The University Of Sheffield.

Compressed Domain Low Level Visual Descriptors

Zhizhong Yu

Department of Electronic and Electrical Engineering,

The University of Sheffield, Sheffield, United Kingdom

Master of Philosophy

July 7, 2015

Abstract

Content-based image retrieval and analysis have been developed for a long time, and various visual descriptors have been proposed. The need of multiple versions of an image spurs the development of image compression and descriptors based on compression domain. However, these descriptors are not able to achieve good performance in terms of quality and resolution scalability. As the appearance of JPEG 2000 compression standard, its coding algorithm and structure of bit stream make the scalability possible. The JPEG 2000 based descriptors can be developed to satisfy multiple compression levels, and keep a good performance even when the images are highly compressed. In this thesis, most existing famous and popular low level visual descriptors are reviewed. Image compression and some image analysis and retrieval approaches are introduced. Two JPEG 2000 based descriptors called state and context are proposed in this research, and an image retrieval system using these descriptors are constructed. Experiments are conducted and the results indicate the proposed descriptors have a good retrieval performance. State and context are further compared with industrial standard MPEG-7 descriptors and state-of-art SIFT method in multiple resolution and quality situations, and the proposed descriptors are proved to be more suitable in compression domain.

Acknowledgements

I wish to thank my supervisor Dr Charith Abhayaratne for his supervision and guidance throughout my whole research study, and his understanding and kindness after I started to have health problems as without his suggestion and help I would not be able to finish my research. I also want to thank my parents who enlightened me in my childhood and provided me an opportunity of good education so that I am able to achieve so much today. When I was ill, it was my parents who accompanied me to visit different hospitals from place to place for medical treatment. I am grateful to them for their understanding and support when I decided to give up my PhD for health reason. My gratitude also goes to my friends and everyone in the VIE lab. Finally, I would like to express my sincere gratitude to my wife for her support and help. Without her care and encouragement, I would not be able to finish my research.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

As the development of technology, the cost of taking and uploading pictures has dropped significantly. For example, the price of digital camera has dropped and mobile phones have been equipped with better cameras, so more and more people upload their photos into the Internet. As a result, enormous amounts of pictures are generated every minute, and tens of thousands of images are uploaded into various databases or transmitted between devices and servers each day. The management of huge image database and analysis of these large data have become a huge challenge. Many methods have been developed to solve this problem so far, and more new approaches are still emerging.

1.1 Motivations

Visual content retrieval is a very important part of visual management and analysis. The idea of image retrieval is to find out the matching images in the target database for a query image. The early stage of image retrieval started with text-based image annotation which was proposed in late 1970s [2]. Images were annotated by text and then were stored in database with a text-based label. When it came to retrieval, query images text-based label was compared with all the other labels in the database. However, as it required manual annotation and human perception of image is actually subjective, this method is not suitable

for numerous huge databases any more.

The concept of content-based image retrieval (CBIR) was then proposed. Images are indexed by visual contents instead of key words [3]. Various features have emerged ever since, and they are categorized in many ways, such as colour, texture or shape [4]. A large number of researches have been conducted, and many descriptors have been developed. The emergence of MPEG-7 standard brings series of new features providing excellent retrieval results, and then they became the industrial standard[5]. These descriptors can be separated from the image content and make the management of images more efficient. Furthermore, the development of new methods is always rapid and more approaches are springing up ever since, such as SIFT[2, 6].

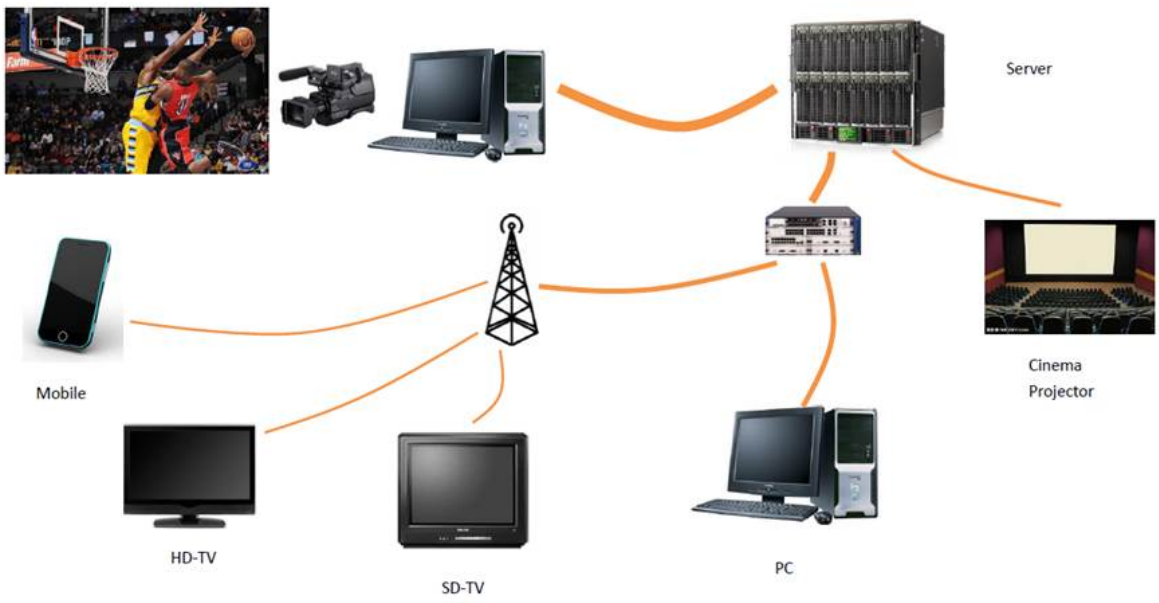

Figure 1.1: Visual contents need to be adapted to cater different needs.

As the development of techniques, however, there is another problem appearing for image analysis and retrieval, and that is visual contents need to be adapted to cater different needs.

Figure 2.4 gives a very good example. A basketball match video or pictures, for example, is taken and uploaded to the server. According to different needs from various users, these visual content needs to be compressed to have multiple resolution-quality versions. HD-TV user wants a high quality program, and compression of video must be kept in a very low level; the mobile user, in the opposite, cares about the speed and fluency, and a high compression rate can achieve this. High resolution and high quality images contain more detail, while low resolution and less quality images take less bandwidth when transmitting. Many image compression standards have appeared like JPEG [7].

Actually, images are almost stored and transmitted in compressed form nowadays [8]. As most CBIR techniques operate in the pixel domain, nevertheless, images needed to be decompressed into the pixel domain before feature extraction which leads to a computational overhead. An alternative is to extract features directly in compressed domain. Figure 2.4 is a clear illustration of the advantage of compressed domain feature extraction. There is no need of decoding and decompression, and many steps are saved compared with pixel domain methods. Feature vectors are extracted straight from the compressed data, which much computation and time.

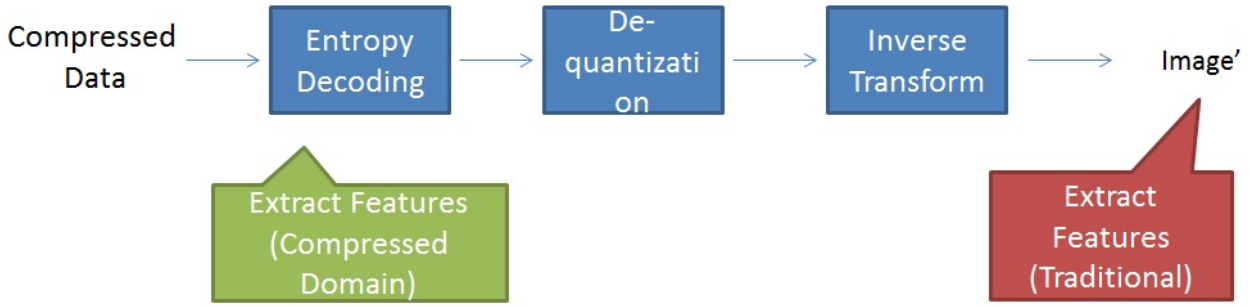

Figure 1.2: The advantage of compressed domain feature extraction.

There is still one more problem about scalable coding. In traditional image compression, compressed images need to be decompressed and re-encoded to obtain different versions. Most standards are not able to achieve both resolution and quality scalability. Fortunately, the

emergence of scalable coding solves the problem without re-encoding. Images can be encoded in the same way, but decoded in many ways. The typical new compression standards is JPEG 2000 [9]. JPEG 2000 is a new compression standard with excellent performance. Two of its advantages are quality scalability and multiple resolution support. It provides a choice of lossy and lossless compression. It has significant performance at low bit-rate. The region-of-interest and error resilience are its advantages as well. These features are achieved mainly by using of wavelet transform and embedded analysis of image coding.

This research focus on developing new low-level visual descriptors based on JPEG 2000 standard to provide a stable and robust retrieval system that has the advantages of scalable coding and are resolution and quality scalability.

1.2 Aims and Objects

The main aim of this research is to investigate low-level visual descriptors in compressed domain that is invariant to resolution and quality, and then to exploit coding modes on describing images.

Objective 1: To study and review state-or-art visual descriptors. Rather than high-level semantic descriptors [2], low-level visual descriptors are investigated in this research. The recent MPEG-7 descriptors [10] are presented and discussed particularly as this standard is famous and proved to performance well. Other techniques related to image retrieval review briefly. At last, a brief discussion of image compression and descriptors in compressed domain are discussed.

Objective 2: To research and develop new descriptors based on JPEG 2000 coding scheme. In order to construct a new JPEG 2000 based descriptor that is robust to quality and resolution variation, it is important to investigate JPEG 2000 coding mechanism. New descriptors are designed and developed based on it.

Objective 3: To build up a retrieval system with proposed new descriptors and conduct practical retrieval. The proposed descriptors are extracted from the code stream and feature

vectors are organized and stored for similarity measurement.

Objective 4: Performance evaluation. The proposed method is evaluated with the giving database under the circumstance of variant quality and resolution. Its robustness against quality and resolution variation is tested. The proposed descriptors are further compared with state-of-art SIFT method and industrial standard MPEG-7 descriptors.

1.3 Work programmes and Contributions

The state-of-art visual descriptors are reviewed and discussed in this research. The discussion ranges from basic categories[4] like colour, texture and shape to latest popular methods like MPEG-7 and SIFT. Reviews are also made on machine learning, image compression and descriptors in compressed domain.

Two new descriptors, i.e., state descriptor and context descriptor, are developed based on JPEG 2000 standard. A retrieval system is set up with proposed descriptors for practical image retrieval. Both state and context descriptors have a good and stable retrieval results and are robust against compression. Their retrieval performances are compared with state-of-art MPEG-7 descriptors, and the proposed descriptors outperform in terms of resolution and quality scalability. The proposed descriptors outperform state of art SIFT method in high, medium and low resolution and quality levels.

Unlike most descriptors that need to be decompressed and re-encoded to generate different versions, the proposed descriptors only need to be encoded once and the bit stream contains structural information that are able to generate any version images according to different requirements. It is a huge improvement as significant computation, time and storage are saved compared with traditional approaches.

The bit-planes of compressed image are further investigated and analysed in this research. With the proposed descriptors, most bit-planes can be discarded to achieve a relevant acceptable retrieval performance, which means the cost of bandwidth can be saved dramatically with a limited sacrifice of image quality.

1.4 Thesis Organisation

The remainder of the thesis is structured into four chapters as following:

In Chapter 2, the state-of-art visual descriptors are reviewed and discussed. Most of popular methods of colour, texture and shape descriptors are explained in detail and the MPEG-7 features are presented as well. Finally, some other advanced approaches are introduced.

Chapter 3 investigates the core block coding system of JPEG 2000 compression standard. Further, two novel descriptors, data descriptor and context descriptor, are developed based on the JPEG 2000 entropy coding scheme.

Chapter 4 introduces the development of a new retrieval system based on the two novel descriptors proposed in Chapter 3. Feature extraction, similarity measurement and further works are discussed. Experiment work are conducted with this retrieval system and the performance are compared with MPEG-7 and SIFT descriptors.

Finally, Chapter 5 concludes the whole thesis. The possible future work are suggested.

Chapter 2

Literature Review

Many efforts have been made to contribute the development of image retrieval. In the early years, the retrieval was based on the annotation like key words or labels. Various approaches have been proposed ever since. This chapter presents and discusses the state-of-art colour, texture and shape descriptors. The latest MPEG-7 standard and other new features like SIFT are analysed. Machine learning is introduced as it is usually used in image retrieval. At last, image compression and descriptor in compressed domain are investigated.

2.1 Colour Descriptors

Some of the most famous colour descriptors are presented in this section. The introduction starts with the most common colour histogram, and goes onto its variations. The MPEG-7 colour descriptors are presented and discussed at the end of this section.

2.1.1 Colour Histogram

Colour histogram [11, 12] describes the distribution of colours in the image. It is widely applicable to various colour space like RGB. First, each channel of the colour is discretized into N colour arrays (bins).The whole image is then scanned pixel by pixel to count the

number of pixels in each bin. The advantages of colour histogram are simple computation and invariance to rotation and translation. However, this method is lack of spatial information which means two totally different images might have similar colour histogram only because their colour distribution are alike. Other problems are it might have very high dimension and it is sensitive to noise.

There are a few distance formulas for calculating the histogram similarity ratings. However, these distances are not able to provide the practical similarity of two images. They do not work until being compared with other similarity distances. Some of the popular distance are histogram Euclidean distance, histogram intersection distance and histogram quadratic distance.

Histogram Euclidean distance

H and G are a pair of histograms, and each of them contains N bins. The histogram Euclidean distance is computed as

$$d^2(H,G) = \sum_{i=1}^{N}(H_i - G_i)^2. \tag{2.1}$$

Histogram intersection distance

The intersection distance of histogram H and G is defined to be

$$d(H,G) = \frac{\sum_{i=1}^{N} min(H_i, G_i)}{min(\sum_{i=1}^{N} H_i, \sum_{i=1}^{N} G_i)}. \tag{2.2}$$

This method reduces the distraction of pixels in the background as this pixel needs to have the same colour as one of colours presented in query image. It is robust to viewpoint variation, occlusion and varying image resolution. However, its lack of spatial information might lead to different images have similar histograms.

Histogram quadratic distance

The quadratic distance [13, 14] between histogram H and G is given by

$$d(H,G) = \sum_{i=1}^{N}\sum_{j=1}^{N}(H_i - G_i)\cdot a_{ij}\cdot(H_j - G_j), \tag{2.3}$$

where a_{ij} is the cross-correlation between histogram bins based on human perceptual similarity of the colour i and the colour j. One of the reasonable value of a_{ij} with regard to d_{ij}, and $a_{ij} = 1 - d_{ij}/d_{max}$, where d_{ij} is the L_2 distance between colours i and j.

2.1.2 Colour Moments

Another colour feature is called colour moments which was proposed by Stricker and Orengo in [15]. The authors bring the probability distribution idea in mathematics into the feature extraction. Three moments are calculated and they are mean, variance and skewness as following:

$$E_i = \sum_{j=1}^{N}\frac{1}{N}P_{ij}; \tag{2.4}$$

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^2}; \tag{2.5}$$

$$s_i = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^3}; \tag{2.6}$$

where i means the i^{th} colour component(channel), and j means the j^{th} pixel of the image. Thus, p_{ij} is the j^{th} pixel in the i^{th} colour component. For each r^{th} component of the image, three moments are calculated. In order to compare colour distributions of two images, the mean, variance, and the skewness of two distributions are compared. One of moment can have more contribution than others depending on different situations.

Colour moments are compact and fast features as only first three moments of distribution are stored instead of a great many information in other features. They are useful when it comes to image of object or region [4]. However, there is no spatial information and not all the colours in the image are represented.

2.1.3 Colour Coherence Vector

As colour histogram is lack of spatial information, a new method is proposed by Greg Pass Ramin Zabih in [16] [17]. Pixels are considered coherent when they belong to a region with pixels larger than certain size.

To build up the colour coherence vector, the whole image is blurred first. Each pixel value is replaced by the average values of its eight adjacent neighbours. The colorspace is then discretized. The next step is to decide if pixels belong to a colour bucket. A pixel is coherent if it belongs to a group of connected pixels of a certain size. Those connected pixels in the group have to be the same colour(same discretized colour) and adjacent. A pixel is adjacent to its horizontal, vertical and diagonal neighbours, which means it should have eight neighbours in theory. Moreover, the size of this group of pixels has to be larger than a fixed value τ. Other pixels are considered incoherent. For the j^{th} discretized colour, the number of conherent pixels is α_j and the number of incoherent ones are β_j. As the sum of α_j and β_j $(\alpha_j + \beta_j)$ is the number of pixels of the j^{th} colour, the colour histogram is given as $\langle \alpha_1 + \beta_1, \alpha_2 + \beta_2, ..., \alpha_n + \beta_n \rangle$.

For the j^{th} discretized colour, a coherence pair is defined as (α_i, β_j). Thus, the colour coherence vector is given as $\langle (\alpha_1, \beta_1), (\alpha_2, \beta_2), ..., (\alpha_n, \beta_n) \rangle$. Colour coherence vector is a combination of colour histogram and spatial information, which is better than traditional colour histogram methods. However, the calculation of colour coherence vector costs huge computation. Its dimension is large as well.

2.1.4 MPEG-7 Colour Descriptors

Colour descriptors played an important role in the development of MPEG-7 descriptors [5]. There are various methods developed in different colour space, like scalable colour descriptor (SCD, colour structure descriptor, or dominant colour. The SCD is set in hue-saturation-value (HSV) space, whereas the CSD is defined in hue-min-max-difference (HMMD) colour space.

Scalable Colour Descriptor(SCD)

Scalable colour descriptor [5] is one of MPEG-7 descriptors. It is colour histogram based feature in HSV colour space. The advantage of scalability makes it better than traditional colour histogram.

The HSV values are quantized to 256 bins with 16 levels in H, 4 levels in S, and another 4 levels in V. They are then truncated and non-linearly mapped into 4-bit integer representation, and a Haar transform is implemented. As the basic unit of Haar transform are a sum operation and difference operation, which are related to primitive low pass filter and high pass filter, summing pairs of adjacent bins generates a histogram with half number of original bins. If this process is iterated time and time again, a histogram of 128, 64, 32, ... bins are generated. The high pass coefficients can be truncated to an integer representation with a low number of bits, as the differences of high pass coefficients between adjacent histogram bins are quite small.

The scalability is able to be achieved in two ways. The first way is based on the multi resolution of the Haar transform, the scalability depends on how many bins are used. The other way is applied on the difference coefficients of the high pass filter. Those coefficients are consist of sign part and magnitude part. One bit must be used to store the sign part, while the scalability is achieved by discarding the least significant bits of the magnitude part.

The similarity can be measured in either histogram domain or Haar transform domain. L1 norm distance with Haar transform domain is recommended for measurement. When all the magnitude are discarded with only sign bit left, the Hamming distance is applied. However, such way of scalability reduce the accuracy of retrieval result [4]. SCD still has no spatial

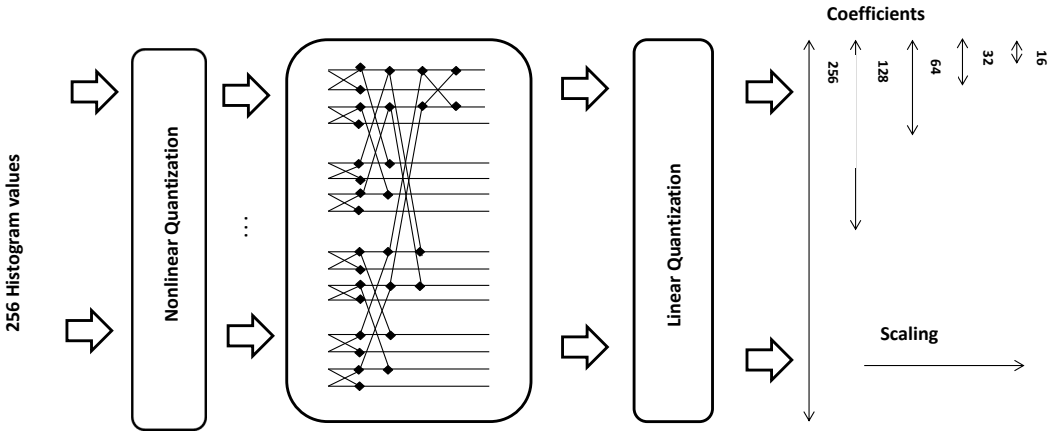

Figure 2.1: SCD

information like the conventional histogram.

