A Review on Various Image Compression Methods in Content Based Image Retrieval

Vandana Vinayak, Sonika Jindal

Abstract: This paper provides an overview about the various compression techniques available in the research area of Image retrieval, especially Content-Based Image Retrieval (CBIR), an evocative and authentic research area for the last decades. CBIR is used for the retrieval of the images based on the content of the images generally known as features. These features may be low level features i.e. color, shape, texture and spatial relationship or the high level features that use the concept of human brain. Now a days, the development and demand of multimedia product grows increasingly fast, contributing to insufficient storage of memory device. Therefore, the theory of data compression becomes more and more significant for reducing the data redundancy to save more hardware space. Compression is the process of reducing the amount of data required to represent the quality of information. Compression is also useful as it helps to reduce the consumption of expensive resources such as hard disk space.

Keywords: Especially Content-Based Image Retrieval (CBIR), Therefore, increasingly fast, provides.

I. INTRODUCTION

As information technology proliferates throughout our society, digital images and video or visual objects are becoming as important as traditional textual based information. With the massive growth in the amount of visual information available, there exists a real need for systems to catalogue and provide retrieval from digital image and video libraries. Digital images are currently used in medicines, fashion, architecture, face recognition, finger print recognition and bio-metrics [1]. CBIR is a search engine used to retrieve the images from the large database according to the users query [2]. It is also known as query by image content (QBIC) and content-based visual information retrieval(CBVIR).CBIR mainly works in two phases- first phase related to feature extraction and another one is for similarity matching. Feature extraction mainly depends upon Low-level features or High-level feature. The low level features of an image are color, shape, texture and spatial relationship. On the other hand, high level features include semantic based image retrieval computed from text description or by complex algorithms of visual contents. The mixture of these content based features is required for better retrieval of image according to the application.

Manuscript published on 30 December 2016.

***** Correspondence Author (s)

Vandana Vinayak, Department of Computer Science & Engineering, Shaheed Bhagat Singh State Technical Campus, Ferozepur (Punjab)-152004, India.

Prof. Sonika Jindal, Department of Computer Science & Engineering, Shaheed Bhagat Singh State Technical Campus, Ferozepur (Punjab)-152004, India.

The most important challenge of CBIR system is to determine the exact and approximate matching image of database to the query image. Image Compression is the art and science of reducing the amount of data required to represent an image is one of the constructive and commercially booming technology in the field of image processing .the number of images that are compressed and decompressed day after day is surprising, and the compression and decompression themselves are almost imperceptible to the user. In compression mainly the redundant data is removed and only the required information is kept. Two dimensional intensity arrays of an image suffer mainly from three types of data redundancies that are Coding Redundancy, Spatial and temporal Redundancy and Irrelevant Information. Compression is achieved when one or more redundancies are removed. During the past decades, various compression methods have been developed to address major challenges faced by digital imaging. These compression methods can be classified broadly into lossy or lossless compression. Lossy compression can achieve a high compression ratio, 50:1 or higher, since it allows some acceptable degradation. Yet it cannot completely recover the original data. On the other hand, lossless compression can completely recover the original data but this reduces the compression ratio to around 2:1. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that producible differences may be called visually lossless.

II. IMAGE COMPRESSION

Image compression is an application of data compression that encodes the original image with few bits to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. The term data compression refers to the process of reducing the amount of data required to represent a given quality of information. Here data and information are not equivalent, data are the resources via information is conveyed. Sometime different amounts of data is utilized to symbolize the similar amount of information. These representations that enclose irrelevant or repeated information are known as redundant data. The relative data redundancy of an image can be calculated as –

$$R = 1 - \frac{1}{C} \qquad (1)$$

Where C, commonly known as Compression Ratio that can Defined as-

$$C = \frac{b}{b'} \qquad (2)$$

Here b denotes the number of bits or information carrying units of original image and, b' denotes the number of bits or information carrying units of compressed image

A. Data Redundancies

In digital Image Compression, an image is characterize by two-dimensional array of intensity values. these two-dimensional intensity arrays experience from three foremost varieties of data redundancies that can be acknowledged as 1) Coding Redundancy: A code is system of symbols like letters, numbers or bits that are used to represent a body of information or set of events. Each bit of information is consigned a sequence of code symbols known as code word. The number of symbols in each code word is its length. The eight-bit codes that are used to signify the intensities in most two-dimensional intensity arrays hold more bits than are required to represent the intensities.

