [8], [9]. All results in the first phase are summarized in Table III, where RLEBH (3) refers to the proposed RL histogram with three-edge block patterns and RLEBH (5) with five-edge block patterns.

From the experimental results given in Table III, the following can be seen.

- 1) The proposed RLEBH (3) achieves the best performance among all the algorithms tested in terms of the total-average precision measurement.
- 2) While the proposed RL histogram significantly outperforms the first benchmark across most categories, it is inferior to the first benchmark for both the category of Bike and Textured. This is because that the first benchmark descriptor is strongly texture oriented, where the mean and the standard deviation extracted from DCT domain collectively captures the texture information quite well. Yet all images under this category are overwhelmingly texture dominant, and thus the precision rate achieved by benchmark-1 is better than RLEBH. However, for other categories with strong texture orientation, the proposed RLEBH remains competitive. Even for the category Bike, the benchmark-1 only slightly outperforms the proposed.
- 3) In comparison with the second benchmark that is purely color-based descriptor, the proposed algorithm is only beaten in color dominant categories, such as Sunset and Sky. Within these two categories, color feature dominates the image content. For other categories, the proposed RL histogram provides superior performances on the ground that it provides additional advantage in capturing connected edge information towards the combination of multiple feature strength, including texture of gradients, shape, contour and edges of visual objects and thus it performs well except for those color dominated images, for which the second benchmark is well tailored.
- 4) In comparison with five-edge patterns, selection of three-edge patterns achieves better performance in six categories, maintains the same performance in one category, slightly inferior in two categories and inferior in 1 category, where

TABLE IV
SUMMARY OF SECOND PHASE EXPERIMENTAL RESULTS
MEASURED BY RECALL RATES

Methods	Bench-mark-1	RLEBH (3)	RLEBH (5)
Clockwise rotate by 10o	0.78	0.50	0.48
Anti-clockwise rotate by 10o	0.74	0.52	0.40
Zoom out 20%	0.96	0.72	0.76
Zoom in 20%	0.94	0.90	0.92
Inverse color	0.02	1.00	1.00
Horizontal upset	0.92	0.98	0.68
Vertical upset	0.92	0.94	0.88
Horizontal displacement 10%	0.82	0.50	0.48
Vertical displacement 10%	0.86	0.52	0.50
Adding Gaussian white noise	0.02	0.90	0.94
Adding salt& pepper noise	0.02	0.42	0.46
Adding speckle noise	0.08	0.60	0.66
Total average recall	0.59	0.71	0.68
ANMRR	0.1	0.08	0.09

the category of building is edge-dominant. But REBH (3) is still better than REBH (5) in terms of the overall performance, which verifies that selection of three-edge patterns is sufficient for content-based applications.

The second phase of experiments aims at evaluating the robustness of the proposed content descriptor in retrieving images with orientation changes, such as rotation, zoom in or out, and noisy effects etc. In this experiment, fifty images were chosen at random from the database. Specifically, each of the chosen images was processed as follows: clockwise rotation by 10 degree, anti-clockwise rotation by 10 degree, zoom out by 20%, zoom in by 20%, inverse color, horizontal reverse, vertical reverse, horizontal displacement by 10%, vertical displacement by 10%, adding Gaussian white noise (means equal to 0, variance equal to 0.04), adding salt & pepper noise (variance equal to 0.04) and adding speckle noise (variance equal to 0.04). Following that, all such processed images are added into the database. When one of the originals is taken as a query image, we would like to see how many of those processed images can be retrieved to test the robustness of the proposed content descriptor. To this end, we measure the retrieved results in terms of the recall rate and the average of the normalized modified retrieval rank (ANMRR) over all queries as described in [16].

All the experimental results produced in the second phase are summarized in Table IV, which indicates that the proposed content descriptor with both three- and five-edge patterns is more robust than the first benchmark in terms of the total average recall rate, and achieves superior performances measured by ANMRR values. Between the 3 edge pattern and 5 edge pattern selections, RLEBH (3) still achieves better performance in terms of the overall recall rate, although there exist some places that RLEBH (3) is inferior than RLEBH (5).