Colour Structure Descriptor(CSD)

Another MPEG-7 descriptor similar to colour histogram is colour structure descriptor [5], while the difference is it contains spatial information. This descriptor is based on HMMD colour space, where HMMD is a new colour space supported in MPEG-7 [5].A structuring element of size 8×8 is used to scan the image. The structuring element is actually a sliding window shifts through the whole image and starts from the top left of the image.

The number of bins of the structuring element is decided by the number of quantized colour M. The bin value $h(m)$ is the number of pixels of colour m appear in the current structuring element. As the structuring element slide the whole image, the values of bins accumulate. Therefore, the final value of each bin $h(m)$ contains position information of colour m. L1 norm

distance is employed to calculate the similarity between histograms. The size of structuring element decides the performance of retrieval, and it becomes conventional colour histogram when the size is 1×1. The computation of CSD is large. Moreover, rotation and noise affect the performance.

Colour Layout Descriptor(CLD)

Colour layout descriptor is a compact descriptor based on YCbCr colour space. An image is divided into $8 \times 8(64)$ blocks. A representative colour is selected from each block, which leads to a 8×8 image. The average colour of all the pixels with a block is recommended to be representative colour. A 8×8 DCT is performed and derived average colours are transformed into coefficients. Colour layout descriptors are generated by choosing a few zigzag scanned and quantized low frequency coefficients. The distance measurement of two descriptors,$\{X, Y, Z\}$ and $\{X', Y', Z'\}$, is implemented as

$$d = \sqrt{\sum_i w_{yi}(X_i - X_i^{'})^2} + \sqrt{\sum_i w_{cbi}(Y_i - Y_i^{'})^2} + \sqrt{\sum_i w_{cri}(Z_i - Z_i^{'})^2}, \tag{2.7}$$

where $\{X, Y, Z\}$ is DCT coefficients of colour component Y, Cb and Cr, respectively. The weights of low frequency are larger as they need to contribute more. A bit-size of 63 is recommended, including 6 Y coefficients, 3 Cb coefficients and 3 Cr coefficients.

CLD is developed for content filtering using image indexing and sketch based image retrieval [5]. It is a computation-saving and high speed matching method. CLD is a grid-based dominant colour descriptor and is able to adapt multiple resolutions.

2.2 Texture Descriptors

There are many different methods to describe texture. They are generally classified into structural, statistical, model based and spectral approaches. The MPEG-7 shape descriptors are introduced at the end of this section.

2.2.1 Structural Texture Features

Structural method describes textures with texture primitives [4, 18]. The primitives are some texon or texture elements. Another thing needs to be defined is the placement rule of primitives. Thus, the analysis of texture are replaced by the analysis on the texture elements. However, it is very hard to define primitives for many images in real world.

2.2.2 Statistical Texture Features

Statistical methods are different from structural approaches. Rather than concentrating on structural elements of the texture, statistical methods represent texture in a distributional point of view. These methods are based on grey level images and are more robust than structural approaches. Some of the most popular features are introduced below:

Tamura Texture features

One famous method of statistical texture analysis is Tamura texture features [19]. This method consists of six different features. They are coarseness, contrast, directionality, line-likeness, regularity and roughness. This method has practical meaning as they are built based on human visual perception. Coarseness, contrast and directionality are presented as they are the most important features.

In order to get coarseness texture feature, the first thing is to calculate the average $A_k(x, y)$ of a $2^k \times 2^k$ window whose center is at location (x, y) for every pixel. For each location (x, y), then, the difference of two non-overlap window averages at location $(x + 2^{k-1}, y)$ and $(x - 2^{k-1}, y)$ are calculated in horizontal and vertical direction. The next step is to find out the specific k which gives the largest E_{max} in either horizontal or vertical direction, and the best size for pixel (x, y) is given as $S_{best}(x, y) = 2^k$. Finally, the coarseness of the whole $M \times N$ image are calculated by taking the average of all the S_{best}. The definition of contrast texture feature is decided by the standard deviation, and the forth moment about the mean. The directionality texture feature is extracted by calculating the square of difference of area between valleys and peak.

Grey Level Co-occurrence Matrix (GLCM)

Another statistical method of texture descriptor is called grey level co-occurrence matrix [20]. This method describe the frequency of combinations of different grey level in a image. GLCM texture descriptor considers the connection between two pixels. One pixel is the reference pixel and the other one is one neighbour (There are eight neighbours, but only one of them is chosen). A $n \times n$ matrix is set up. For example, if grey value are ranged from 0 to 255, then the size of this matrix is 256×256, where the first 256 are the reference pixel values and the second 256 are the neighbour pixel values. The given image is then scanned from the top left pixel, line by line, to right bottom one. The pixel are being scanned currently is the reference pixel. The match of its grey value and its chosen neighbour's grey value are counted, and their relative position in the matrix is added by 1.

Another $n \times n$ matrix is set up in the same way, but with a neighbour pixel of a opposite direction. For instance, the previous chosen direction is the north neighbour, then current direction must be south. The given image is then scanned and matches are recorded in the same way. The next step is to add these two matrices and a symmetrical matrix is generated. Finally, the symmetrical matrix is normalized to get probabilities.

One feature of GLCM is about lines paralleling to the diagonal. The closer they are to the diagonal, the smaller the image contrast is. A few features can be extracted based on GLCM, namely contrast and homogeneity. Contrast presents the difference of certain pixel and its neighbouring pixels, while homogeneity value is small when the difference over the whole image is not obvious. There are some more features like entropy, energy and correlation. This method is robust and compact [4]. However, GLCM is still limited, and it is not able to represent many other textures. Another drawback is it requires huge computation.

2.2.3 Model Based Texture Features

Autoregressive (AR) model

The autoregressive model is based on autoregressive model theory in mathematics. Un-

known parameters and values can be estimated by known values. In [18], this method is used to achieve unsupervised texture segmentation. The autoregressive model is define as below:

$$x_n = \sum_{i \in N_n} \beta_i x_i + \varepsilon_n, \tag{2.8}$$

where x_n is the pixel value at location n, and ε_n is the noise. N_n is the neighbours of n, and β is the distribution parameters. Autoregressive model is not sensitive to rotation [4, 18]. However, how to decide the pattern size is a difficult question and it quires vast computation.

Markov random field (MRF) model

The other famous texture model is Markov random field [18, 21]. This method is based on a concept that the value of a pixel is dependent on its neighbourhood. This idea starts with Markov Chains that every future state of a sequence of variables is only dependent on the present state. The first order Markov Figure 2.2 chain and the conditional probability is given as below:

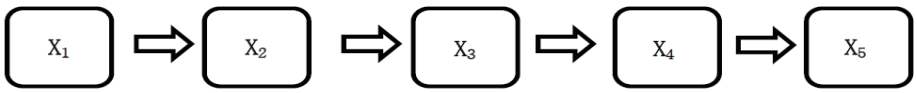

Figure 2.2: First order Markov chain

Similarly, when it comes to two dimensions, Markov random filed is illustrated in Figure 2.3 where the relationship between different order models and the range of chosen neighbouring pixels are presented.

If it is first order model, for example, the value of N is the sum of weighted four neighbours. MRF is used to estimate the natural texture, and the parameters are measured for texture generation in [21]. Textures like grass, tile, sand have good results. In [22], the author investigated MRF on small samples. When the box used smaller than 20×20, the estimation of real texture are not good.

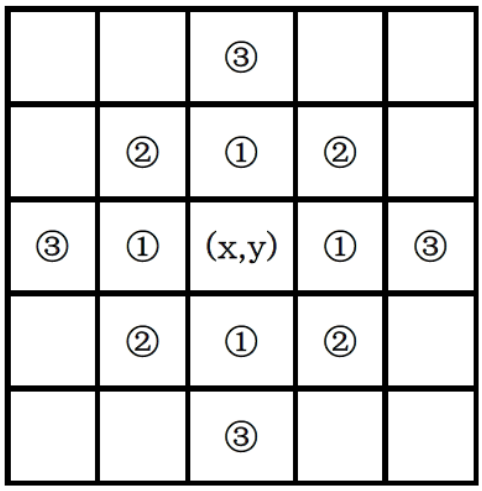

Figure 2.3: Neighbours of pixel(x, y), where① are the first order, ①+② are the second order, and ① + ② + ③ are the third order.

Fractal Dimension (FD) model

Fractal dimension is a very famous method. The concept of fractal is that it has the characteristic of self-similarity, and the smaller parts consist of it are similar to itself. When it comes to image analysis, fractal dimension indicate the coarseness of the image. A famous Differential Box Counting (DBC) method was proposed in [23], where images are considered a curve in 3D space.

A $M \times M$ image is divided into grids of the size $m \times m$. The relationships among them are $1 \leq m \leq M/2$ and $r = m/M$. The third dimension of this 3D space is the grey level of pixels. There is a column of boxs on each grid, and the size of a box is $m \times m \times m$. Let the maximum grey level falls into the l^{th} box and the miniumim grey level falls into the k^{th} box, which are located on the $(i, j)^{th}$grid. Then this grid has the contribution as $n_r(i, j) = l - k + 1$, and the contributions from the whole image(all grids) are $N_r = \sum n_r(i, j)$. Finally, the fractal dimension D can be calculated as:

$$D = \frac{log(N_r)}{log(\frac{1}{r})}. \tag{2.9}$$

Usually, the finer textures are, the bigger D is. However, it is not robust against scale and different textures can have the same FD. In [24], six features are proposed to discriminate FD of original, high and low grey value, and horizontal and vertical smoothed images.

2.2.4 Spectral Textures Features

Spectral texture features are actually extracted in frequency domain, which is different from other approaches. After the transformation, the textures are interpreted in a different way in frequency domain which is easier for analysis.

Fourier Transform Based Texture Features

One of the common spectral approach is Fourier transform based [18, 25]. In Fourier analysis, a complicated periodic function is able to be presented as the sum of a few weighted sinusoids and cosinuoids of different frequency. When it comes to 2D situation, waves with different frequencies in different directions are able to generate various patterns of texture. By investigating Fourier spectra, different textures have their own characteristics. Textures with regularity, their power spectra are concentrated. Those with low regularity have scattered spectra. The spectra of directional textures are of directionality, and those with fine textures have more in high frequency. The advantage of Fourier transform based approach is that it is easy and quick to calculate [4, 18]. However, this feature has no spatial information and is not robust against rotation and scale.

Wavelet Transform Based Texture Features

The wavelet transform is one of the most popular time-frequency methods, and it has been used in textures analysis more and more nowadays. Instead of sinusoids and cosinuoids functions, wavelet transform represents function with wavelet. Wavelet transform generates four sub-bands. The length and width of each sub-band is the half of original image size.

Different sub-band has different information. LL sub-band contains low frequency information, whereas HL, LH, and HH sub-bands hold high frequency information. All these information are useful for texture analysis. The HL sub-band has the directionality of horizontal, and LH is in vertical direction. The directionality of HH sub-band is diagonal. Figure 2.4 is a example of 2-level wavelet transform.

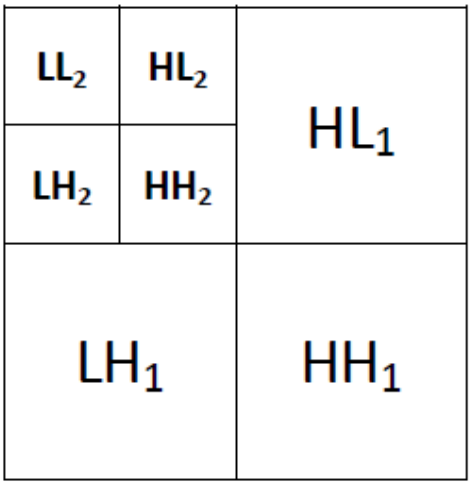

Figure 2.4: 2-level Wavelet transform

In [20], wavelet transform is used for texture extraction. The author prefers one-level wavelet transform and HH1 domain are chosen for analysis instead of other sub-bands as it is thought to have more information. Various texture features are used acquired and a sliding window are employed to extract features. Those features are contrast, diagonal moment, energy, entropy, homogeneity, second diagonal moment, and uniformity. In [26], an image is divided into 4×4 blocks and features are extracted from each block. Six features are employed. Three of them are the average colour components, and the other three are the energy of high frequency sub-bands. The square root of the second order moments of coefficients are computed in HH, HL, and LH sub-bands, as moments of wavelet coefficients can effectively represent characteristics of textures [27]. Multiple resolution and sub-bands with different

frequency information of wavelet transform texture method are advantages for texture analysis. However, its disadvantages of not enough orientation information and rotation sensitive are raised in [4].

2.2.5 MPEG-7 Texture Features

MPEG-7 standard provides three texture features, and they are Texture Browsing Descriptor, Homogeneous Texture Descriptor (HTD) and Local Edge Histogram Descriptor. The first two features are introduced in the following paragraphs.

Texture Browsing Descriptor

Texture browsing descriptor describes textures by three characteristics, regularity, directionality, and coarseness [5]. The regularity are ranged from level 0 to 3, which takes 2 bits of storage. If the regularity value is 3, it means textures have strong periodic pattern; whereas 0 indicates the textures are irregular. The directionality ranges from 0° to 150°, and is divided into 6 value with 30° each. It can represent at most 2 directions, and each direction takes 3 bits. The value of 0 means the textures move no obvious directionality: while high value indicates dominant directionality. The coarseness is ranged from 0 to 3. The bigger value is the coarser textures are. The image is filtered by scale and orientation related filters so that descriptors are generated [28]. Texture browsing descriptors are useful for database browsing, and its perceptual advantages is able to contribute of candidates selection in image retrieval. This descriptor is compact as it only takes at most 12 bits for storage.

Homogeneous Texture Descriptors (HTD)

Homogeneous texture descriptor is generated by series of filters and is in the frequency domain [5]. The frequency domain is divided in both angular and radial directions. In angular direction, it is equally divided in steps of 30°. While in radial direction, it is partitioned into five octaves. As a result, the frequency are divided into 30 channels, and 2D Gabor functions are then applied to individual feature channels. The energy e_i and energy d_i deviation are computed in each of the i^th channel. A feature vector of HTD is then obtained from mean

intensity, standard deviation and energy [5]. The similarity between query image and images in the database are matched by measuring the distance between TD of query images the all the other TDs of images in the database. Scale and rotation invariant is also able to be achieved with these vectors.

Edge Histogram Descriptor(EHD)

Edge histogram descriptor is a straightforward texture descriptor that is useful for image to image matching[5]. There are five types of edges: horizontal, vertical, 45° diagonal, 135° diagonal and non-directional. An image is divided into $4 \times 4(16)$ sub-images, and a 5-bin histogram of edge distribution is generated for each sub-image which corresponds to 5 different edge types. To generate the edge histogram, each sub-images are further partitioned into blocks. The number of blocks within each sub-image is constant, and the size of block needs to be a power of 2. Thus, the block size is scalable, and it varies according to the size of image. Each block is further divided into 2×2 blocks for a third time, and each of them is treated as a single pixel, whose intensity is the average of all pixel values within current 2×2 block. The edge detector operators are applied on these special 2×2 block "pixel ", and those ones whose edge strengths are greater than a threshold are edge blocks and contribute to the histogram. The whole process is illustrated in Figure 2.6. The edge detector operators are corresponding to five edge types and are presented in Figure 2.5. There are 80 bins in total, and the edge histogram bin values are normalized and non-linear quantized to achieve 3 bits/bin. Similarity is measured by calculating L1 norm distance between two edge histograms.

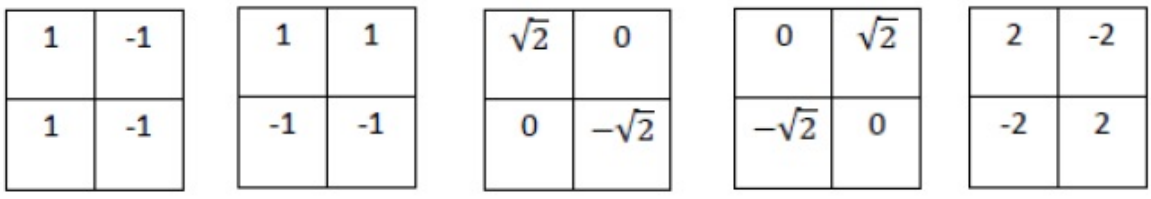

Figure 2.5: Edge histogram descriptor computation.

A variation of EHD is to extend the histogram by global and semi-global histograms. The global histogram is a global edge distribution by combining all the 16 sub-images. The semi-global histograms are vertical and horizontal edge distribution. Thus, there are 5 bins for global histogram, 65 bins for semi-global histograms, and 80 basic bins, so the total number of bins is 150. The similarity are measured by a weighted L1 norm distance. Further detail can be found in [5].

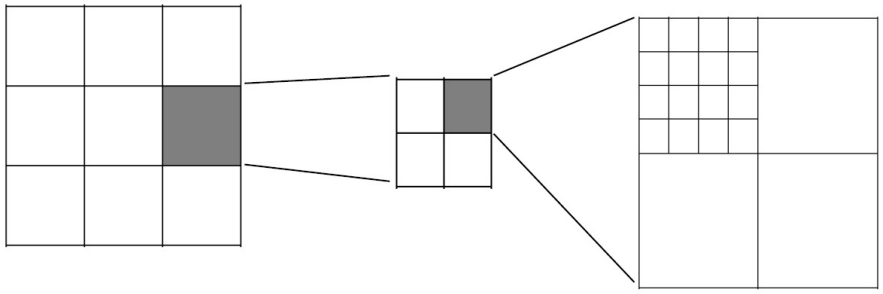

Figure 2.6: An example of chain codes.

EHD is scale invariant, but not rotation invariant. It is effective for natural images with non-uniform edge distribution and those are similar in semantic meaning. EHD can be combined with other descriptors like colour features, and the performance can be improved.

2.2.6 SIFT

Scale-invariant feature transform was proposed by David Lowe [6, 29]. This feature comes with many advantages and are very popular as it is able to find objects with a near real-time performance. A brief implementation and review of this method is presented in this subsection.

The first step of implementation is to construct scale space, which is the base of finding features that are invariant across various scales. The construct scale space is acquired by a convolution operation of Gaussian function and input image. For each octave, therefore, the original image is convolved with Gaussian functions and then a set of scale space images

are generated. The second step is to obtain the difference of Gaussian (DoG), as the scale-space extrema in DoG is able to detect stable keypoint locations. The DoG images are then obtained by subtracting adjacent scale space images. After current octave, the Gaussian image is down-sampled by factor of 2 repeatedly.

The next step is find local DoG extrema. Each pixel is compared with its eight direct neighbours and 3×3 neighbours in two adjacent scales so as to find out the maxima and minima. A few checks need to be done to remove points with low contrast points with edge response. The following step is assigning keypoints orientations. For each point,$L(x, y)$, the orientation $\theta(x, y)$ and $m(x, y)$ magnitude are calculated.

The gradient orientation and magnitude of every pixel in a sample region around keypoint are obtained to accumulate into orientation histograms to create a keypoint descriptor. The coordinates are rotated according to the keypoint orientation in order to achieve orientation invariant, and a Gaussian weighting function window is employed for weighting. The value of each bin is corresponding to the sum of gradient magnitudes in current direction. In [6], the size of sample region is 16×16 and it is divided into 4×4 subregions. A 8 bins orientation histogram is constructed for each subregion. Thus, a 128 element feature vector is generated for each keypoint. Finally, the feature vector is modified to achieve illumination invariant.

SIFT is invariant to scale, rotation, illumination change and viewpoint change. It is also not sensitive to noise and distinctive that suits retrieval mission in a massive database. However, it is hard to extract keypoints for object with smooth boundary, and sometime SIFT is not able to provide enough keypoints.

The application of SIFT in content-based image retrieval can be seen in [30]. As different images have different number of keypoints, the author model every feature vector as a hyper point under unknown distribution, and the Earth Movers Distance is employed. The dissimilarity of two keypoints are measured by EMD, and more details can be found in [31]. In [30], the author concludes that SIFT is invariant to colour component (channel), and the description quality of keypoint and the size of feature vector are positive related, which means

the quality of keypoint drops when the size of feature vector decreases. Furthermore, the use of the colour information in local feature vector is recommended as it has better performance than the original SIFT method.