To represent the intensities of M*N image, a discrete random variable *rk* is used in the interval of [0,L-1] and this random variable occur with the probability *pr (rk)* is as-

$$p_r(r_k) = \frac{n_k}{MN} \qquad (3)$$

and k = 0, 1, 2,...., L-1

Where L is number of intensity values and nk is the number of times that kth intensity appear in the image. If number of bits used to symbolize each value of *rk* is *l(rk)*, then the average number of bits required to represent each pixel is

$$L_{avg} = \Sigma l(r_k) p_r(r_k) \qquad (4)$$

The average length of code words allocated to various intensity values is set up by summing the products of number of bits used to represent each intensity and the probability that the intensity occur. The total number of bits required to represent an M*N image is *MNLavg*.

Coding redundancy occur when codes assigned to the set of intensity values do not take whole benefit of the probabilities of the event. It is almost always present when the intensities of an image are shown using natural binary code. Main reason for this is, most of the images are having objects that have regular and predictable shape and reflectance. A natural binary encoding assign the same number of bits to the both, most and least probable values, failing to minimize the average value and resultimg in coding redundancy.

2) Spatial and Temporal Redundancy: As the pixels of an image in 2-D intensity arrays are interrelated spatially that means each pixel is similar to its neighboring pixels or they are dependent on their neighboring pixels, information is gratuitously simulated in the representation of the pixels.

In most of the images, pixels are related to each other spatially. because most pixel intensities can be predicted well from their neighboring pixel's intensities, the information carried by single pixel is very less. Most of its visual involvement is superfluous in the sense that it can be inferred from its neighbors. To lessen the redundancy that unite with spatial and temporal intensity pixels, the 2-D intensity array must be transformed into more efficient representation. Various length measures can be used that can find the differences between adjacent pixels. This type of transformation is known as Mapping. A mapping can be reversible or irreversible. If the pixels of the original 2-D intensity array can be restructured from the transformed data set without any error, a mapping is said to be reversible otherwise a mapping is said to be irreversible.

3) Irrelevant Information: Mostly the 2-D intensity array of an image have some information that is ignored by human visual system, that information is irrelevant means it is of no use.

B. Image Compression Model

An image compression system consist of two functional components - an encoder and a decoder. Compression is done by encoder and its corresponding operation i.e. decompression is done by the decoder. There is also a device known as codec who can perform both the operations[3]. An input image f(x,...) is fed into the encoder and the output of this is a compressed representation. This output is stored for later use and when it is fed to decoder, a reformed output image f'(x,...) is generated. Sometimes the output image f'(x,...) of decoder may be the exact replica of input image f(x,...), then it is known as error free or lossless compression system. If it is not the exact replica and image is distorted, then it is known as lossy compression system.

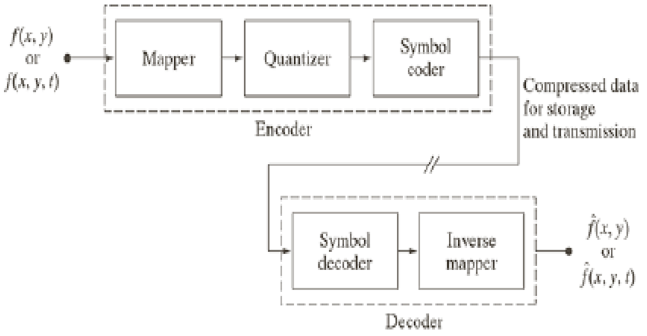


Fig. 1. Block Diagram of Image Compression Model

ENCODING PROCESS - The main task of encoder is to remove the redundancy through the series of independent operations. The first operation of encoder is done by the mapper. Mapper transforms the input image f(x,...) into the required format to eliminate the spatial and temporal redundancy. This operation is reversible and may or may not reduce directly the amount of data required to represent the image. Example of mapping is Run-length coding that on the whole acquiesce compression. In the next step quantizer reduces the accuracy of mapper's output in accordance with pre-establish fidelity criteria.