The robustness of the proposed descriptor can be explained by 1) the edge pattern is block based and extracted in compressed domain. Therefore, any regional corruption inside image will be measured in terms of blocks rather than pixels. While small region corruption is measured big in terms of pixels, it is much smaller or even ignorable in terms of blocks, as one block contains 64 pixels. 2) Corrupted pixels inside the same block may compensate with each other due to the properties of DCT transform and the fact that edge pattern is

extracted in terms of sub-blocks with 16 pixels in total. In other words, the corrupted pixels may not necessarily lead to wrong edge pattern extraction.

V. CONCLUSION

In this paper, we proposed an effective content descriptor for both JPEG compressed images and MPEG compressed videos. In comparison with existing techniques, the proposed algorithm illustrates the following advantages: 1) robust to a range of orientation changes which are frequently encountered during transmission, processing, editing and storages; 2) directly operates in compressed domain, which are suitable for processing large compressed image or video databases; and 3) extremely low computing cost with RLEBH (3) suitable for real-time implementation and fast image or video content management applications. As a result, the proposed algorithm would have significant potential for practical applications either as a stand-alone tool or as a component working with other tools towards visual content management and descriptions.

Possible further research could consider RL of edge pattern states rather than individual edge patterns. One example of such edge pattern state could be constructed as joining edges with connected vertical and horizontal patterns across multiple rows or columns. Other possibilities could include some popular elements inside contours, texture, or shapes, etc.

REFERENCES

- [1] S. F. Chang, "Compressed-domain techniques for image/video indexing and manipulation," in *Proc. Int. Conf. Image Process.*, 1995, pp. 314–317.
- [2] C. W. Ngo and T. C. Pong, "Exploiting image indexing techniques in DCT domain," *Pattern Recongn.*, vol. 34, pp. 1841–1851, 2001.
- [3] G. Feng and J. Jiang, "JPEG compressed image retrieval via statistical features," *Pattern Recongn.*, vol. 36, pp. 977–985, 2003.
- [4] D. G. Sim, H. K. Kim, and R. H. Park, "Fast texture description and retrieval of DCT-based compressed image," *Electron. Lett.*, vol. 37, no. 1, pp. 18–19, 2001.
- [5] J. H. Liu and H. M. Gu, "Image retrieval in various domains," *Comput. Graph.*, vol. 27, pp. 807–812, 2003.
- [6] M. Coimbra and M. Davies, "Approximating optical flow within the MPEG-2 compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 103–107, Jan. 2005.
- [7] S. Soltane, N. Kerkeni, and J. C. Angue, "The use of two dimensional discrete cosine transform for an adaptive approach to image segmentation," in *Proc. SPIE Image Video Process.*, 1996, vol. IV, pp. 242–251.
- [8] J. W. Han and L. Guo, "A shape-based image retrieval method using salient edges, signal processing," *Image communication* 18, pp. 141–156, 2003.
- [9] M. Banerjee and M. K. Kundu, "Edge based features for content-based image retrieval," *Pattern Recongn.*, vol. 36, pp. 2649–2661, Nov. 2003.
- [10] D. S. Kim and S. U. Lee, "Image vector quantizer based on a classification in the DCT domain," *IEEE Trans. Commun.*, vol. 39, no. 4, pp. 549–556, Apr. 1991.
- [11] B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," in *Proc. SPIE: Storage Retrieval for Still Image and Video Databases IV*, Mar. 1996, vol. 2670, pp. 404–414.
- [12] H. S. Chang and K. Kang, "A compressed-domain scheme for classifying block edge patterns," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 145–151, Feb. 2005.
- [13] H. Li, G. Liu, and Y. Li, "An effective approach to edge classification from DCT domain," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2002, pp. 940–943.
- [14] J. Jiang, "A low cost content adaptive and rate controllable near lossless image codec in DPCM domain," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 543–554, Apr. 2000.
- [15] G. Theo and W. M. Arnold, "PicToSeek combining color and shape invariant features for image retrieval," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 102–119, Jan. 2000.
- [16] H. J. Bartsch, *Handbook of Mathematical Formulas*. Reading, MA: Academic, 1974.
- [17] I. Popovici and W. D. Withers, "Custom-built moments for edge location," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 4, pp. 637–642, Apr. 2006.