2.3 Shape Descriptor

Shape descriptors are another important kind of features as they have some connection to the human visual understanding of images. They are classified into two classes: contour-based and region based approaches [32]. Each class is further divided into structural methods and global methods. Some famous and popular methods are introduced in the following paragraphs. The MPEG-7 shape descriptors are introduced at the end of this section.

2.3.1 Contour-based Shape Features

Contour-based shape features concentrate on object boundary. According to [32], these types of features are further divided into two categories. The first one is structural methods. The typical methods are chain code and polygon. The other category is global methods like Fourier descriptors and Wavelet descriptors. Some of these methods are introduced in the following paragraphs.

Chain Codes

Chain codes describe the boundary of an object by a sequence of unit-size lines of specific directions [32]. The object boundary is superimposed with a grid, and the boundary vertices are generated by finding the nearest grid point. As long as the basic unit chain code, a 4-directional chain code or a 8-directional chain code is chosen, and one start point is decided, a chain code sequence is able to be generated. There are two problems of a chain code. The first one is that different start points can have different chain code sequences. To solve this problem, the chain code is considered a circular sequence. Find out the sequence resulting in the minimum magnitude and its start point is chosen. The other problem is rotation

invariant. A method called first difference of a chain code can solve this problem. In anti-clockwise direction, the next chain code subtracted from the previous one is the first difference. It is then considered a circular sequence so as to achieve rotation invariant. A simple example of eight-directional chain code is presented in Figure 2.7. However, chain code is sensitive to scale and noise.

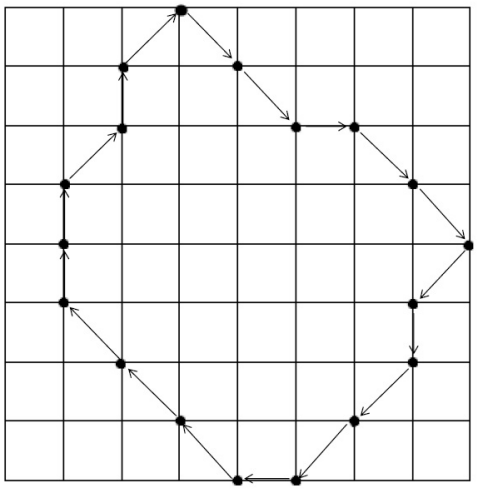

Figure 2.7: An example of chain codes.

Polygon Methods

The boundary of an object is able to be approximated by a polygon. A simple way [7] to achieve this goal is finding a straight line connecting two extreme points and then find the points with the farthest distance from this straight line. The polygon can be generated by connecting these points together. In [32], primitives, which is the boundary segments, are polygon vertices. A feature of four element string is employed and the editing distance is used to measure the similarity between two strings. A binary or m-nary tree are built up by organising features, which makes the feature matching easier. This feature, however, is sensitive to rotation and scale. A method of chain vectors is proposed in [33], where given N interest points are acquired from polygon approximation. A pair of basis vector are selected and normalized as a unit vector along x-axis. As long as this coordinate system is set up,

the rest of the vectors are transformed into this system. Any other pair of points can be chosen as basis vector to achieve transform invariant. Other information of the basis vector like length, location and angle is recorded as parameter as well. Euclidean distance are used to measure similarity. An example of polygon approximation is presented in Figure 2.8. The line that joins two extreme points(B, D) are discovered, the the farthest points(A, C) on both sides from this line are then found. A polygon are then constructed based on four points A, B, C, D.

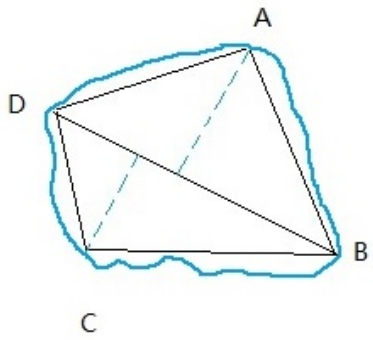

Figure 2.8: An example of polygon approximation.

The Fourier Descriptor

In [7], a method of Fourier descriptor is proposed. Let $(x_0, y_0), (x_1, y_1), (x_2, y_2), ..., (x_{K-1}, y_{K-1})$ denotes series of points on the boundary of a object in anti-clockwise direction. This two dimensional space is able to be converted into one dimensional space by treating the coordinate as a complex number as below:

$$s(k) = x(k) + jy(k), \tag{2.10}$$

where $k = 0, 1, 2, 3, ..., K-1$. According to [7], the reconstruction function of $s(k)$ can be acquired by Fourier descriptor. Based on Fourier descriptor's properties, it is not sensitive to rotation, scale, start point, translation and noise and is easy to normalize [32].

Statistical Moments

Different moments like mean or variance in statistical analysis can be used to describe and analysis shape features is introduced in [7]. A segment of region boundary can be converted into one dimensional curve segment. By connecting its two end points and rotating the line segment connecting two end points into x-axis, this boundary segment is able to be treated as a probability density function $p(x_i)$. x_i is discrete random variable, where $i = 1, 2, 3, ..., N$. The amplitude are considered histogram of $p(x_i)$. The r^{th} moment is given as

$$\mu_r = \sum_{i=1}^{N} (x_i - m)^r p(x_i) \tag{2.11}$$

where μ_2 is the variance and the mean of x is given as

$$m = \sum_{i=1}^{N} x_i p(x_i) \tag{2.12}$$

This methods is not sensitive to rotation and easy to obtain. But the physical meanings of high order moments are hard to describe.

2.3.2 Region-based Shape Features

Region-based shape features consider the whole region of object rather than just boundary. These features are also divided into global methods and structural methods [32]. Serious moments methods like geometric moments and Zernike moments are typical global methods. Structural methods are approaches like convex hull. Some of popular approaches are presented in this part.

Convex Hull In Euclidean space, a region A is convex if for any two points within this region, the straight line segments connecting these points is within the region. In Euclidean space, the convex hull of a region is the smallest convex region B that meets the condition $A \subset B$ [32], and the rest part is called convex deficiency. A few approaches of convex hull extraction

are proposed like boundary tracking way, and so on. The polygon approximation is first employed so as to smooth irregular boundary, which decreases computation time significantly. Both convex hull and convex deficiencies are then divided from the region, and again both convex hull and convex deficiencies are obtained from previous convex deficiencies. This process will be repeat again and again until all the derived convex deficiencies are already convex.

Geometric Moment Invariants

Moment features are another popular shape features based on statistical properties of image. A famous paper about moment invariants was published by Hu in 1962 [34]. In this paper, the $(p+q)^{th}$ order moments of an image is given. In physics, moment is used to describe the distribution of the density of mass. Similarly, the zeroth order moment represents the mass of an object and the first order moment are used to describe the centre of mass. For digital two dimensional image, integration is able to be interchanged with summation processes [34]. The normalized central moments are when acquired with the property of invariant. Thus, Hu proposed seven moments that are invariant to shift, rotation, and scale. In [32], the author points out that geometric moment invariants have advantage of simple shape description, but perform badly on shapes like arbitrarily distorted contour-based shapes or region-based shapes with something inside on scale, and so on.

Orthogonal Moments

There are more moments are proposed like orthogonal moments, for example, Legendre moments and Zernike moments are introduced in [32]. Zernike introduces a sequence of polynomials $V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho)exp(jm\theta)$, and they are orthogonal in $x^2 + y^2 \leq 1$. Zernike moment is invariant to rotation and easy to build higher order moment. It is considered preferable for shape description [32]. There are also other orthogonal moments like Legendre moments and pseudo-Zernike moments. Shape moments are robust and not difficult to calculate and match. However, the connection between their physical features and high order moments are still a problem.

2.3.3 MPEG-7 Shape Features

A few shape descriptors are introduced in MPEG-7 standard. Similarly, they are divided into contour-based and region-based categories as well. The other way of classification is based on 2-D or 3-D space [10].

One region-based method is angular radial transformation (ART) which is based on moments [35]. Similarly, it is defined on a unit disk in polar coordinates. The ART coefficients for similarity matching is computed from the image intensity and the ART basis function where both angular and radial directions are used. ART descriptors are not sensitive to segmentation noise and compact.

Curvature scale-space (CSS) is a contour based approach [10, 35]. A sequence of points are selected on the contour of object and x and y coordinates are obtained, then filters are applied. The coordinate values of prominent peaks are extracted. 3D shape descriptor is based on a shape spectrum [10]. 2D/3D shape descriptor is actually a presentation of 3D object by 2D snapshot from various angles [10]. Generally, MPEG-7 shape descriptors are extracted from image and can be used in retrieval separate from it.

2.4 Machine Learning

In [3], the author points out that a single similarity measure is not able to achieve robust retrieval results and the ranking is not perceptually meaningful as well. Thus, more things need to be done to improve retrieval results. According to [2], moreover, the gap between low level image query and high level queries needs to be reduced. There are many ways to achieve these objects and one important way is machine learning. Machine learning is divided into supervised learning and unsupervised learning. Both of them are introduced in this section.

2.4.1 Supervised Learning

Supervised learning is based on input measure, and it is a prediction of the value of outcome measure [2]. It improves retrieval speed and accuracy in massive database significantly, and it analyses low level descriptor and conclude high level concepts. Some of important supervised learning methods are briefly introduced in this subsection.

Support Vector Machine (SVM)

Support vector machine has been used in many areas like text classification and it is also a ideal method for image retrieval. It has rigorous theoretical foundations with accurate performance [2, 36]. SVM is a binary classification. Assume the set of training example be $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ where $\{x_1, x_2, ..., x_n\}$ is an input vector in space $X \subseteq R^d$ and y_i is its class label, and $y_i \in \{1, -1\}$. There are many hyperplanes that separate positive and negative training data. The aim is to find a optimal separating plane with maximal margin. Margin is the distance between the closest dad point of each class and hyperplane. When there are more than one concept for retrieval, a SVM is trained for each concept. In [37], for example, 23 concepts needs to be trained through SVM. More details about SVM are presented in [38].

Decision Tree

The decision tree is another widely used method for classification. A decision tree classifies the input data by a set of rulse, which are paths from the root to the leafs. A leaf is a class and each node in the tree is a test of a certain rule. Thus, it is crucial to construct a good decision tree. There are a few famous methods like ID3, C4.5 and CART [2, 39, 40].

Most decision tree algorithms are based on a core algorithm which employs a top-down greedy manner. ID3 is a typical example. It starts with all training examples are at the root and following processes are run repeatedly: The attribute that is able to partition training examples best is chosen as root node. A branch is created for each possible value of root node. The training examples are trained onto a proper branch. More detail about ID3 and other approaches can be found in [2]. [39] is an example that C4.5 is used to classify image

database. Another example of CART decision tree can be seen in [40].

Decision tree generates rules that easy to understand. The computation is not high and robust against noisy or incomplete features [2]. However, the classification of continuous attributes is still a problem. Large number of classes may increase errors faster.

Bayesian Classification

Another famous approach to achieve supervised learning is Bayesian classifier [41]. The use of Bayesian classifier in content based image retrieval traced back to [42]. When it comes to learn high level concepts out of low level features, Bayesian classifier is a good candidate. It is based on Bayes' rule that posterior is proportional to prior times likelihood: $P(A \mid B) \propto P(A)P(B \mid A)$. Therefore, an example falls into a specific class when this class has the most probability. An example of semi-Naive Bayesian method can be seen in [41]. [43] is another example of binary Bayesian classifier that extracts high level concepts from low level content based image descriptors. This moethod can be implemented with high efficiency. However, due to the assumption that the classes are conditional independent, the accuracy decreases when the examples are connected.

2.4.2 Unsupervised Learning

Clustering is the typical unsupervised learning technique, and is often considered synonymous with the key words unsupervised learning [2]. Clustering is a method that data are divided into different groups (clusters), and examples in a certain group are similar to each other while examples in different clusters are quite different. Unlike supervised learning, unsupervised learning has no target attribute with the data, which means there is no priori data example given for learning. There are a large number of clustering algorithms and a few of them are presented here. More clustering methods like Normalized cut or CLUE are introduced in [2].

K-means clustering

K-means algorithm is a very famous and traditional method and it belongs to partitional clustering methods. This method and its variations are widely used for image clustering.

K-means algorithm divides data into k clusters, where k is given by user. The algorithm first randomly chooses k data points to be the initial cluster centres, which is called centroids. All the data points are assigned to their closest centroids. Within each new cluster, new centroid is computed using current membership and previous centroid is discarded. This process is repeated again and again until convergence criterion is met. In [44], k-means clustering is employed map from low level features to textual keywords. Some variations of k-means clustering and its implementation are seen in [41, 45]. K-means clustering is a easy method with high efficiency. However, it is sensitive to initial centroids and outliers. Another problem is that user needs to specify k manually.

Hierarchical Clustering

The other type of clustering is hierarchical approaches. Hierarchical clustering generates a tree of cluster, which is also called Dendrogram [46]. There are two different ways of clustering. One is bottom up way, and the other one is top down. In the bottom up way, a pair of cluster with most similarity or closest distance are merged. The process continues until there is only one cluster left. The bottom up method is, whereas, the other way round. The root cluster is split into child clusters and the child clusters are split in the same way until every cluster has only one single point.

Another problem is the measurement of distance of two clusters, and there are many. Single link method defines the distance between two clusters is the shortest distance between all instances. This method is good at finding clusters with arbitrary shape, while it may affected by noisy points. Complete link method is a different way of measurement, where the distance between two clusters are the distance of two furthest data points in the two clusters. This approach is more practical and compact, but it is sensitive to outliers. Other methods like average link or centroid link are variations. An example of image retrieval with hierarchical clustering based on colour content is presented in [46]. Another example of hierarchical clustering with multiple colour features is in [47].

2.5 Image compression

As the development of technology, more and more multimedia data with higher quality and resolution are generated every day. As a result, the demand of larger storage space and transmission bandwidth keep rising. A good way to solve this problem is compression, which is able to reduce the size of multimedia data significantly and keep the contain at the same time to fit the limitation of storage space and bandwidth. The development of compression has a long history. A large number of methods have been invented to satisfy various needs. In this thesis, only a few famous methods in image compression domain are introduced. More details can be found in [7].

The basic idea of image compression are redundancy reduction and irrelevancy reduction [48]. The purpose of redundancy reduction is to reduce duplication from images. These redundancies can be either spatial redundancy or spectral redundancy. Irrelevancy reduction aims at removing image contents that is not sensitive to human eyes. Finally, image compression focuses on decreasing the number of bits presenting an image. A typical image compression model is seen in Figure 2.9.

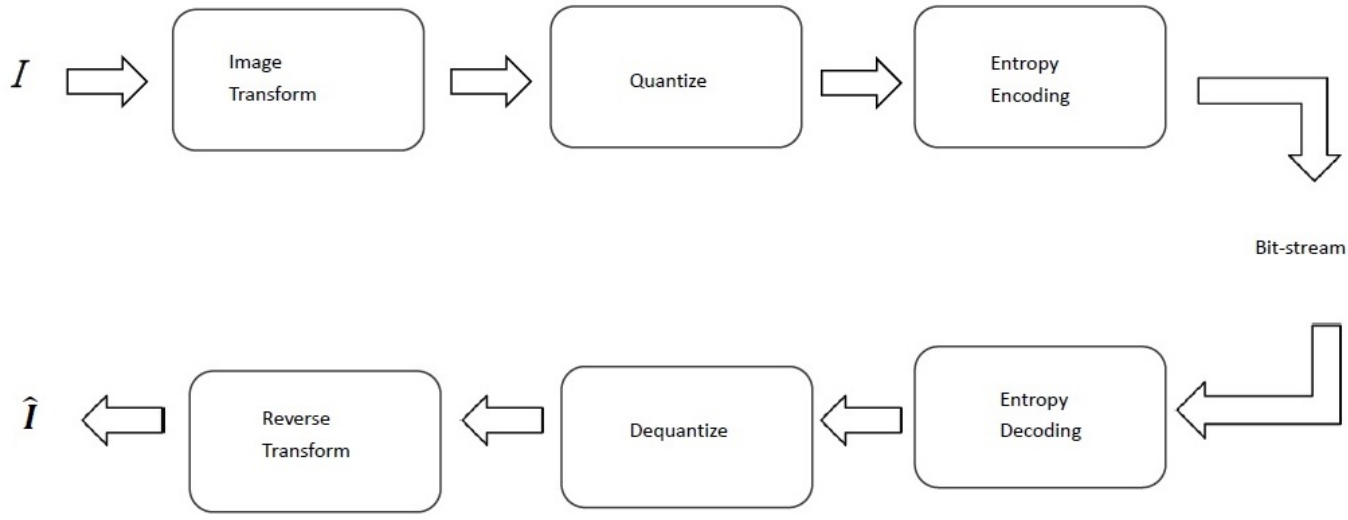

Figure 2.9: A typical compression model

Image transform is the first step and various approaches have been developed such as Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), and so on. Quantization is then applied to reduce the number of bits of transform coefficients. Finally, quantized values are coded by Entropy Encoder. The probabilities of quantized coefficients are determined, and based on that different codes are generated. As a result, the output code stream is smaller than input stream. A few commonly used still image compression standards are introduced in this section.

2.5.1 JPEG

Joint Photographic Experts Group (JPEG) is a commonly used image compression standard [7]. JPEG is based on DCT which can be considered discrete Fourier-Cosine series. The encoding of JPEG standard starts with colour space transformation. The input image colour space are converted into YCbCr colour space, and then Cb and Cr channels is able to be downsampled according to the sensitiveness of human visual system. Each channel is then divided into 8×8 blocks, where all the processing steps are implemented. The DCT is first applied and compression is achieved mainly by concentrating most of signal into lower spatial frequencies. The DCT coefficients are quantized uniformly, and are then ordered in a zig-zag scan pattern. This zig-zag sequence is important for entropy coding as low frequency non-zero coefficients are scanned before those with high frequency. A further compression is achieved by entropy coding as it utilize the statistical properties of coefficients. Generally, JPEG is a great compression algorithm. However, "blocking artifacts" are generated in compressed image since the input image needs to be blocked, and these artifacts become more obvious when the compression rate rises.

2.5.2 Wavelet Based Compression

The wavelet transform is used in image compression, which is similar to Fourier transform, but better in many ways. The idea of wavelet transform is to use a series of basis functions to

represent any arbitrary function. These basis functions are generated by scaling and shifting mother wavelet. More details about wavelet transform can be seen in [7]. Wavelet based compression methods are more robust against errors. A higher compression rate is achieved by wavelet transform, and there is no blocked artifacts like JPEG. Many wavelet based coding algorithms haven be proposed and some of them are introduced here.

Embedded Zerotree Wavelet (EZW)

Embedded Zerotree Wavelet (EZW) [49] compression method is based on octave-band decomposition Figure 2.10.

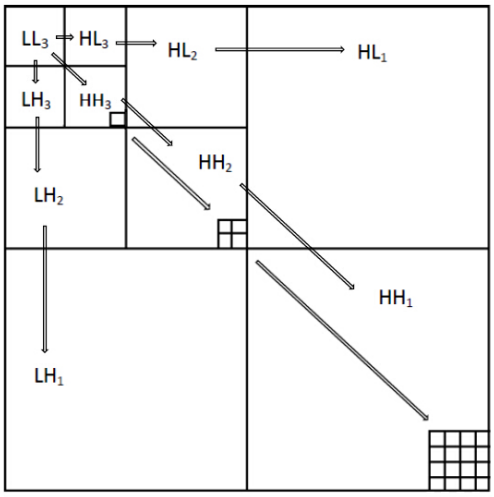

Figure 2.10: Octave-band decomposition

There is a parent-child relationship. Each coefficient in the lower frequency subband has four children in the corresponding location in higher frequency subband. The highest frequency subband is at the the bottom right, while the lowest frequency one is located at top left. Based on this, an assumption has been made that if a wavelet coefficient is insignificant concerning threshold T, then all of its children has a high possibility to be insignificant with respect to T as well. This coefficient is called zerotree root, and this zerotree root and all of its children form a zerotree. A specific symbol is assigned to zerotree root which represents all its children is able to be predicted as insignificant. Thus, there is no need to encode

its children and these coefficients in the higher frequency subbands can be discarded, which improve the compression efficiency significantly. In EZW encoding, the threshold decreases after each scanning pass until it is smaller than the smallest coefficients. Different symbols are assigned and the coefficients are arranged in a order of importance, which is from the lower frequency subbands to higher frequency subbands. In all, the encoding are implemented in a progressive way. The bits added to the bit stream increase the quality of reconstructed image, while both encoder and decoder are able to stop at any point to meet a certain bit rate.