The objective is to keep inappropriate information away from compressed representation. Quantizer operation is irreversible and this can be omitted where lossless compression is preferred.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Next step in encoder is handle by symbol coder. Coder creates fixed or variable length codes and maps the output in accordance with the code. In several, variable length code is utilized. Operation done by Symbol coder is reversible.

DECODING PROCESS - The decoder consist of two components. The first one is Symbol decoder and the other is an inverse mapper. Both perform in reverse order of mapper and symbol coder. Quantizer block is not included as quantization results in irrreversible information loss.

III. COMPRESSION METHODS

A. *Huffman Coding*

Huffman coding is a technique which is used for removing the redundant coding or we can say a method for construction of minimum redundancy code. It was proposed by Dr. David A. Huffman in 1952 .Basically in image, the image pixels are redundant and the principle of Huffman coding is to use a lower number of bits to encode the data that occurs more frequently [4]. It yields the smallest possible number of code symbols per source symbols[5].It is form of statistical coding which attempts to reduce the amount of bits needed to represent a string of characters. Code book is used for storage of codes which may be constructed for each image or can be a set of images. For every case the code book plus encoded data must be transmitted to enable coding[6].The algorithm completes its aims by allowing characters or sign to vary in length. Tiny codes are assigned to the most frequently used symbols and longer codes to the symbols which appear less frequently in the string.

The Huffman Algorithm encodes messages using the following steps:

- Create a list of the symbols, sorted by frequencies from Largest to smallest.
- Combine the two smallest frequency values.
- Re-sort the frequency list. Repeat Steps 1, 2 and 3 until the all of the frequencies have been added up.

Example:

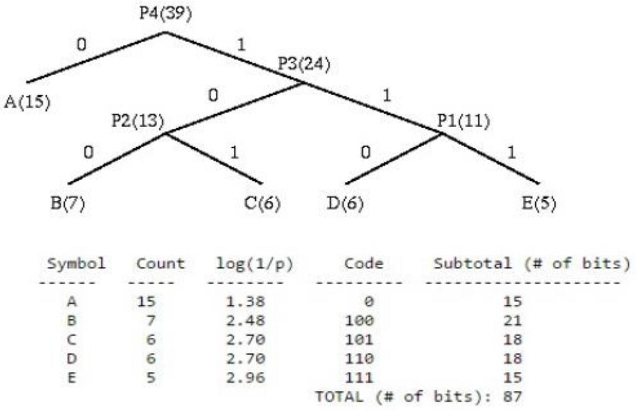
Initialization: Put all nodes in an OPEN list, keep it sorted all times.(ex. ABCDE).

2. Repeat until the OPEN list has only one node left.

a) From list OPEN pick two nodes having lowest frequencies; create a parent node of them.

b) Give the sum of children frequencies to parent node and insert it into OPEN list.

c) Assign code 1,0 to branches of the tree and delete children from OPEN

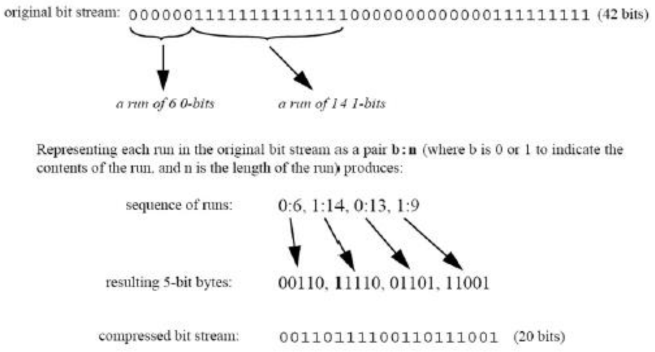


```
Symbol    Count    log(1/p)      Code       Subtotal (# of bits)
------    -----    --------    ---------    --------------------
  A        15        1.38          0                15
  B         7        2.48        100                21
  C         6        2.70        101                18
  D         6        2.70        110                18
  E         5        2.96        111                15
                               TOTAL (# of bits):   87
```