Set Partitioning in Hierarchical Tree (SPIHT)

Various enhancements are proposed based on EZW, and Set Partitioning in Hierarchical Tree [50] is an good improvement. The zerotree structure in EZW is a efficient way to represent insignificant data. However, there is another tree whose root is significant, whereas the remaining nodes are insignificant, which cannot be represented efficiently with zerotree structure. Thus, SPIHT is proposed with spatial orientation tree and concepts about set of offspring of nodes and set of descendants of nodes. The SPIHT generates a embedded bit stream which makes re-construct possible when the bit stream is truncated at any point. The progressive transmission of the coefficient values are employed that coefficients are ordered by magnitude and then the most significant bits are transmitted first. As a result, the performance of SPIHT is better than EZW in most cases.

Scalable Image Compression with Embedded Block Coding with Optimized Truncation (EBCOT)

This method is the fundamental of JPEG 2000 standards. It is based on independent Embedded Block Coding with Optimized Truncation of the embedded bit-streams [9, 51]. This is also a wavelet based algorithm with both spatial and resolution scalability. The target bit rate at the time of compression is not necessary. Unlike JPEG, it does not need to be compressed multiple times so as to achieve a target bit rate either. After wavelet transform, each wavelet subband is divided into small block units called code-blocks. For each code-block, a separate embedded bit stream is generated. Similarly, the coefficients are

quantized and then analysed and coded bit-plane by bit-plane with coding strategy related to significance. A new concept called quality layer is brought in that each code block stream is divided into quality layers. As a result. Thus, the EBCOT algorithm is resolution, spatial and distortion scalability. The separate code blocks scheme achieves error resilience as well as implementation efficiency.

2.6 Image Retrieval in Compressed Domain

Nowadays images are almost stored and transmitted in compressed form [8]. As most CBIR techniques introduced in previous few subsections operate in the pixel domain, images needed to be decompressed into the pixel domain before feature extraction which leads to a computational overhead and low efficiency. An alternative is to extract features directly in compressed domain.

Many compressed domain descriptors appeared based on different compression approaches. These methods do not need to decode the compressed data. In contrast, they are able to extract features straight from the compressed domain and finish retrieval without decompression. Compressed domain retrieval approaches can be spatial based like vector quantization and fractals; while they can also be transform domain based, such as methods using cosine transform, Fourier transform or Wavelet transform [52].

2.6.1 Vector Quantization and Fractal Based Retrieval Techniques

Vector quantization can be considered an indexing technique [52], where a Euclidean space is mapped into a finite set Y. The finite set Y is called codebook with a N vectors. The input vector is mapped onto one of a set of the codebook using a nearest neighbour rule, and the input image is presented by codebook labels. Two different techniques are introduced in [52]. In [53],the author introduces a method start with one codebook entry and the cluster with the largest variance new are added as new entries. A LBG algorithm is employed to

optimise the codebook. The author thinks the image retrieval based on VQ data provides both colour content information and spatial information, which because the image is divided into blocks that coded as a whole. A Modified Hausdorff distance is recommended to compare VQ codebooks as it is more robust.

The concept of fractal has been introduced in 2.2.3. When it comes to image compression, it is the opposite of fractal image generation. Fractal image compression looks for sets of fractals in an image that represent and describe the whole image, rather than generating an image from a given formula. Once the proper sets of fractals are decided, they are reduced to compact fractal transform codes. A few different fractal codes based techniques are presented in 2.2.3, and one of them makes a comparison between fractals and wavelet, from which it is concluded that fractals outperforms for various type of images while wavelet are more effective for images with strong texture. However, the author stresses the conclusion is valid under given framework.

2.6.2 JPEG Compressed Domain Retrieval Techniques

It is mentioned in [8] that JPEG is the dominant image format on the Internet. Many researches of CBIR have been done in JPEG compressed domain. In [54], the author points out that working in compressed domain is faster than in pixel domain, as the latter usually needs approximate 15% of the processing time. A comprehensive overview of the most popular techniques of CBIR in JPEG compressed domain is given in [54]. All the algorithms operate on DCT data, which avoid the inverse DCT that is the most expensive operation when decoding JPEG images.

An early approach was proposed by in Shneier and Abdel-Mottaleb [55]. An image is divided into windows. The average of each block across the window is a key for this window. Windows of the same image are paired and the difference are compared with a threshold. Hamming distance are used to measure similarity between two images.

Another approach was developed by using low energy JPEG coefficients to increase effi-

ciency [56]. Top four coefficients are recommend according to many experiments. The sum is calculated and a histogram is generated. The similarity is measured by calculating L1 norm distance between histograms.

In [57], Schaefer proposed an approach using both texture and colour descriptors. These descriptors are only extracted from the DC terms. LBP operator [54] is applied on DC data of Y component(luminance), and a LBP histogram is generated. Chromaticity DC data are used to generate a colour histogram. L1 norm distance are employed when comparing both LBP and colour histograms.

It has been found in [58] that average colour of a block of size 8×8 can be acquired from DC component directly. Similarly, the average colour of a 4×4 block can be obtained from F_{11}, F_{10} and F_{01} AC components. It is proved, however, just employing DC component is more effective than both DC and AC components. By analysing the DCT blocks statistically, another method of describing an image is developed in [59]. The distribution of moments are analysed so as to obtain an estimation of texture information. The variance and mean of each block is obtained from all AC components and DC component respectively, and a histogram of them are built.

An edge histogram detector is developed in [60]. It is actually similar to a MPEG-7 texture descriptor called nonhomogeneous texture descriptor [5] whose introduction can be seen in 2.2.5. The variance of AC coefficients determine whether an image block contains an edge. The edges within blocks are considered horizontal, vertical, 45° or 135° according to the difference between F_{10} and F_{01} AC coefficients. The input image is divided into 4×4 blocks, and each block has a histogram. A combination of local, semi-global and global edge histograms is used for similarity measurement.

Another method with multiple descriptors are presented in [61]. Colour, texture and edge information are collected in this research. A colour histogram is constructed from F_{11}, F_{10} and F_{01} AC coefficients of 4×4 blocks. There are 12 key descriptors describing the texture which can be divided into two types. An energy histogram is built up by using different bands

of energy coefficients for each block, which comprises the first type texture features. The edge information within each block is extracted to generate the second type texture features. Different descriptors are normalised before similarity measurement.

Experiments and comparisons have been made under certain circumstance in [54]. All JPEG based methods introduced above are generally comparable in terms of speed. The techniques used in [58] and [60] are similar to pixel domain algorithms, and therefore their retrieval performance are quite close to that of the pixel domain algorithms. The method in [58] is proved the best one as it is both the fastest one and outperforms other methods in terms of retrieval. As only DC components are used, it takes less time than other methods. The second, third and fourth best approaches are [56], [61] and [57] respectively. It seems that features based on AC coefficients have a relatively poor performance in this comparison.

It is proved in [54] that a dramatic reduction of computational complexity is achieved by retrieving images straight in compressed domain instead of pixel domain while the retrieval performance keeps comparable.

2.6.3 Wavelet Based Retrieval Techniques

In [62], a texture analysis and classification method based on tree-structured wavelet transform is proposed. The author points out that signals like quasi-periodic signals, whose dominant frequency channels are located in the region of middle frequency, are quite different, and significant information of a texture often appears in channels of middle frequency. As a result, the traditional pyramid type wavelet transform whose partitions are conducted in low frequency subbands may not be suitable. The tree-structured wavelet transform is a variation of wavelet transform. During the wavelet transform, the energy of each decomposed image are calculated. If the energy of a subimage is too small than the average, which means it has few information, the decomposition of this subimage is stopped. In contrast, subimage with large energy is decomposed further, and feature vectors are extracted from the energy of these subbands. The indexing is conducted by matching feature vectors of images.

A method using moments and wavelet is proposed in [63]. It is based on the assumption that with limited camera operation, pdf of wavelet bands of similar images are similar as well. Different images can be distinguished by the amount information from vertical, horizontal and diagonal directions at different scales. In order to reduce the complexity of straight comparing histograms of subbands, distribution parameters are chosen for comparison. The generalized Gaussian density function is employed to model histogram, and the standard deviation of the coefficients σ and the shape parameter γ are used to describe the histogram. Therefore, the similarity between images is obtained by calculating the difference between parameters.

Another approach is proposed in [64] that the retrieval can be done by directly comparing wavelet coefficients. After wavelet decomposition, image is rescaled to size of 128×128. Features are extracted from each image for retrieval. They are average colour and the sign and indices of m largest magnitude wavelet coefficients. It is reported that this method has a good performance. However, it is not rotation and translation invariant as the index is connected to location of coefficients. A similar technique is introduced in [65], images are also rescaled into the size of 128×128 after wavelet decomposition. To do image matching, based on the variance of low pass subband, one fifth of the images are first retrieved. The difference of coefficients are used to choose a fewer number of images. Finally, image retrieval is implemented based on the different of coefficients of low pass subband and the other vertical, horizontal and diagonal subbands as well. This methods is better than Jacobs', but still not able to solve the situation of rotation and translation.

A joint wavelet based method called WaveGuide is presented in [66]. Images are compressed with wavelet transform and a set of combination of descriptors, including colour, texture and shape descriptors, are extracted in wavelet domain. Image is compressed with four level wavelet transform. The number of significant coefficients in each subband is collected as texture feature as it indicates the significance of a specific subband. Nonuniform colour histograms and colour moments are calculated for each of the colour component in the YCbCr colour space as colour features. The spatial moments are obtained as the shape

features. The contribution of each feature can be adjust by weighting function dependent on the actual characteristic of different images. This method has a comprehensive use of colour, texture and shape descriptors, and it reduce the computational complexity.

In [67], a new retrieval scheme based on JPEG 2000 is proposed. The author introduces a new concept of information tree and a tree-distance measurement. Each EBCOT sub-block corresponds to a node of the information tree, and each information tree corresponds to a bit-plane. Thus, the similarity measurement is done by comparing information tree of different images. By evaluating the entropy of different bit-palnes, mid-level bit-planes are chosen so that information tree carries maximum information. To implement image retrieval, the JPEG 2000 bit stream is partly decoded to obtain information about blocks, parameters and so on. The block-wise wavelet transform and quantization is then conducted on the original image according to these information so that the information trees are able to be set up. The similarity measurement is achieved by comparing the distance information trees.

Rather than extracting significant features, signatures are obtained from the distribution of wavelet transform in [68, 69]. For image with texture, the coefficients of each subband after wavelet transform are distributed with a generalized Gaussian law with parameters α (scale factor) and β (shape parameter). The texture of image is able to be described with estimated parameter $\hat{\alpha}$ and $\hat{\beta}$. The similarity measurement is finished by calculating the distance. The distance is achieved by a weighted sum of the divergences, where the divergence of the distribution law of wavelet coefficients of every subband. When certain subbands are more relevant, their weights can be increased to make more contribution. The weights can be learnt automatically during classification. The author has another image retrieval scheme in [70] where histograms are employed to build signatures. If N is the number of decomposition levels, the signature consists of $3N + 1$ histograms. The largest and smallest value are calculated for every N, and each subband has a 32 bins histogram. After normalization, the distance are calculated with a weighted function.

A scalable coding based feature extraction method is proposed by Charith in [71, 72].

Low level visual descriptors are mapped into resolution-quality spaces so that they are robust against content adaptation like the change of resolution or quality. The proposed descriptors are based on Embedded Zerotree Wavelet (EZW) coding algorithm which has been introduced in 2.5.2. The wavelet coefficients at a specific bi-plane can be classified into six classes according to information provided by significance switching mask [72] during coding. A sliding window is employed to scan from the most significant bit-plane to the least one, and a significance histogram is generated by counting if a specific symbol appears in current window. A hierarchical feature descriptor is constructed in this way, and it is mapped straight to bit-planes and subbands of the wavelet coefficients. When making comparison, only related bit-planes and subbands are considered according to different content adaptation mode. The proposed descriptor is compared with MPEG-7 descriptors, and outperforms latter such as colour, texture and edge. Furthermore, it has a robust retrieval performance against content adaptation compared to MPEG-7 descriptors.

2.7 Summary

In this chapter, the state-of-art visual descriptors, content based image retrieval (CBIR) related techniques are discussed. The visual descriptors are presented in an order of colour class, texture class and shape class. The state-of-art methods like MPEG-7 and SIFT are debated. Image retrieval related machine learning techniques are also briefly introduced. The image compression and CBIR in compressed domain are investigated which includes wavelet based, JPEG based and other compressed domain image retrieval methods.

In conventional CBIR, low-level visual features are usually extracted from the original version of visual content, i.e., the highest resolution and quality. However, there are more than one version of the same content generated in practical usage in order to satisfy the variations in bandwidths, storages and display device resolutions. A concept of content adaptation resilient descriptors has appeared in compressed domain CBIR, and a literature has been made into an EZW based descriptor which maps into scalable structure and bit streams are

extracted during scalable coding so that they are robust against the content adaptation. There is substantial potential to improve the performance and robustness of adaptation resilient descriptors in compressed domain, and within the scope of this thesis, advanced methods can be proposed. In the following part of this research, new descriptors and retrieval system with adaptation resilience are developed in a new JPEG 2000 compression standard, and features and advantages of JPEG 2000 algorithm are incorporated.

Chapter 3

Proposed Low Level Descriptors Based on JPEG 2000

In this chapter, new low level descriptors are developed based on state-of-art JPEG 2000 standard and an image retrieval system based on proposed descriptors are designed. The JPEG 2000 compression standard is introduced first and a detailed explanation is given on JPEG 2000 block coding procedure. Two new descriptors, data states and contexts, based on JPEG 2000 coding algorithm are then proposed and developed. The extraction of features and other details are presented.

3.1 JPEG 2000 Compression Standard

As the development of image compression, more and more methods have been proposed, such as DCT, DFT, and DWT based techniques. Some of the approaches focus on different image transformation, while others concentrates on different coding schemes and so on. The invention of JPEG was proposed date back more than a decade ago, and this standard is still very popular nowadays. However, JPEG is not able to solve many advanced requirements nowadays.

One of the major deficiencies of JPEG standard is a lack of scalable coding. With the help

of scalable coding, the visual contents only need to be encoded once for the highest resolution-quality version and then decoded in many ways to cater various needs. If the image, in other words, needs to be adapted to multiple versions, JPEG standard has to decode and encode for each of them separately, which is inefficient. As for image retrieval, scalable feature vectors can be realized by scalable coding techniques, and an illustration is given in Figure 3.1. When the descriptor and feature extraction scheme are based on scalable coding techniques, the feature vector is adaptable. Whatever version the query image is, image feature vectors in the database are able to discard irrelevant part to match the one of query image. Unfortunately, normal feature vectors are invariable. When the query image resolution-quality version is not the same as the ones in the database, new feature vectors must be extracted from all images in order to match the one of query image.

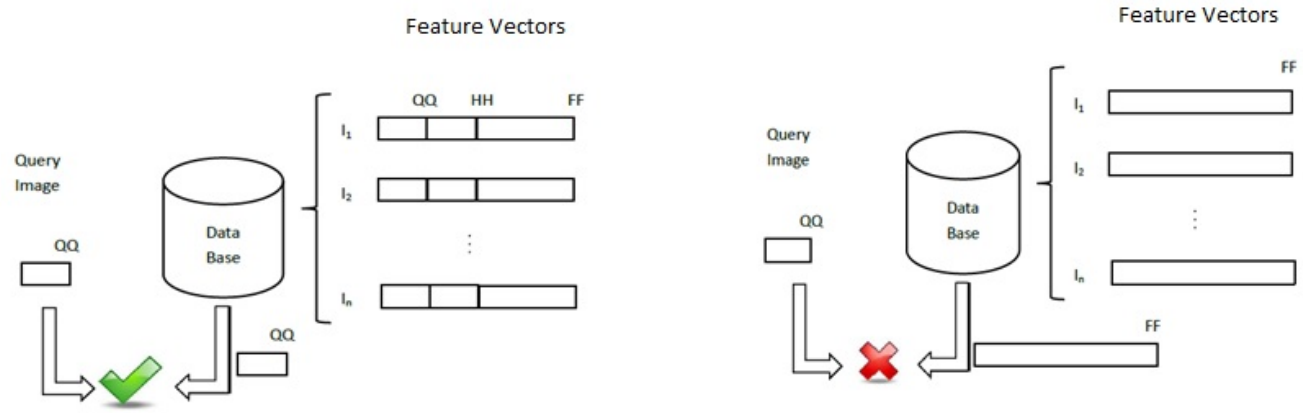

Figure 3.1: A comparison: Scalable feature vectors(left) and Normal feature vectors(right).

As the size and quality of images are very important for digital imagery nowadays, the JPEG 2000 standard is proposed to cater multiple needs like efficiency and scalability. It has to face various applications like mobile devices, printing, E-libraries, Internet, digital photography. Each of them brings new requirements to fulfil. As a result, a few key features of JPEG 2000 standard are:

At low bit rate situation, a better performance is proved by JPEG 2000 standard compared

with other standards. Both lossy and lossless compression can be achieved in progressive coding. The image reconstruction can be done with the increase of the spatial resolution or pixel accuracy by progressive transmission. An open architecture is provided so that the system can be optimized according to different needs. Some parts of image, namely Region-of-interest, can be defined manually to ensure a lower distortion and higher quality in transmission. The design of code stream makes it robust to bit errors. More features are explained in [9, 73].

3.1.1 JPEG 2000 Working Mechanism

The image compression is realized by compression engine, which is consisted of encoder and decoder. At the encoder of JPEG 2000 compression engine, the input image is first wavelet transformed, and then the wavelet coefficients are quantized and entropy coded so as to form the bit stream. This bit stream is the compressed data which can be either transmitted or stored. The decoder does the same thing, but the reverse direction. The bit stream is first entropy decoded and dequantized. A inverse wavelet transform is then applied, and then the image is reconstructed.

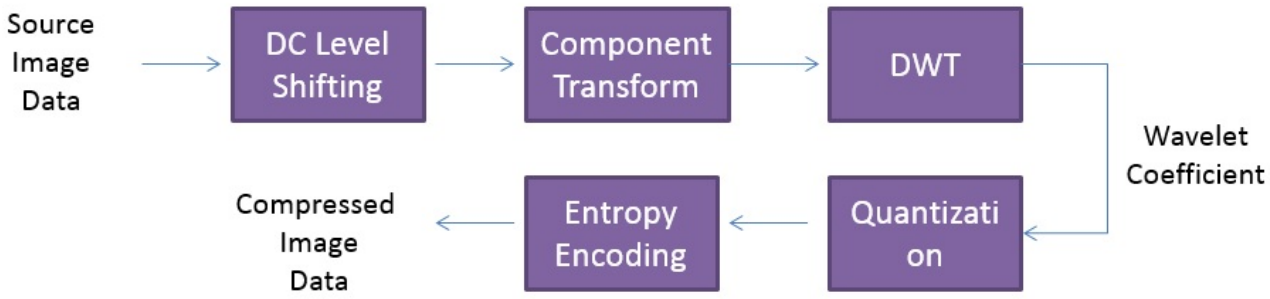

Figure 3.2: JPEG 2000 encoder

Figure 3.2 illustrates different procedures of encoder. The input image is first decomposed into components, for example, an image with YCbCr colour space has three colour channels,

and each channel is a component. Each component is decomposed separately.

Each component is divided into rectangular nonoverlapping blocks called tiles. Tiles are the basic units with the same dimensions, and all the operations such as the wavelet transform, quantization and coding are implemented on the tiles. The propose of tiling is to decrease the requirement of memory. Furthermore, it makes reconstructing parts of the image possible rather than decoding the whole image. If samples of the components are unsigned, DC level shifting is applied on samples of the tile by subtracting 2^{p-1} (p is the precision of the component). The next step is component transformation. It can be either irreversible component transformation which is for lossy coding or reversible component transformation for lossy or lossless coding. More detail about component transformation can be found in [9, 74].