Fig. 2. Representation of Huffman Coding

B. *Run Length Coding*

Images with repeating intensities can be compressed by representing runs of identical intensities as run-length pairs, where each pair specifies the start of new intensity and number of consecutive pixels that have similar intensity. This technique referred as Run-Length Encoding (RLE). It was developed in 1950s. Compression is achieved by eliminating spatial redundancy groups of identical intensities. Run length coding is particularly effective for compressing binary images as there are only two possible intensities, adjacent pixels are more likely to be identical. Additional compression can be achieved by variable length coding the run lengths themselves.



original bit stream: 000000111111111111111000000000000001111111111 (42 bits)

Representing each run in the original bit stream as a pair b:n (where b is 0 or 1 to indicate the contents of the run, and n is the length of the run) produces:

sequence of runs: 0:6, 1:14, 0:13, 1:9

resulting 5-bit bytes: 00110, 11110, 01101, 11001

compressed bit stream: 00110111100110111001 (20 bits)

Fig. 3. Example of Run-Length Coding

The BMP file format uses two different representation modes: encoder and absolute. In encoded mode, two byte representation is used. The first byte specify number of consecutive pixels that have color index contained in second pixel. In absolute mode, first byte is zero and second is one of the following conditions:

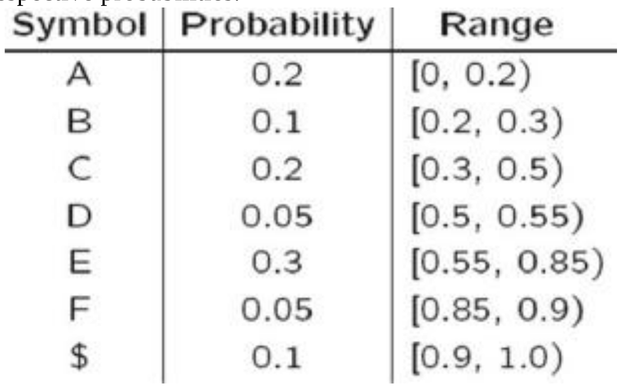
Second Byte Value	Condition
0	End of Line
1	End of Image
2	Move to a new position
3-255	Specify pixel individually

C. *Arithmetic Coding*

Arithmetic coding is a data compression technique that encodes data (the data string) by creating a code string which represents a fractional value on the number line between 0 and 1. Arithmetic coding maps a string of data symbols to a code string in such a way that the original data can be recovered from the code string. The encoding and decoding algorithms perform arithmetic operations on the code string.

One recursion of the algorithm handles one data symbol [7]. It generates non-block codes. The code word itself defines an interval of real numbers between 0 and 1. As the number of symbols in the message increases, the interval used to represent it become smaller and number of information units represent the interval become larger. Consider an example where a six-symbol sequence *A;B;C;D;E; F* is coded and sequence is having one special character.

At the start of coding process, the message is assumed to occupy the entire half-open interval [0,1) [8]. The table represents the interval is subdivided into regions based on probabilities of each source symbol. Symbol A is associate with sub-interval [0,0.2) as it is the first symbol of message that is being coded. Similarly each symbol of sequence is associated with different intervals according to their respective probabilities.



Symbol	Probability	Range
A	0.2	[0, 0.2)
B	0.1	[0.2, 0.3)
C	0.2	[0.3, 0.5)
D	0.05	[0.5, 0.55)
E	0.3	[0.55, 0.85)
F	0.05	[0.85, 0.9)
$	0.1	[0.9, 1.0)

Fig. 4. Example of Arithmetic Coding

The intervals of the symbols are expanded to the full height and the end points labeled by values of the narrowed range. The narrowed range is then subdivided in accordance with original source symbol probabilities and the process continues with next symbol.