The tiles are then decomposed into multiple levels for analysis by wavelet transform. Subbands compose these decomposition levels. The coefficients of different subbands have different spatial frequency information in vertical, horizontal or diagonal directions. When the samples are one dimensional, the forward discrete wavelet transform decomposes them into high pass and low pass subbands. Low pass subbands are a down-sampled, low resolution version of original samples; while high pass subbands are a down-sampled remaining version that contains details. There are reversible and irreversible ways to implement DWT. The default reversible way is Le Gall 5-tap/3-tap filter, and the default irreversible way is Daubechies 9-tap/7-tap filter. In JPEG 2000 standard, the reversible compression employs 5-3 integer wavelet transform and a reversible component transform, by which the original image is able to be recovered from the compressed data stream if no compressed data is discarded [75]. In irreversible compression, a 9-7 floating point wavelet transform and an irreversible component transform are used, but both of them have round-off errors which lead to original image is not able to be recovered from the data stream even when no data is discarded. [75] has a further study into the difference between reversible and irreversible compression. Two filtering modes can be chosen, i.e., convolution based and lifting based, and further details can be found in [73].

After the wavelet transform, the wavelet coefficients are quantized. The coefficients are divided by the quantization step-size Δ_b. Different subbands b can have different quantization step-sizes, but each subband has only one step-size. All the quantized coefficients must have signs. In most cases, quantization is a lossy operation, but the quantization step-size is set to be one when reversible compression.

An arithmetic coding system is employed to conduct entropy coding. This system is context dependant, and it compresses binary symbols in respect to an adaptive probability model. JPEG 2000 standards adopts MQ coder. There are 19 coding contexts which will be explained in the following sections. In order to reduce the cost of coded segments and make fast probability adaptation possible, for any given type of bits, the number of contexts is less than 10. A lazy coding mode is introduced to decrease the number of symbols. More details can bee found in [1]. ROI is able to be chosen during the coding process, and markers are added so as to achieve error resilience. As the code stream is generated, the header of the stream provides information on the original image and coding details so that decoder is able to construct the image. The process of how JPEG 2000 encoder works have been presented so far, the way decoder works is similar but in an opposite direction.

3.1.2 Spatial Partition and Block Coding

Each subband is partitioned into nonoverlapping rectangular blocks after quantization. Spatially consistent rectangles from three different subbands, i.e., HL, LH and HH band, at the each resolution level comprise a precinct. Again, each precinct is further partitioned into more nonoverlapping rectangular blocks, named code blocks. The size of code blocks is 64×64 or smaller. It is the basic input unit of entropy encoder.

The definitions a few terms will be used later are introduced here first. In JPEG 2000, the significance of location $\mathbf{j}$ at the p^{th} bit-plane , and the of this location is given as

$$\sigma^{(p)}[\mathbf{j}] = \begin{cases} 1, & \text{if } v^{(p)}[\mathbf{j}] > 0, \\ 0, & \text{if } v^{(p)}[\mathbf{j}] = 0, \end{cases} \tag{3.1}$$

where $v^{p}[\mathbf{j}]$ is the value of location $\mathbf{j}$ at the p^{th} bit-plane. When $\sigma^{(p)}[\mathbf{j}] = 1$, the current location is significant, otherwise it is insignificant. The context of a specific sample refers to its eight direct neighbours, i.e., two vertical, two horizontal and another four diagonal neighbours. When all of the eight neighbours are insignificant, this sample has zero context; otherwise it has nonzero context.

The system visits code blocks in raster order and then codes them bit-plane by bit-plane. The coding starts with the most significant bit-plane with nonzero element, one bit-plane each time, to the least significant bit-plane. The independent embedded block coding means each code block is coded independently. Each bit-plane of each code block is scanned. Four continuous bits in a column comprise a stripe. Scanning starts from the top left stripe of the code block, then moves to the second stripe in the same row. After all the stripes in a same row are scanned, the scanning jumps to the second row of stripes. Scanning continues in the same way until the whole code block has been scanned. During the scanning, each bit of coefficients is coded in only one of three passes. They are the significance propagation pass, the magnitude refinement pass and the cleanup pass, which are known in short as pass0, pass1 and pass2, respectively.

In significance propagation pass, a bit is coded if its location is insignificant, but at least one of its eight direct neighbours are significant. In other words, this pass codes those coefficient bits that are not significant and have a nonzero context. All other coefficient bits are skipped. In magnitude refinement pass, a bit is coded if it became significant in a previous bit-plane. In other words, this pass codes coefficient bits that have already become significant apart from those just became significant in the preceding significance propagation pass. In cleanup pass, all bits that has not been coded in previous two passes are coded. That is to say that this pass codes coefficients which are not significant and have a zero context. Contextual information

about the bit plane samples are collected within each coding pass by arithmetic coder. Bit stream is generated according to its internal state and these information. This part will be explained in detail in the next subsection.

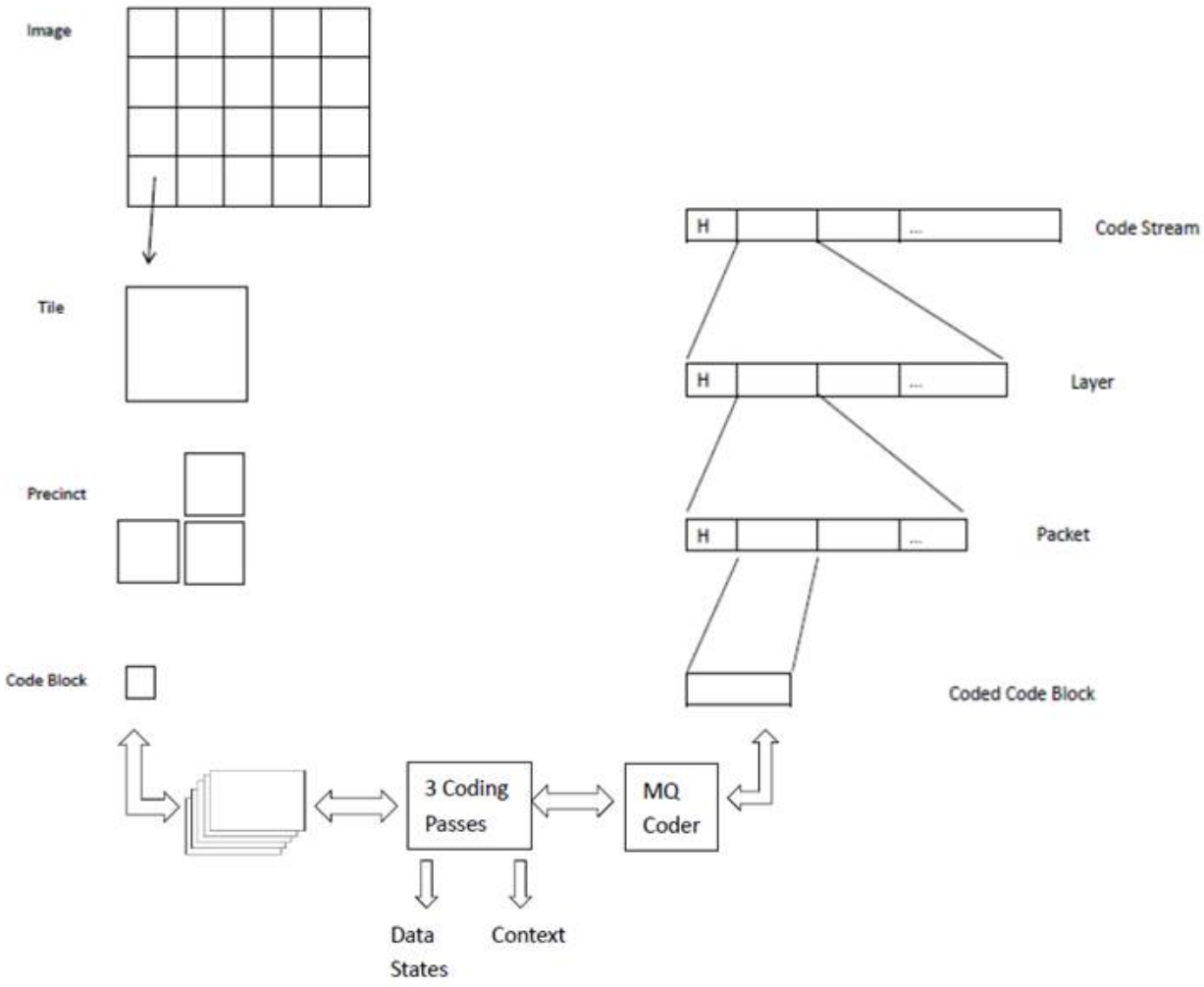

Figure 3.3: Correspondence between the spatial and the bit stream representations.

A different bit stream is generated for every code block separately.Bit streams of different code blocks do not share information. Truncation points are allocated to each code block by rate distortion optimization. The distortions and lengths are calculated and stored in the bit stream as well during the encoding process. Bit streams of code blocks in a precinct

build up the body of packet. The layer then consists of packets from each precinct of each resolution level. The final code stream consists of a sequence of layers. In other words, a layer is a quality increment for the full resolution image, and each layer increases the image quality successively. A packet is similarly a quality increment for a resolution level at a certain location. Components are coded separately and code stream are interleaved on a layer basis. The correspondence between the spatial and the bit stream representations is presented in Figure 3.3. Both encoder and decoder follow this code stream structure and the separate coding strategy that makes quality progression, resolution progression, component progression and spatial progression possible [9]. One more operation passes over all code blocks when the whole image has been compressed. The purpose of this operation is to achieve a specific target bitrate and distortion. It decides how the bit stream of each code block need to be truncated.

3.2 The JPEG 2000 Block Coding

In this subsection, the bit-plane coding is explained in detail, and MQ coder is briefly introduced. The bit-plane coding procedure turns a sign-magnitude version of coefficients into an embedded representation for each code block. The MQ coder has the ability of turning binary symbols to compressed bit streams. Within each code block, each coefficient bit of each bit plane is coded in only one of three passes. They are significance propagation pass, magnitude refinement pass and cleanup pass. The context κ is determined during the pass coding within a 3×3 neighbourhood, and is depended on the sign information and the state of its eight direct neighbours and itself. Within each pass coding, a symbol x and context κ pair is generated as the input of MQ coder. The MQ coder generates an embedded bit stream. The introduction starts with the MQ coder, and then a deep study of three passes are presented.

3.2.1 The MQ Coder

The MQ coder is an adaptive binary arithmetic coder which generates binary outcomes based on probability estimates. The probability estimates are adaptive models which are able to evolve in different contexts [1].

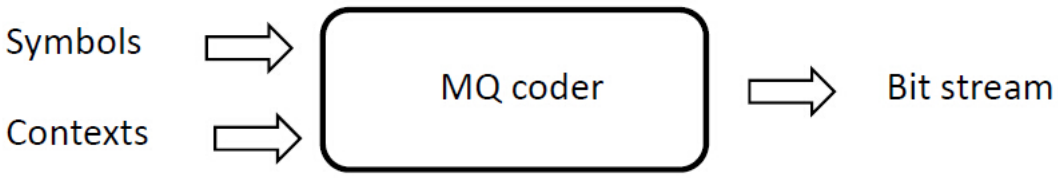

Figure 3.4: MQ coder

The MQ coder are illustrated in Figure 3.4. Generally speaking, it encodes a sequence of input symbols $x_n \in \{0, 1\}$ and related context labels k_n, and generates a single compressed codeword [76]. According to the possibility of the occurrence, these symbols are divided into two categories: less probable symbols (LPS) and more probable symbols (MPS). A probability model can be represented as an interval. It is divided into two sub-intervals, each of which represents the probability of each symbol. When a LPS or MPS occurs, the sub-interval corresponding to this symbol turns into a new interval. New intervals are repeatedly and recursively split in the same way until all symbols are received. Essentially, the division of intervals are dependent on contexts, and decided by the MQ coder that consists of probability mapper, internal state variables and other parts. A further study of MQ coder and how it works can be found in [1].

3.2.2 Significance Propagation Pass

Coding pass $P^{(p,0)}$ is the first coding pass. A bit belongs to this pass if its location is not significant, but has at least one significant neighborhood; that is at least one of its eight neighbors have to be significant [1]. Pass0 only includes coefficient has a nonzero context(at least one significant neighborhood) though it is insignificant currently, for it is most likely to become significant in current bit-plane. Other samples are skipped.The sign coding is called immediately if a bit becomes significant.

The implementation of pass0 is in [1], and *Encode-Sign Procedure* is a part of *Encoder-Pass0 Procedure*. The flow chart of these tow procedures are shown in Figure 3.5 and Figure 3.6.

The value of $v^p[\mathbf{j}]$ is the p^{th} bit of location $\mathbf{j}$, and its sign is $\chi[\mathbf{j}]$. $\pi[\mathbf{j}]$ indicates whether $\mathbf{j}$ has been coded in pass0. The most important element $\sigma[\mathbf{j}]$ is the significance of a bit. Its definition has been given in previous section. $\kappa^{sig}[\mathbf{j}]$ is the first type of contexts that ranges from context 0 to context 8. They are determined by location $\mathbf{j}$'s neighbours' significance, i.e., the value of $\kappa^h[\mathbf{j}]$, $\kappa^v[\mathbf{j}]$ and $\kappa^d[\mathbf{j}]$. $\kappa^h[\mathbf{j}]$ is dependent on its two direct horizontal neighbours, namely its value is the sum of $\sigma[j_1, j_2 - 1]$ and $\sigma[j_1, j_2 + 1]$. Similarly, $\kappa^v[\mathbf{j}]$ and $\kappa^d[\mathbf{j}]$ are the sum of σ of their vertical and diagonal neighbours respectively. The other factor affects the value of $\kappa^{sig}[\mathbf{j}]$ is to which subband the current code block belongs to. If code block belongs to LH band, which is vertical high pas of coefficients, horizontal significant neighbours are the most important indicators of the significance current location, as significant samples have the most possibility to occur from feature in horizontal direction. The vertical neighbours have the second priority as the indicators of significance. If the code block belongs to HL band, the contribution of vertical and horizontal neighbours are reverse. For code blocks that is in HH band, it is diagonal neighbours contribute the most.

$\kappa^{sign}[\mathbf{j}]$ is another type of contexts that ranges from context 10 to context 14. They are related to a combination of its neighbours' χ and σ. According to [1], χ^{flip} is the sign-flipping factor which depends on $\chi^{-h}[\mathbf{j}]$ and $\chi^{-v}[\mathbf{j}]$ as

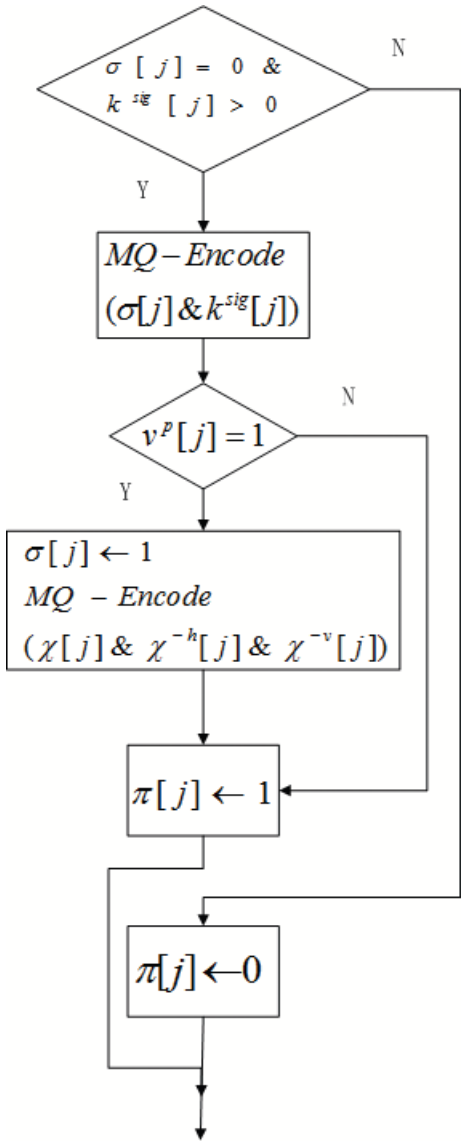

Figure 3.5: Flow chart of significance propagation pass

$$\chi^h[\mathbf{j}] = \chi[j_1, j_2 - 1]\sigma[j_1, j_2 - 1] + \chi[j_1, j_2 + 1]\sigma[j_1, j_2 + 1], \tag{3.2}$$

$$\chi^v[\mathbf{j}] = \chi[j_1 - 1, j_2]\sigma[j_1 - 1, j_2] + \chi[j_1 + 1, j_2]\sigma[j_1 + 1, j_2], \tag{3.3}$$

Distribution of $\chi[\mathbf{j}]$ is identical to the distribution of $-\chi[j]$, given a neighbourhood where the signs of all neighbours are flipped, so the definition is given as

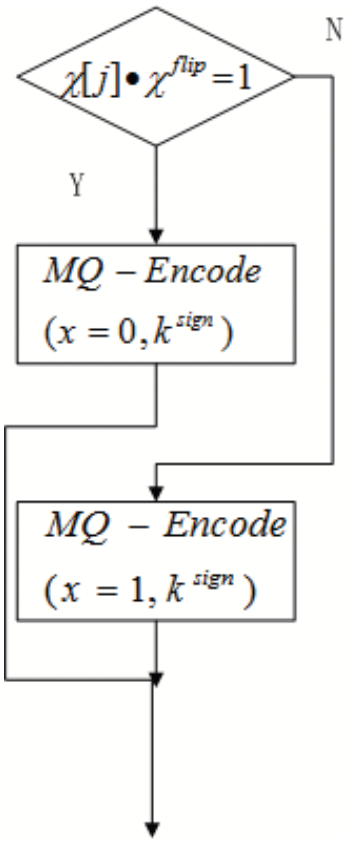

Figure 3.6: Flow chart of sign coding

$$\chi^{-h}[j] = sign\left(\chi^{h}[j]\right) min\left\{1, \left|\chi^{h}[j]\right|\right\}, \tag{3.4}$$

$$\chi^{-v}[j] = sign\left(\chi^{v}[j]\right) min\left\{1, \left|\chi^{v}[j]\right|\right\}, \tag{3.5}$$

According to Table 3.1, x takes the value 0 if $\chi[j] \cdot \chi^{flip} = 1$ and 1 if $\chi[j] \cdot \chi^{flip} = -1$.

There is a run mode of significance coding, which is introduced to reduce the compression complexity. Technically, the run mode belongs to significance coding. As the implementation of run mode is in Cleanup pass in [1] , it will be introduced later in pass2.

In summary, the symbol and context pairs that input into MQ coder are $x = v^{p}[\mathbf{j}]$ and $\kappa = \kappa^{sig}[\mathbf{j}]$ respectively in significance coding. In sign coding, the input symbol and context pair are $x = 0/1$ and $\kappa = \kappa^{sign}[\mathbf{j}]$. Other internal state symbols like $\sigma[\mathbf{j}]$ serve in pass0 and

Table 3.1: Assignment of flipping factor for sign coding. [1]

$\chi^{-h}[j]$	$\chi^{-v}[j]$	χ^{flip}
1	1	1
1	0	1
1	-1	1
0	1	1
0	0	1
0	-1	-1
-1	1	-1
-1	0	-1
-1	-1	-1

play important roles in the logic judgements in the procedure. Internal state symbol $\sigma[\mathbf{j}]$ decides if a symbol and context pair should be outputted to the MQ coder and the further sign coding should start, while its own value will be changed once the condition is satisfied. The value of $\pi[\mathbf{j}]$ is also decided in this pass which prepares for the next pass. When an insignificant coefficient starts to have one or more significant neighbours, it is very likely to become significant as well. Thus, the task of significance propagation pass is to handle these coefficients with potential.

3.2.3 Magnitude Refinement Pass

The second pass, $P^{(p,1)}$, encode bits which has became significant in a previous bit-plane, i.e., bit which is higher than p, apart from those have just been coded in pass0($P^{(p,0)}$). Figure 3.7 illustrates how the magnitude refinement coding works.

where $\overleftarrow{\sigma}[\mathbf{j}]$ denotes the value of the significance state variable, $\sigma[\mathbf{j}]$, "delayed" by a bit-plane. When the sample first becomes significant,it remains zero until the first magnitude

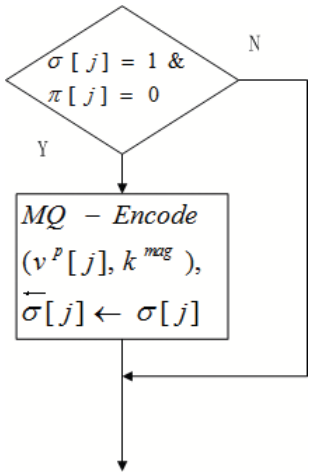

Figure 3.7: Flow chart of magnitude refinement pass

refinement bit has been coded [1]. Thus, $\overleftarrow{\sigma}[\mathbf{j}]$ provides a crude indicator of the coarser index magnitude, $v^{p+1}[\mathbf{j}]$, when coding $v^{p}[\mathbf{j}]$.