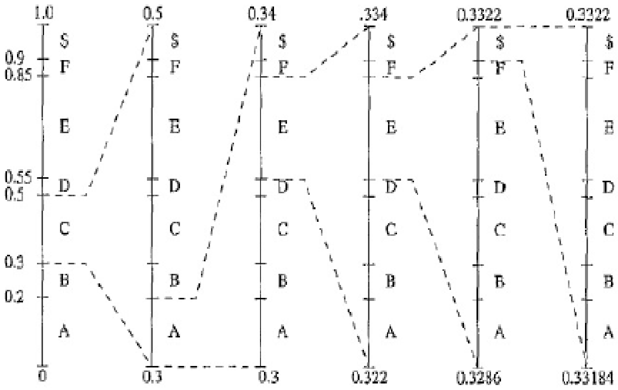


Fig. 5. Graphical display of ranges

D. Symbol Based Coding

In Symbol based coding, an image is represented as a collection of frequently occurring sub-images called symbols. Each symbol is stored in symbol dictionary and image is coded as a set of triplet (x1; y1; t1), where (x, y) pair specify the location of symbol in an image and token t is the address of symbol in the dictionary. Storing repeated symbols only once can compress the images [9].

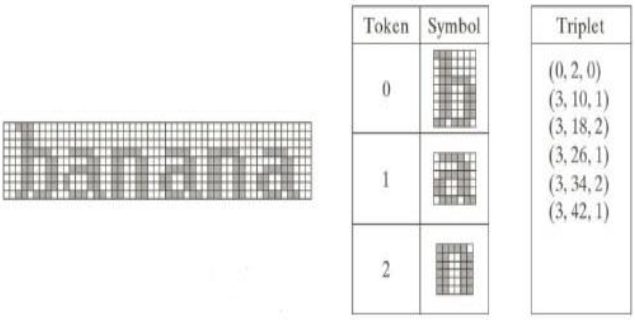


Fig.6. Example of Symbol-based Coding

Consider an example of single word banana that is a combination of unique three symbols: b,a and n. Here b is the first symbol identified in coding process, its bitmap is stored in location 0 in symbol dictionary. The first triplet in encoded image representation is (0, 2, 0). Similarly the process continues till all the symbols are stored in the symbol dictionary. To decode the symbol based representation, simply read the bitmaps of symbols specified in triplet in the dictionary and place them at spatial coordinates specified in each triplet. Advances in matching algorithm and increased computer processing made it possible to select dictionary symbols and find where they occur in timely manner. As only exact symbol matches are allowed, the resulting compression is loss-less.

E. Block Truncation Coding

Block truncation coding (BTC) is a simple and fast lossy compression technique for digitized gray scale images.[10]The block truncation coding algorithm is a simple, block-based and spatial domain. This compression technique is developed by Delp and Mitchell. As the amount of image data increase day by day, large storage and bandwidth are needed to store and transmit the images, which is quite costly [12].The key idea of BTC is to perform moment preserving (MP) quantization for blocks of pixels so that the quality of the image will remain acceptable and at the same time the demand for the storage space will decrease. The truncated block of the BTC is the one bit output of the quantizer for every pixel in the block.

Block Truncation Coding divides the image into small no overlapping blocks of equal size and process these blocks independently.[9] It is a reversible and linear transform used to map each sub image into transform coefficients, which are later quantized and coded. Consider an input image of size M * N ,it is first divided in sub-images of size n*n, which are then transformed to generate MN=n2 subimage transform array. The goal of transformation process is to decor relate the pixel of each sub-image. The quantizer then eliminates the coefficients that carry least amount of information. The decoder implements the inverse sequence of steps of the encoder and decompress the image.

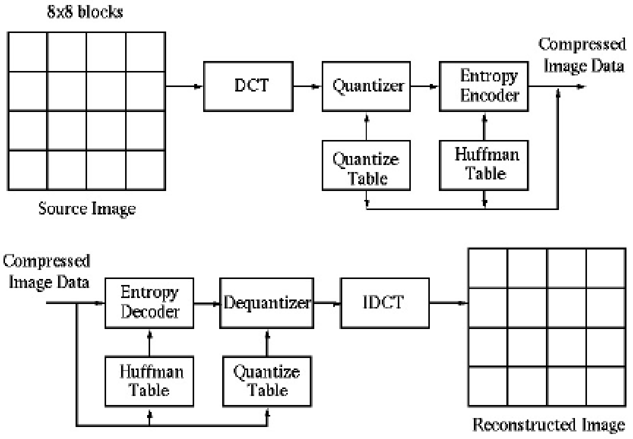


Fig. 7. Block Truncation Coding

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

1) Transform Selection: Block Truncation Coding system based on variety of transforms that have been constructed. The choice of a particular transform in an application depends on amount of reconstruction error that can be tolerated and computational resources available. Compression is achieved during the quantization of transformed coefficients. A simple transformation that is useful in transform coding is Walsh Hadamard Transform (WHT). Other useful transformations are Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT).

2) Sub Image Size Selection: Second factor affecting transform coding error and computational complexity is sub-image size. Images are subdivided so that redundancy between adjacent sub-images is reduced to some level. As the sub-image size increases, level of compression and computational complexity also increases. The regular sub-image size of an image is 8*8 and 16*16.

3) Bit Allocation: The reconstruction error is a function of the number and relative importance of transform coefficients that are discarded, as well as precision that is used to represent the retained coefficient. The retained coefficients are selected on the basis of maximum variance called zonal coding and on basis of maximum magnitude is called threshold coding. The overall Process of truncating, quantizing and coding the coefficients of transformed sub-image is known as bit allocation.

IV. CONCLUSION

The principal objective of this paper is to present an overview about digital image compression and to describe most commonly used compression methods. Here compression refers to the process of reducing the amount of data required to represent a given quality of information. It plays a key role in document image storage and transmission, the internet and commercial video distribution etc. Compression can be achieved by removing some redundancies like data redundancy, code redundancy, spatial and temporal redundancy etc. Various compression methods are described to compress digital images but the level of presentation is introductory in nature. Huffman coding is used for various file formats like CCITT, JBIG-2, MPEG-1,2,4 and for fixed value of source symbol. Arithmetic coding performs better for JPEG 2000 image format. Run-length coding removes spatial redundancy and generate better output for BMP images rather than TIFF format. Block truncation coding performs better than all other methods as it is very fast encoding method, easy to implement. Standard BTC involves less computational complexity and requires little memory space.

REFERENCES

1. J. O. A. Tamer Mehyar, “An enhancement on content based image retrieval using color and texture features,” vol. 3, no. 4. Journal of Emerging Trends in Computing and Information Sciences, April 2012.
2. S. J. Nitika Sharma, “A review on global features based cbir system.” International Conference on information and mathematical sciences.
3. V. Tcheslavski, “Basic image compression methods,” 2008.
4. M. Sharma, “Compression using huffman coding,” vol. 10, no. 5. IJCSNS International Journal of Computer Science and Network Security, May 2010.
5. K. S. Julie Zelenski, “Huffman encoding and data compression.” Springer 2012, CS106B, May 23 2012.
6. D. S. Mridul Kumar Mathur, Seema Loonker, “Lossless Huffman coding technique for image compression and reconstruction using binary trees,” vol. Vol 3 (1). International Journal of Computer Technical Applications, pp. 76–79.
7. J. Glen G. Langdon, “An introduction to arithmetic coding,” vol. 28, no. 2. IBM J. RES. DEVELOP., March 1984.
8. P. P. Venkataram, Lossless Compression Algorithms, 2016, ch. 6.
9. R. E. W. Rafael C. Gonzalez, Digital Image Processing, 3rd ed. Pearson Education, 2014.
10. O. N. Pasi Franti and T. Kaukoranta, “Compression of digital images by block truncation coding:a survey,” no. 37(4). The Computer Journal, 1994, pp. 308–332.
11. P. Jing-Ming Guo and J.-H. Chen, “Content-based image retrieval using error diffusion block truncation coding features,” vol. 25, no. 03. IEEE Transactions On Circuits And Systems For Video Technology, 2015.