Contexts κ^{mag}, i.e., context 15, context 16 and context 17, are determined by a combination of $\kappa^{sig}[\mathbf{j}]$ and $\overleftarrow{\sigma}[\mathbf{j}]$. In other words, the significance of its eight neighbours, and whether $v^{p+1}[\mathbf{j}] \geq 2$ [1] are considered.

In summary, the symbol and context pair that inputs into MQ coder is $x = v^{p}[\mathbf{j}]$ and $\kappa = \kappa^{mag}[\mathbf{j}]$ respectively in magnitude refinement coding. Other internal state symbols work for this coding pass and contribute to the logic judgement. Internal state symbols $\sigma[\mathbf{j}]$ and $\pi[\mathbf{j}]$ decide if pass1 should be activated, while internal state symbol $\overleftarrow{\sigma}[\mathbf{j}]$ works with context $\kappa^{sig}[\mathbf{j}]$ deciding the context κ^{mag}. The value of internal state symbol $\overleftarrow{\sigma}[\mathbf{j}]$ is given after MQ coding. A set of significant coefficients are generated after pass0, and their following bit will be coded in pass1.

3.2.4 Cleanup Pass

Clean-up pass, $P^{(p,2)}$, is the final pass. Bits that are not coded in pass0 and pass1 are coded in this pass. In other words, these locations are insignificant. This pass consists of two different coding modes, namely run mode and normal mode [77]. The former is similar to pass0, while

the latter improves in compression performance.

According to experience [1], most samples of a code block are insignificant in majority of bit-planes, it is necessary to employ a run mode to code multiple insignificant samples with a single binary symbol. Run mode are activated when four consecutive locations in the column are insignificant and each of them has only insignificant neighbourhoods. A few symbols are used in run mode and a brief introduction is given. r is a flag symbol, which is used to identify if run mode has occurred, and if so then how many bits do not need to code significance [1].

Two contexts κ^{run} and κ^{uni}, context 9 and context 18 respectively, appear in run mode. The appearance of κ^{run} means the run mode is activated. According to Figure 3.8, κ^{uni} only appears when the run interruption happens [1], which depends on the value of samples in the stripe.

Otherwise, normal mode is activated to code bits that have not be coded in pass0 and pass1. The coding procedures of normal mode is similar to significance propagation coding with same type of contexts, i.e., $\kappa^{sig}[\mathbf{j}]$. Figure 3.8 illustrates the connection of two modes. The sign coding is also invoked after the significance coding with contexts $\kappa^{sign}[\mathbf{j}]$.

In summary, the symbol and context pairs that inputs into MQ coder are more complicated in this pass. When the run mode is activated , the pairs is $x = 0$ and κ^{run} if run is not interrupted; the pairs are $x = 1$ and κ^{run}, $x = \lfloor \frac{r}{2} \rfloor$ and κ^{uni} and $x = r \bmod 2$ and κ^{uni} if run is interrupted. When the run mode does not occur, the input pairs of MQ coder are the same as pass0. Other internal state symbols work for the coding procedure and ensure it is able to run properly. Date state r is an important internal state throughout pass2 and it indicates whether the run interruption occurs and the number of samples to be MQ encoded. Other internal state symbols are the same as pass0. As each coefficient bit is coded only in one of three passes, those coefficients have not been coded in pass0 and pass1 are coded here.

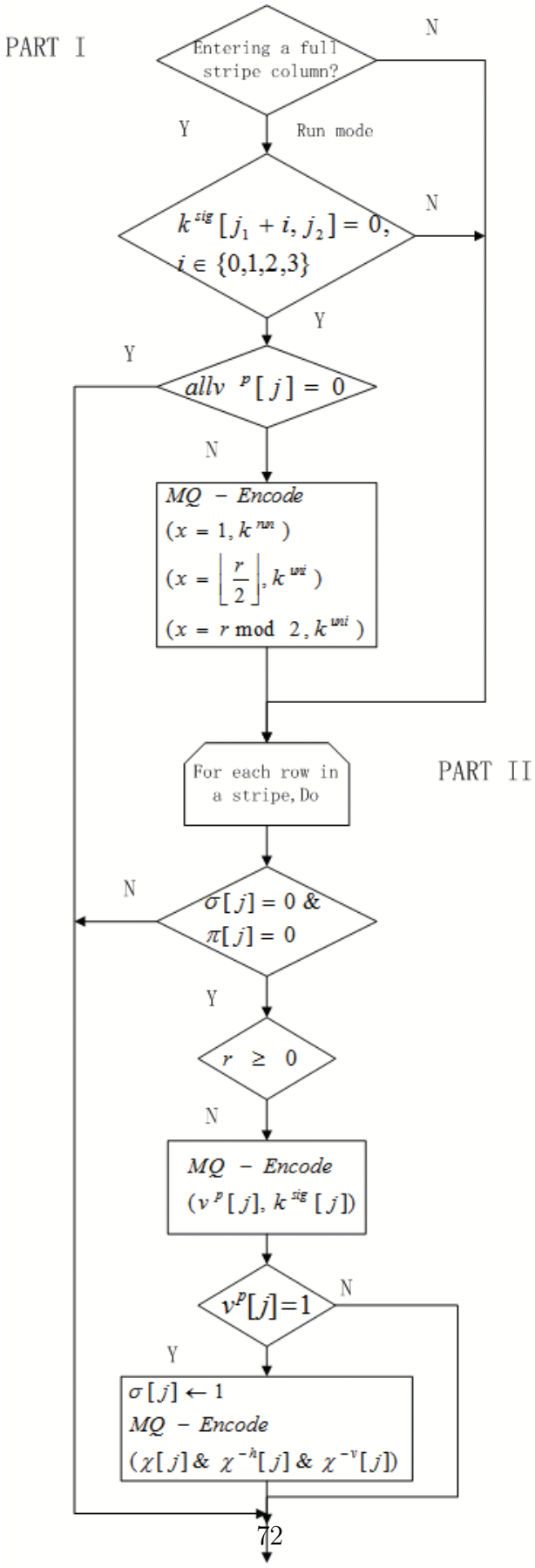

Figure 3.8: Flow chart of cleanup pass

3.3 The Design of The New Low Level Descriptor Scheme

In arithmetic coding system, many symbols are introduced during block coding. They are generally classified into two categories. One category is the 18 contexts that appear in three coding passes. The other category is other functional symbols employed in three passes. For example, symbol $\sigma[j]$ is a binary significance state; $\pi[\mathbf{j}]$ indicates whether $\mathbf{j}$ has been coded in pass0; The value of $v^p[\mathbf{j}]$ is the p^{th} bit of location $\mathbf{j}$. Inspired by this, more information can be extracted by combining multiple symbols or contexts together to serve the analysis, but not every symbol meet the requirement as some symbols have few practical meaning. Furthermore, by developing these combinations, new descriptors are generated for image retrieval and analysis. Those new descriptors derived from symbol category is called state, and descriptors developed from context category is named context. They are introduced separately.

3.3.1 Data States

Symbols appear in each of three coding passes are carefully selected for the analysis, but some symbols are skipped during selection as they has no practical meaning. The selected symbols and their combinations in each pass must cover all the possible situations that happens during coding. The construction of new data state descriptors are explained in order of coding passes.

Significance propagation pass

As the sign coding in Figure 3.6 is actually part of significance propagation coding in Figure 3.5, their symbols are considered together. The significance data state $\sigma[j]$ in pass0 and the sign symbol x in sign coding are selected. The relevant flow chart is in Figure 3.9, where the relationship of each symbols and the possible combinations are presented clearly.

When a location is coded in pass0, whether $v^p[\mathbf{j}] = 1$, and if so the sign that is coded in sign coding generates a combination of $\sigma[j]$ and x. A new sate is assigned to each leaf node and consequently Table 3.2 is acquired, where 3 new data states are created. The combinations of $\sigma[j]$ and x are presented by data state 2 and data state 3, which means the negative significance and positive significance respectively.

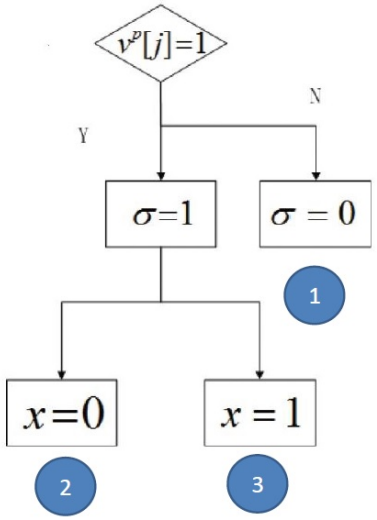

Figure 3.9: Relationship flow chart in Pass0

Table 3.2: data states of Pass0(Significance Propagation)

Data state No.	Symbols combination
1	$\sigma = 0$
2	$\sigma = 1$,$x = 0$
3	$\sigma = 1$, $x = 1$

Magnitude refinement pass

According to pass1 in Figure 3.7, two symbols are chosen. $v^p[\mathbf{j}]$ is the value of current bit-plane p, and $\overleftarrow{\sigma}[\mathbf{j}]$ is the significance data state of higher bit-plane. Their relationship tree is constructed in Figure 3.10 based on Figure 3.7 and [1]. One thing needs to be notice is the $\overleftarrow{\sigma}[\mathbf{j}]$ before MQ encoding is selected for analysis rather than the one after encoding.

Each leaf node is a new data state and then Table 3.3 is acquired. The value of $\overleftarrow{\sigma}[\mathbf{j}]$ before MQ encoding is considered, and so does the value of $v^p[\mathbf{j}]$. The combinations of $v^p[\mathbf{j}]$ and $\overleftarrow{\sigma}[\mathbf{j}]$ creates 4 new data states.

Cleanup pass

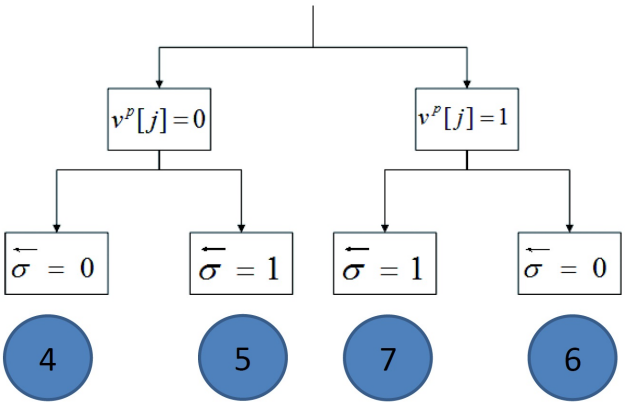

Figure 3.10: Relationship flow chart in Pass1

Table 3.3: Data states of Pass1(Magnitude Refinement)

Data state No.	Symbols combination
4	$v^p[j] = 0$, $\overleftarrow{\sigma}[j] = 0$
5	$v^p[j] = 0$, $\overleftarrow{\sigma}[j] = 1$
6	$v^p[j] = 1$, $\overleftarrow{\sigma}[j] = 0$
7	$v^p[j] = 1$, $\overleftarrow{\sigma}[j] = 1$

The analysis of pass2 is complicated, because there are two distinct modes, normal mode and run mode. The normal mode is similar to pass0, but the run mode is different as it is designed for the situation that four consecutive samples in the column are insignificant and each of them has only insignificant neighbourhoods. In Figure 3.8, the encoding procedure is divided into two branches. One branch is PART I corresponding to run mode, and the other one is PART II corresponding to normal mode. If the condition of run mode branch is met, encoder enters the run mode. When the encoding in run mode has finished, the encoder enters normal mode. If the encoder does not enter the run mode, then it moves on to normal mode automatically.

A few symbols that appear in Figure 3.8 are selected in this pass, and the relationship

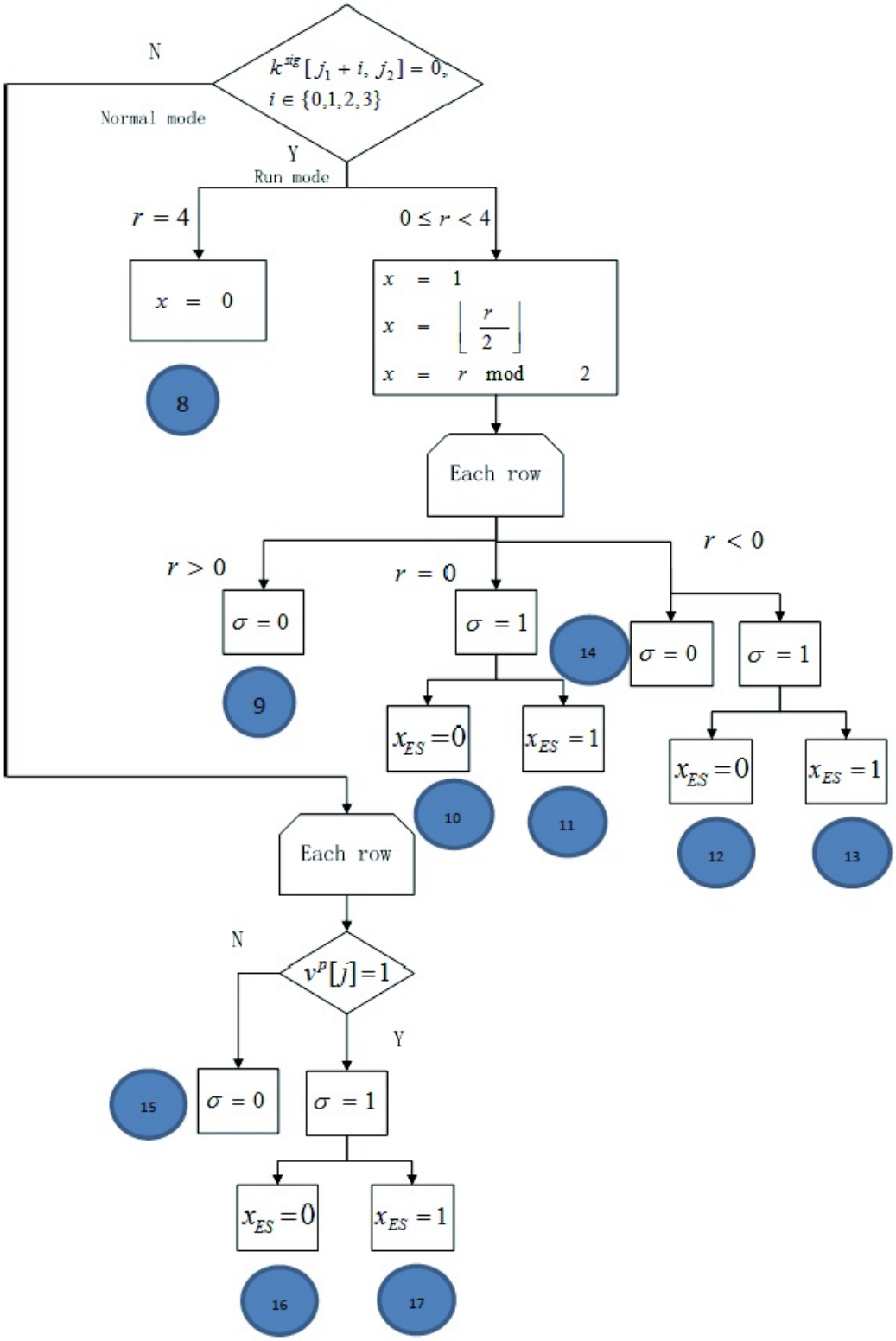

Figure 3.11: Relationship flow chart in Pass2

Table 3.4: Data states of Pass2(Cleanup)

Data state No.	Symbols combination
8	$x = 0$
9	(when $r > 0$) $x = 1, \lfloor \frac{r}{2} \rfloor, r \mod 2, \sigma[j] = 0$
10	(when $r = 0$) $x = 1, \lfloor \frac{r}{2} \rfloor, r \mod 2, \sigma[j] = 1, x_{ES} = 0$
11	(when $r = 0$) $x = 1, \lfloor \frac{r}{2} \rfloor, r \mod 2, \sigma[j] = 1, x_{ES} = 1$
12	(when $r < 0$) $x = 1, \lfloor \frac{r}{2} \rfloor, r \mod 2, \sigma[j] = 1, x_{ES} = 0$
13	(when $r < 0$) $x = 1, \lfloor \frac{r}{2} \rfloor, r \mod 2, \sigma[j] = 1, x_{ES} = 1$
14	(when $r < 0$) $x = 1, \lfloor \frac{r}{2} \rfloor, r \mod 2, \sigma[j] = 0$
15	$\sigma[j] = 0$
16	$\sigma[j] = 1, x_{ES} = 0$
17	$\sigma[j] = 1, x_{ES} = 1$

tree of symbols are illustrated in Figure 3.11. Symbol x is selected, but it is different from the x in pass0. The symbol x appears in run mode, and its value varies based on whether run interruption happens. The sign symbol in sign coding is written as x_{ES} to distinguish. σ is the significance data state. r is a temporal variable that ensures the running of coding procedure, but it plays a important role in cleanup coding. In run mode, its value depends on $v^p[j_1 + i, j_2]$, where $i \in \{0, 1, 2, 3\}$ [1]. The value of r leads to different branches, which can be seen in Figure 3.11. In different branches, the contexts are different, and the combination of symbols may vary. When $0 \leq r < 4$, the run interruption occurs and the value of x varies. The possible significance differs, and consequently the possible sign are different as well. If encoder does not enter run mode, then it enters normal mode directly, where the selection of symbols are the same as pass0. A new data state is assigned to each leaf node in Figure 3.11, and a list of data states is acquired in Table 3.4.

Strategy for Merging Data states

17 new data states has been created from three coding passes. These data states are unique, which means there is only one data state matches a given location. Each data state is a particular combination of symbols and internal state symbols appear in coding passes, and they have practical meanings which have been explained in previous section. However, the meanings of the data states are individual and lack of generalization. The data states need to be re-organized for more practical purpose and achieve more specific aim. The Inspiration comes from [49, 71, 72, 78], where EZW encoder considers sign of significance coefficients when encoding significance map. The sates are re-grouped and classified by the significance of coefficients and their signs.

Each bit are classified into Already Significant, Not Yet Significant or Currently Becoming Significant in terms of significance. With respect to sign of significance coefficients, the former two significance data states can be further classified based on whether their signs are positive or negative. This is illustrated in Figure 3.12, which is the idea guides how to regroup and merge data states.

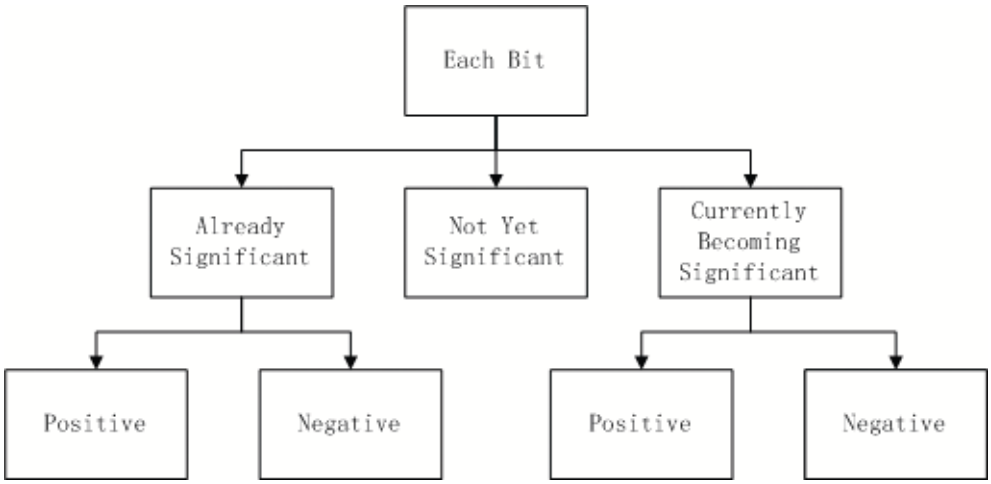

Figure 3.12: Merging data states based on the significance and sign

According to this guidance, data states created from three coding passes are further classified in Table 3.5. data states created in magnitude refinement pass are classified as already significant class, as this pass only codes samples that has already become significant in a

Table 3.5: Merged data state descriptors

	Data states	Merged data states
Already Significant	4,5,6,7	if positive, ⓔ if negative, ⓕ
Not yet Significant	1,15,8,9,14	ⓖ
Currently Becoming Significant	positive 2,16 10,12 negative 3,17 11,13	 ⓐ ⓑ ⓒ ⓓ

previous bit-plane. Data states in this class are further divided based on whether they are positive or negative, and they are Already Significant Positive(ASP) and Already Significant Negative(ASN). When a sample is not significant, then there is no need to consider its sign. These samples, therefore, are classified as Not Yet Significant(NYS). The class not yet significant chooses data states from significance propagation pass and cleanup pass. As for the class currently becoming significant, it mainly concerns data states generated from significance coding, which happens in both pass0 and pass2. In pass0, data states that are significant can be further classified based on whether they are positive or negative. They are Currently Becoming Significant Positive(CBSP) and Currently Becoming Significant Negative(CBSN). It is same when it comes to those data states generated in normal mode in pass2, but those data states that are obtained in run mode should not be confused with former ones. In order to distinguish, they are defined as Currently Becoming Significant Positive R(CBSPR) and Currently Becoming Significant Negative R(CBSNR).

After merging, previous 17 data states are re-organized into 7 data states. They actually belongs to three different classes, i.e., already significant, not yet significant and currently

becoming significant. Within each class, they are further divided into positive or negative. So the 7 data states are Not Yet Significant(NYS), Already Significant Positive(ASP), Already Significant Negative(ASN), Currently Becoming Significant Positive(CBSP), Currently Becoming Significant Negative(CBSN), Currently Becoming Significant Positive R(CBSPR), and Currently Becoming Significant Negative R(CBSNR). As these 7 data states have practical meanings, they form the proposed new low level descriptors for image retrieval and analysis.

3.3.2 Context

Similar to data state descriptors, context descriptors are also achieved by merging contexts appear in three passes. However, the aim of context descriptors is different from data state ones, the latter mainly focuses on individual location, while contexts concern a group of samples, i.e., individual location and its neighbourhood. Actually, the contexts in MQ coder are designed carefully and have abundant practical meanings. Thus, the purpose of creating context descriptors is to exploit the existing 19 contexts in three passes for better utilization.

Significance propagation pass

There are two types of contexts introduced in significance propagation pass, and they are $\kappa^{sig}[\mathbf{j}]$ and $\kappa^{sign}[\mathbf{j}]$. The former one ranges from context 0 to context 8, while the latter one ranges from context 10 to context 14. $\kappa^{sig}[\mathbf{j}]$ represents the situation of significances of its eight direct neighbours, and it is related to which subband it is at as well. $\kappa^{sign}[\mathbf{j}]$ depends on a combination of its neighbours' significances and signs at the same time. As the data state descriptors have already concentrated on significance and sign, so there is no need to repeat the same work. In addition, $\kappa^{sign}[\mathbf{j}]$ contains a mixed information of significance and sign together, which has no contribution for image retrieval according to empirical observation. Therefore, $\kappa^{sign}[\mathbf{j}]$ is abandoned, and only $\kappa^{sig}[\mathbf{j}]$ is selected.

Magnitude refinement pass

The context type in magnitude refinement pass is κ^{mag}, which are context 15, context 16

and context 17. The context is decided by $\kappa^{sig}[\mathbf{j}]$ and $\overleftarrow{\sigma}[\mathbf{j}]$. $\overleftarrow{\sigma}[\mathbf{j}]$ is the significance of current location but delayed by a bit-plane. As a result, κ^{mag} contains information about $\overleftarrow{\sigma}[\mathbf{j}]$ of current location and $\sigma[\mathbf{j}]$ of its neighbours.

Cleanup pass

The context types that cleanup pass has are $\kappa^{sig}[\mathbf{j}]$, $\kappa^{sign}[\mathbf{j}]$, κ^{run} and κ^{uni}. As the significance coding in this pass is almost the same as pass0, the context selection scheme keeps the same that $\kappa^{sig}[\mathbf{j}]$ is chosen whereas $\kappa^{sign}[\mathbf{j}]$ is skipped. κ^{run} and κ^{uni} only appear in the run mode. Furthermore, κ^{uni} is only introduced when run interruption occurs. Thus, κ^{run}, i.e., context 9, is selected to represent run mode.

Overall, there are 9 contexts of $\kappa^{sig}[\mathbf{j}]$, 3 contexts of κ^{mag} and 1 context of κ^{run} are selected to set up new context descriptors, so 13 new contexts are generated. Similarly, these contexts are unique, which means there is only one context matches a given location.

3.4 The feature extraction for low level descriptors

The design of new descriptors have been presented in previous section. The next step is to extract these descriptors from the compressed image and generate feature vectors. A few details about bit-planes and descriptors are discussed first as they are the base of feature extraction. The feature extraction scheme is then introduced, and the scalability of feature vectors are further discussed.

3.4.1 Bit-plane and Proposed Descriptors

The concept of bit-planes have been introduced in previous sections, where code blocks are coded in three coding passes bit-plane by bit-plane. When an image is represented in binary numbers, a bit-plane is a set of bits corresponding to a specific bit position in each pixel. For example, a pixel with maximum value of 255 needs 8-bit representation, and the image consists of such pixels has 8 bit-planes. Figure 3.13 is an example of 8-bit grayscale image

and 8 of its bit-planes.

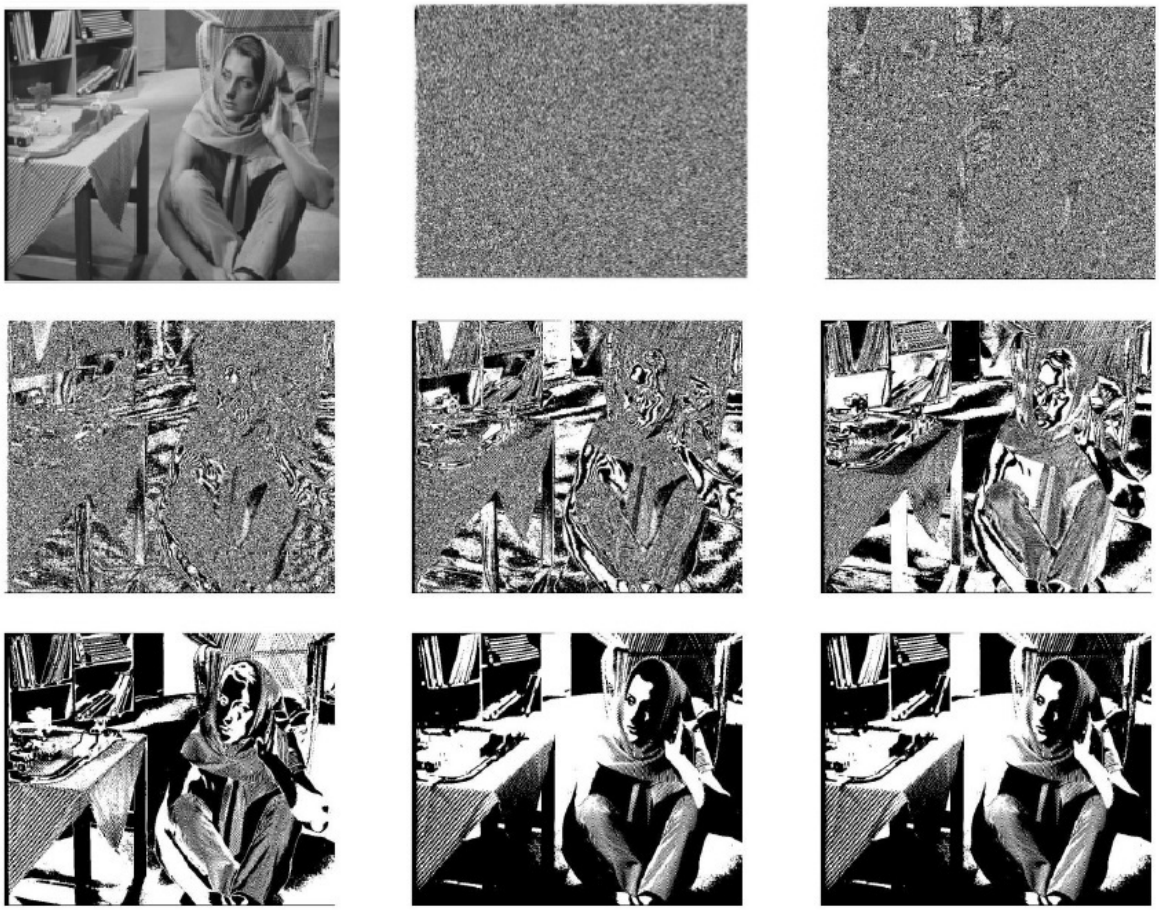

Figure 3.13: 8-bit grayscale image and the 1^{st} , 2^{nd} , 3^{rd} , ... , 8^{th} bit-planes.

It is known that each coefficient bit is coded in only one of three passes during pass coding. In subsection 3.3, it is clear that every possible value of selected symbols and their combinations cover all the possible situations in each pass, i.e., all the possible outcomes are listed as leaf nodes in Figure 3.9, Figure 3.10 and Figure 3.11. Therefore, each bit matches only 1 of 7 proposed data states in Table 3.5. Similarly, each bit matches only 1 of 13 proposed contexts. The proposed data state and context descriptors are acquired bit by bit and bit-plane by bit-plane separately.

One example of data state descriptors and bit-planes are illustrated in Figure 3.14. The first image is the example grayscale image, Barbara. It is compressed with JPEG2000 stan-

dard. During block coding, data state descriptors are acquired when three passes scan each bit of all the bit-planes. All 8 bit-planes are listed in this figure, starting with the most significant bit-plane until the least significant bit-plane. Different data states are presented by different colours which are indicated in the colour bar beside. 7 data states are presented in different colours, and the correspondence between each data state and colour is shown in the side colour bar.

Similarly, an example of contexts of each bit-plane is presented in Figure **??**. A sequence of bit-planes are listed following the original image from the most significant bit-plane to the least significant one. Again, 13 different contexts have their own corresponding colours.

By observing and analysing all the bit-planes, there are a few conclusion. The number of significant samples increases from the highest bit-plane to the lowest bit-plane. There is not much information in the highest bit-plane, but the shape and texture of original image become clear in the middle bit-planes. When it comes to low bit-planes, shape and texture become indistinct, and high level subbands are more serious in particular. In the lowest bit-plane, most samples are already significant. In all, bit-planes in the middle have the most information as the shape and texture of original image are clear, and subbands in different levels are obvious to detect. Therefore, middle bit-planes are important for analysis as they provides adequate information.

3.4.2 Feature Extraction

As long as both data state and context descriptors are acquired. The next step is to extract these features and to organize them into statistics. A feature vector is constructed, so the similarity of two images are measured by calculating the distance between two vectors.

As the pass coding is conducted in bit-plane level, and wavelet transform coefficients are organized in subband structure. The proposed descriptors are also organized in the same pattern. In other words, the extraction and storage of either data states or contexts are conducted in a subbands structure and in a bit-planes order.

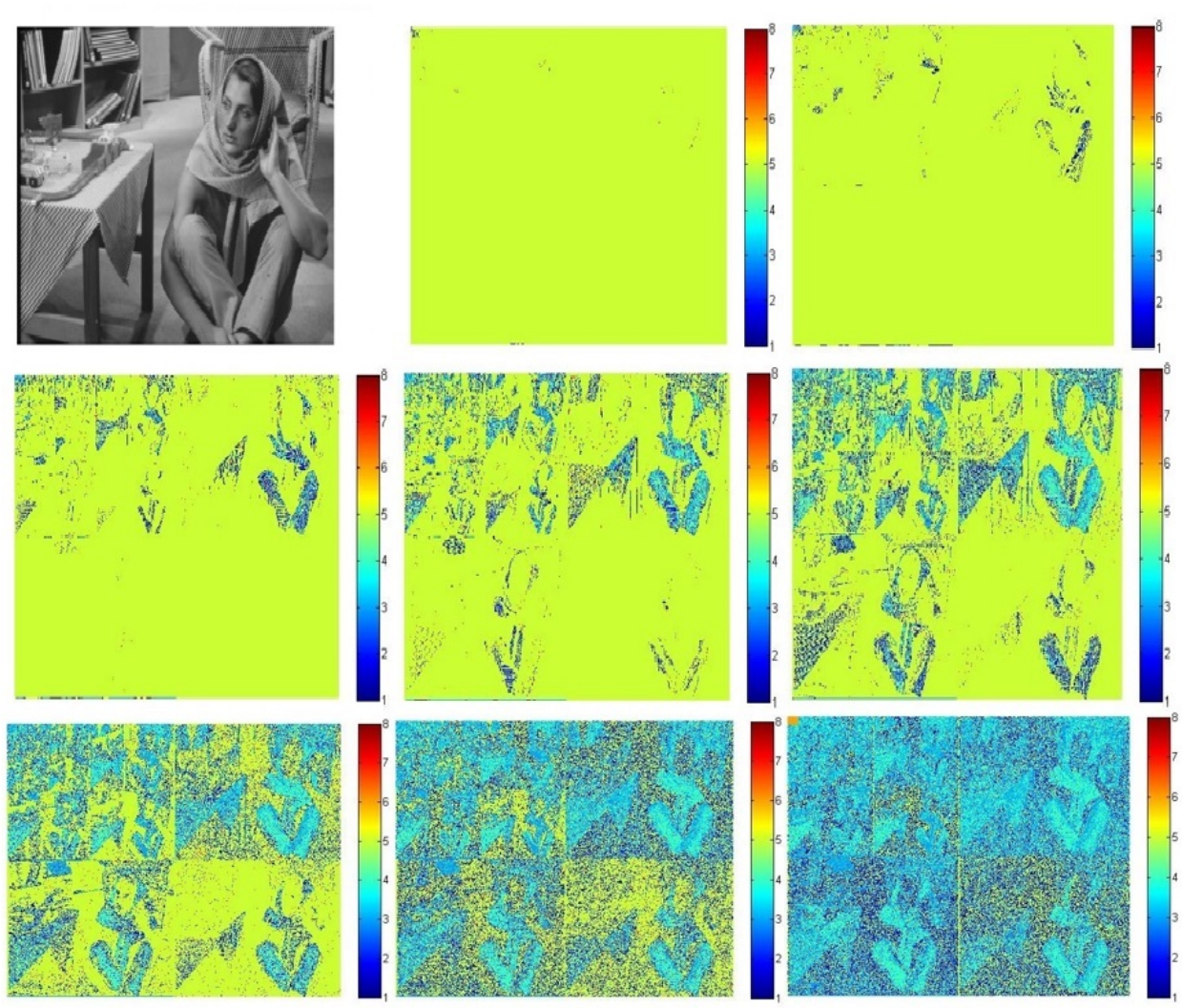

Figure 3.14: Data states of a given image Barbara. The first image is the original image, and the following sequence of images are ordered from the most significant bit-plane to the least significant bit-plane. The correspondence between data state number and colour is shown in the colour bar beside. State 8 has no practical meaning as it is for error detection. The original size versions of these images can be found in the appendix.

A scalable scanning window that is used in [5] is employed for feature extraction because it is a local statistical method that has better retrieval performance than global descriptors. Data states and contexts are extracted and stored separately. A sliding window is used to

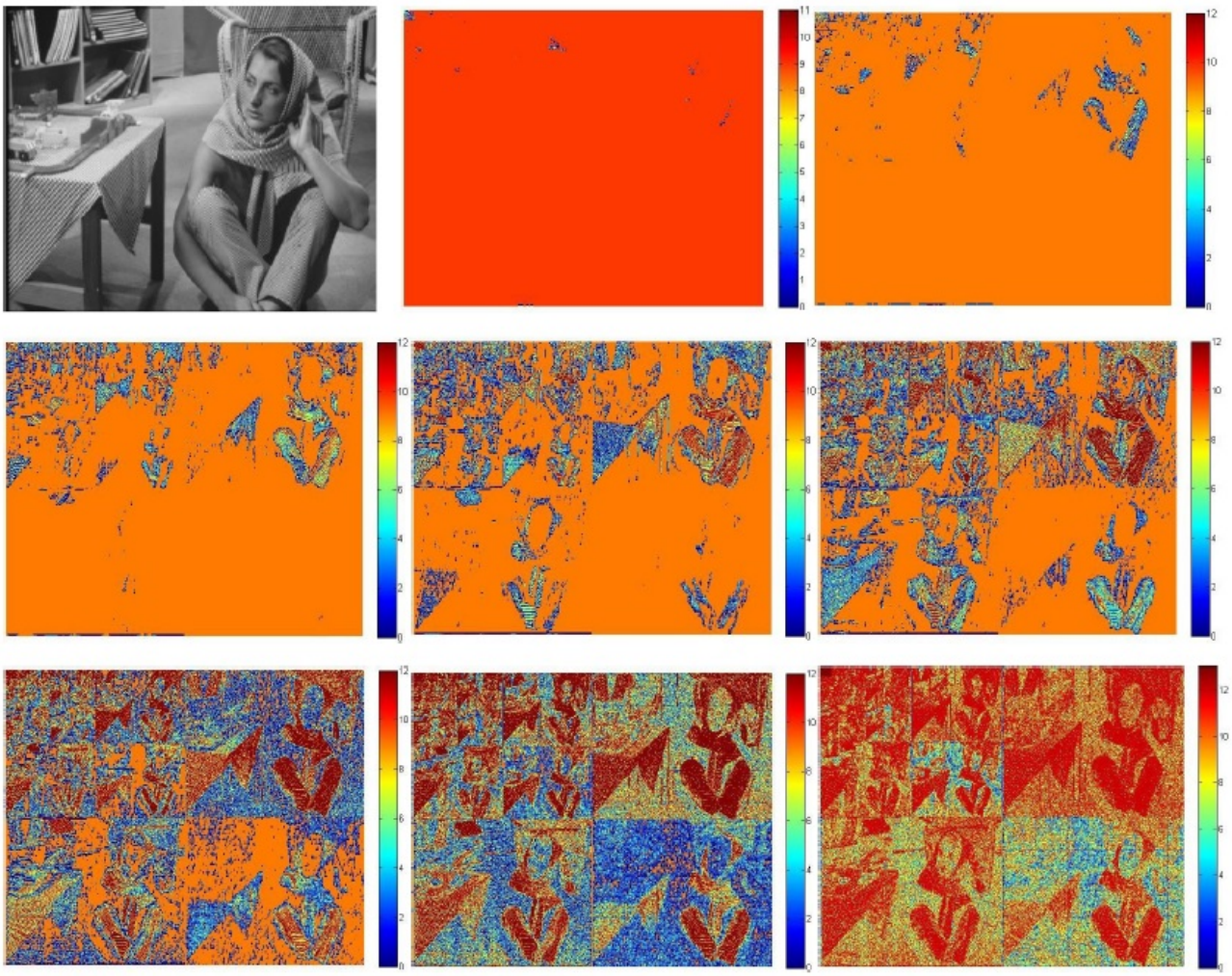

Figure 3.15: Contexts of a given image Barbara. The first image is the original image, and the following sequence of images are ordered from the most significant bit-plane to the least significant bit-plane. 13 different contexts and their corresponding colours are shown in the side bar. The original size versions of these images can be found in the appendix.

scan and extract descriptors [80]. The sliding window is a rectangular structure with variable size of $2^n \times 2^n$, where $n = L, L-1, ..., 1$. L is the level of wavelet transform, and n depends on the level of current subband.

The extraction of data states is a good example. There are 7 data states, and thus a significance histogram $h(n)$ with 7 bins are constructed, where each bin represent one data

state. A 'present or absent' scheme is employed to fill in each bin. The sliding window scans the whole image(bit-plane), starting from the let top corner, row by row, and ends until the whole bit-plane has be scanned completely. Within each sliding window, if a data state is present, then the corresponding bin is increased by one. However, no matter how many times a specific data state appears in current sliding window, the bin value can be incremented only by 1. An example is given in Figure 3.16. Although data state 1(S1) appears twice in the current sliding window, its corresponding bin is incremented by 1 only once. A significance histogram is constructed for each subband, and each bin value of this histogram is normalized by the sum of all the bin values. As long as all the subbands in current bit-plane have been extracted, the procedure moves on to next bit-plane until all the bit-planes are extracted. As a result, a feature vector is generated whose number of elements is $N = B \times [7 \times (3L + 1)]$, where B is the number of bit-planes and L is the number of wavelet transform levels.

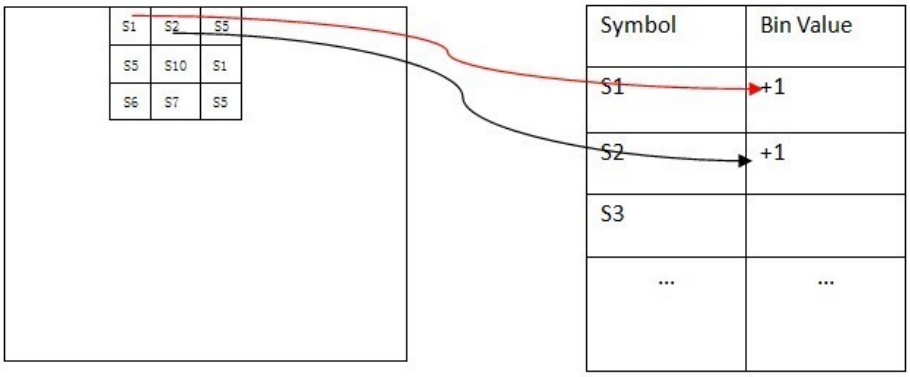

Figure 3.16: Sliding window scan

After the feature vector is generated, L_2 norm distance are employed for similarity measurement. The similarity measurement of two images is achieved by calculating the distance

of each of the elements in corresponding location of two feature vectors. Feature vectors are extracted for all the images in the database, and they are stored separately from the original images. When retrieving, the feature vector of query image is compared with feature vectors of all the other images in the database. The distances are ranked and the best matches are those with the nearest distances. When it comes to context descriptors, the process is the same as data state descriptor. The only difference is that there are 13 contexts instead of 7 data states.

3.4.3 Resolution and Quality Scalability

It has been introduced that multiple scalabilities are achieved by JPEG2000 standard, but only resolution and quality scalability are concerned in this research. The need of resolution and quality scalability comes from daily life. A good instance happens in mobile communication network. When a user retrieves an image with his mobile phone, the query image may be compressed because of the restriction of bandwidth, which means the resolution and quality may be reduced. The proposed method in this research focus on solving such problems by achieving resolution and quality scalability.

Actually, scalability is attained by analysing and truncating feature vectors. The resolution scalability is achieved by discarding wavelet subbands. It is known that the next level subbands is half the size of previous one, so the resolution of image can be decreased by 2 when discarding the lowest level subbands, i.e., LH, HL and HH band. In terms of quality scalability, it varies by the number of bit-planes selected. The more bit-planes selected, the better quality gained. Therefore, in the situation of mobile communication network with limited bandwidth, both resolution and quality can be reduced by discarding subbands and bit-planes.

Many existing methods can discard bit-planes and reduce subbands. However, most of them are not able to keep the same retrieval performance after resolution or quality reduction. For instance, MPEG-7 descriptors are sensitive to compression. The other problem is the time waste during the query image decoding. The initial intention of this research is to solve such

problem, i.e., to design a descriptor that is not sensitive to resolution and quality variation. Moreover, since the feature vectors are already extracted and stored separately, there is no need to decode image during retrieval.

3.5 Summary

In this chapter, JPEG 2000 compression standard is introduced and its block coding procedure are explained in detail. Three bit-plane coding passes are analysed. Each pass handles the wavelet coefficients and contributes to the MQ coder. New low level descriptors, data states and contexts, are designed and developed based on different symbols and contexts appear in three bit-plane coding passes. Data states are designed based on a collection from the MQ coder input symbols and internal states of coding passes, and both magnitude and sign are considered; while contexts are constructed based on the MQ coder input contexts appear in three coding passes. The feature extraction scheme is presented and the scalability of feature vectors are further discussed. Both data states and contexts achieves resolution and quality scalability. In next chapter, the application of proposed features are conducted on practical image database. Experiments work are presented and the results are discussed.

Chapter 4

Image Retrieval Using Proposed Descriptors

The basic mechanism of JPEG 2000 standard and its block coding have been explained in previous chapters. The proposed descriptors and feature extraction are introduced as well. In this chapter, a retrieval system with the proposed descriptors is built up, and this system is applied into practical image retrieval. The image retrieval and experiments are conducted and the results are then analysed.

4.1 Retrieval Application Scenario

In the practical image retrieval scenario, images are stored in database in the format of compressed data streams. When a query image is requested, the feature vectors are extracted from the query image based on the chosen descriptor, like MPEG-7 descriptor for example. The feature extraction is also need to be conducted on all the images in the database. Each image needs to be decompressed completely to extract features, which cost huge computation and too much time.

The retrieval method proposed in this research, however, is able to decompress the compressed image with the least computation and time cost. Instead of decompressing images

completely, the decompression is scalable according to the various compression ratios. For example, if the query image is half resolution and medium quality, compressed images in the database are decoded gradually, i.e., bit-plane by bit-plane and subband by subband, and stop when the target compression ratio has achieved. The feature vectors are extracted and constructed from the decompressed part and the distance can be calculated with the query image at the same time, which is much fast and more efficient than traditional retrieval method.

4.2 Experimental Work and Results

The proposed descriptors and retrieval system are applied to practical experiments. The experimental work and related details like database and evaluation criterion are introduced. The experiments are then conducted and the results are discussed.

4.2.1 Image Database and Ground Truth

An image database of around 400 images [79] are used in this research. These images are divided into 10 classes according to different main objects or foreground objects, such as football, cherries, and so on.

Ground truth is employed to evaluate the retrieval performance. Ground truth in this research is a matrix where each image of the database has 42 tags of description. These tags are high level descriptors such as sky, grass, car etc., and are manually identified by human inspection. Each class are assigned at most 5 tags that ensures a comprehensive description. The more similar two images are, the more tags they match. Thus, retrieval results are evaluated by counting the number of tag matches between query image and other images. Ideally, the rank of image retrieval result should be close to human manual inspection.

4.2.2 Retrieval Performance Evaluation

Results evaluation employs precision-recall curve and ANMRR(average normalized modified retrieval rank). These evaluation standards are introduced as below.

Precision and recall are two basic evaluation methods in image retrieval [3]. Precision is the ratio of the number of relevant images retrieved to the size of retrieved images set. Recall is the ratio of the number of relevant images retrieved to the number of relevant images in the whole databased. Ideally, both precision and recall is supposed to be 100%. However, they are usually inversely related in practical situations. For instance, in order to increase recall, the size of retrieved images set needs to be increase, which leads to the decrease of precision. Similarly, precision increases when the size of retrieved images set decrease, but recall decreases as a result. As there is not enough information when considering precision or recall individually, a precision recall curve is proposed where the horizontal axis is recall, and the vertical one is precision. The precision is measured at different levels of recall. On the other hand, precision and recall depend on the number of retrieved image, i.e., the size of retrieved image set, so single precision or recall carry little information because situation varies when the size of retrieved image set changes. This problem is solved in precision recall curve because multiple sizes of retrieved image set are chosen. Usually, a retrieval descriptor performs better if its precision-recall curve is above curves of other descriptors, which means when recall is the same in most conditions, precision is higher.

Another retrieval evaluation concept is about rank. When two results have the same precision, the one with higher rank is better. Average normalized modified retrieval rank(ANMRR) is a performance evaluation metric based on rank, and it is used in all of the MPEG-7 colour core experiments [5, 54]. The higher ANMRR is, the less accurate retrieval is. Its value is between 0 and 1, where 0 means the whole ground truth were found as the highest ranking images, and 1 indicates none of ground truth images was found. It is the average of NMRR over all queries, the definition of NMRR is given as

$$\text{NMRR}(q) = \frac{\text{AVR}(q) - 0.5(NG(q) + 1)}{1.25K - 0.5(NG(q) + 1)}, \tag{4.1}$$

where $NG(q)$ is the number of relevant images(correct matches) in the retrieved image set of a query q, and

$$\text{AVR}(q) = \frac{1}{NG(q)} \sum_{k=1}^{NG(q)} rank^*(k), \tag{4.2}$$

where

$$f(x) = \begin{cases} rank(k) & \text{if } rank(k) \leq K(q), \\ 1.25K(q) & \text{if } rank(k) > K(q). \end{cases} \tag{4.3}$$

In this function, $rank(k)$ means the rank of relevant images(correct matches), and $K(q)$ is a threshold defining a window of results. $K(q) = min(2GTM, 4NG(q))$, where GTM is the largest $NG(q)$ over all query images, and 1.25 is the penalty factor for matches that fall outside of the window of results. Based on NMRR, ANMRR is given over all queries

$$\text{ANMRR} = \frac{1}{NQ} \sum_{q=1}^{NQ} \text{NMRR(q)}, \tag{4.4}$$

where NQ is the number of query images.

In general, AVR is the average rank. As $NG(q)$ varies in different data set, MRR(Modified Retrieval Rank) is used to reduce the influence of $NG(q)$ to evaluation function. MRR is then normalized because its limit inferior is affected by NG, and therefore NMRR(Normalized Modified Retrieval Rank) is generated. Finally, ANMRR is the average of NMRR over all NQ query images.

4.2.3 Experimental Work

The purpose of the experiment is to test whether the proposed descriptors perform well in the actual image retrieval. Furthermore, if they are invariant to resolution and quality scalability are concerned. The performance of proposed methods are compared with the popular state of art approach, MPEG-7 descriptors.

Based on the ideas above, both image resolution and quality have 3 different versions.

In terms of resolution, different resolution versions are acquired by dividing both width and height by 2^n, where n decided the level of resolution. When resolution is full, $n = 0$, which means that it is the same size as the original image; when the resolution is half, $n = 1$, which means both width and height are half the length of original version; When resolution is quarter, $n = 2$, which means both width and height are one quarter of the original version.

When it comes to quality, different versions of quality are achieved by discarding part of bit-planes. When quality version is high, all the bit-planes are kept; when quality version is medium, 3 most significant bit-planes are discarded; when quality version is low, 6 most significant bit-planes are discarded.

The combinations of resolutions and qualities are: full resolution and high quality(RFQH), half resolution and medium quality (RHQM), quarter resolution and low quality (RQQL), full resolution and medium quality (RFQM), full resolution and low quality (RFQL), half resolution and high quality (RHQH), half resolution and low quality (RHQL), quarter resolution and high quality (RQQH) and quarter resolution and medium quality (RQQM).

Each image in the database is chosen as query image. A retrieval is conducted for query image. The average precision, average recall and ANMRR are calculated. There are 20 different selection of the size of retrieved image set, and a precision-recall curve is able to be drawn.

The same work are repeated with MPEG-7 descriptors, i.e., CLD(colour layout descriptor), SCD(scalable colour descriptor) and EHD(edge histogram descriptor). The first two descriptors are colour descriptors, and the last one is texture descriptors. Feature vectors are

generated by CALIPH(common and light weight photo annotation) software [81], and three different versions, i.e., full resolution and high quality, half resolution and medium quality, and quarter resolution and low quality, are generated. L2 norm distance is employed to measure the similarity for CLD and EHD, and L1 norm distance is used for SCD [5]. Their performances are also evaluated by precision, recall and ANMRR. A comparison are then made between proposed data state and context descriptors and MPEG-7 descriptors.

One thing needs to be notice is that only Y component is used for feature extraction and image retrieval, which can be considered as a grayscale version of original image.

4.2.4 Results

Data state descriptor

For a given query image, cambridge73, Figure 4.1 to Figure 4.3 shows the retrieval results. The size of retrieved image set is 11 for the display purpose. Three typical compression rate are selected for illustration. The query image in Figure 4.1 is full resolution and high quality. In Figure 4.2, the query image is half resolution and medium quality; while in Figure 4.3, resolution and quality of query image become quarter and low. For the purpose of clear demonstration, the size of query image in all the figures keep the original version. This example shows a great consistency of the proposed descriptor against resolution-quality adaptation.

Figure 4.4 are the results in terms of precision-recall curves of proposed descriptor. Three typical versions, namely resolution full and high quality (the original version), resolution half, and medium quality and resolution quarter and low quality, are chosen for demonstration. The variation of retrieval performance against content adaptation is low, and a better consistency is shown in half resolution and medium quality level than quarter resolution and low quality level.

More experiments are conducted to find out whether resolution or quality has more effect on performance. Resolution keeps full in Figure 4.5, while quality has three versions: high

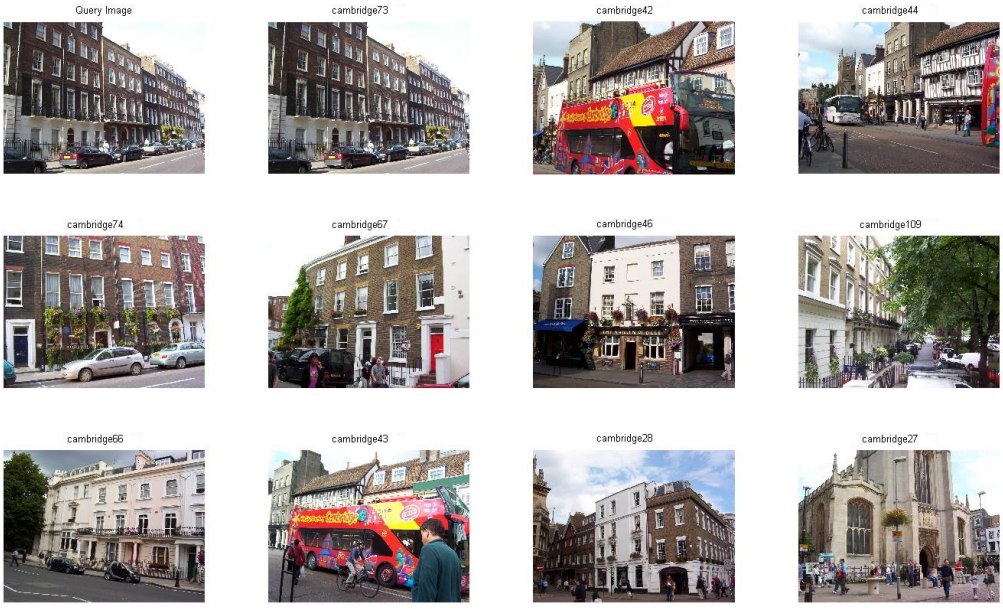

Figure 4.1: Retrieval results of query image(full resolution, high quality) cambridge73.

,medium and low. In Figure 4.6, everything keeps the same except resolution reduced to low. According to these two figures, the reduction of quality causes no significant decrease of performance while the resolution keeps same.

In Figure 4.7, quality keeps high but resolution has three different versions: full, half and quarter, and in Figure 4.8 quality is low while other things keeps the same. The performance keeps in a good level when resolution drops; and this relationship does not change even when quality decreases. Compared with Figure 4.5 and Figure 4.6, it seems resolution has more effect on the performance when quality keeps the same than quality varies while resolution stays the same.

According to the experiments, the proposed data state descriptor is scalable to resolution and quality. The reduction of either resolution or quality has no obvious influence on the performance. The ideal compression rate is 50%, but quarter compression can still give good result.

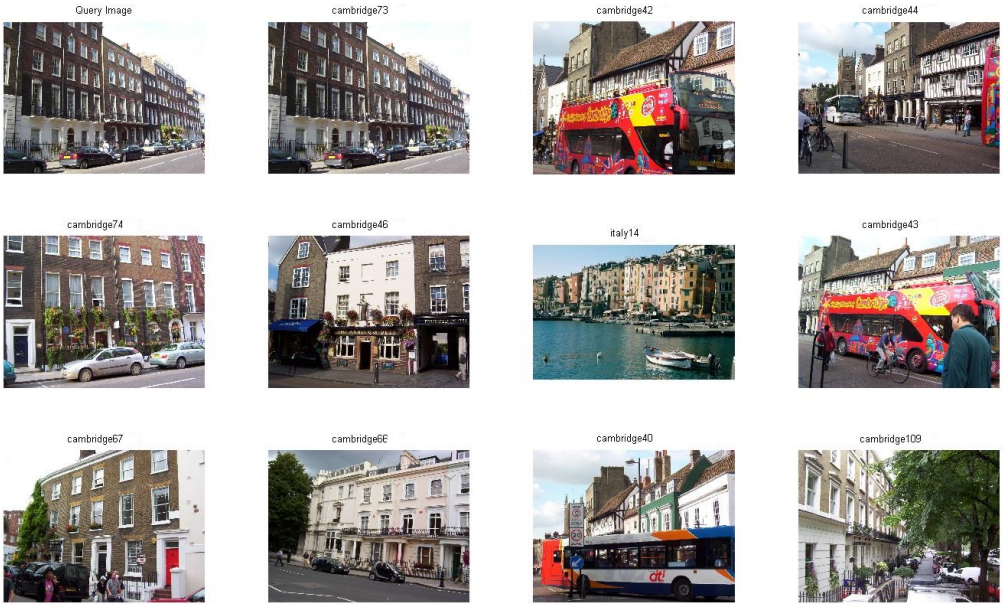

Figure 4.2: Retrieval results of query image(half resolution, medium quality) cambridge73.

Context descriptor

The retrieval results of the same query image, cambridge73, is given in Figure 4.9 to Figure 4.11, where the size of retrieved image set is 11 as well. Three typical compression rates are selected for illustration. The query image in Figure 4.1 is full resolution and high quality. In Figure 4.10, the query image is half resolution and medium quality; while in Figure 4.11, resolution and quality of query image become quarter and low, respectively. For the purpose of clear demonstration, the size of query image in all the figures keep the original version.

According to the figures, a good consistency of retrieved matching images are presented against resolution-quality reduction. Figure 4.12 is a precision-recall curve of context descriptors. Three curves represent three different versions of image resolution and quality. They are full resolution and high quality(RFQH), half resolution and medium quality(RHQM) and quarter resolution and low quality(RQQL). The performance generally keeps in a good level

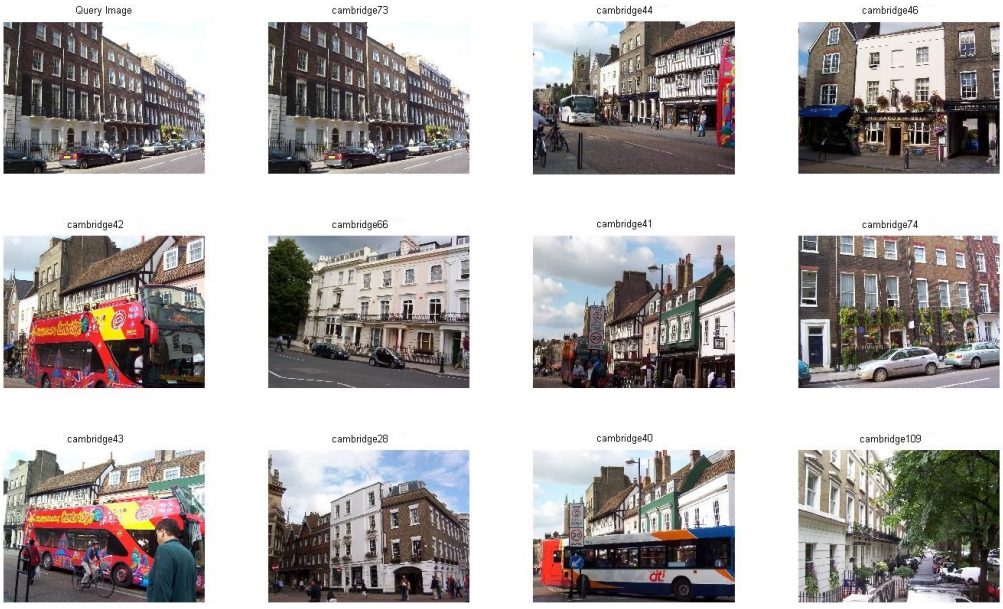

Figure 4.3: Retrieval results of query image(quarter resolution, low quality) cambridge73.

even when both resolution and quality are only quarter of the original image. Moreover, the performance is not influenced less when resolution and quality dropped by half than by 75%.

Experiments have been done in Figure 4.13 to analyse the influence of resolution reduction and quality reduction. The first curve(RFQH) is a reference. Resolution and quality are reduced by half and three fourth in turns in the rest four versions. However, according to Figure 4.13, the rest four curves are actually close to each other, which means it is hard to conclude whether quality or resolution has more effect on the performance.

A comparison is made in Figure 4.14 between data state descriptor and context descriptor. Context is slightly better in RFQH and RHQM situation, but data state outperform context when compression drops into quarter. On the whole, their performance are similar and both of them are robust against resolution and quality variance. When the compression rate is less than 50%, the retrieval performance can be almost as good as the original images.

Table 4.1 lists ANMRR of data state and context in 9 different compression versions.

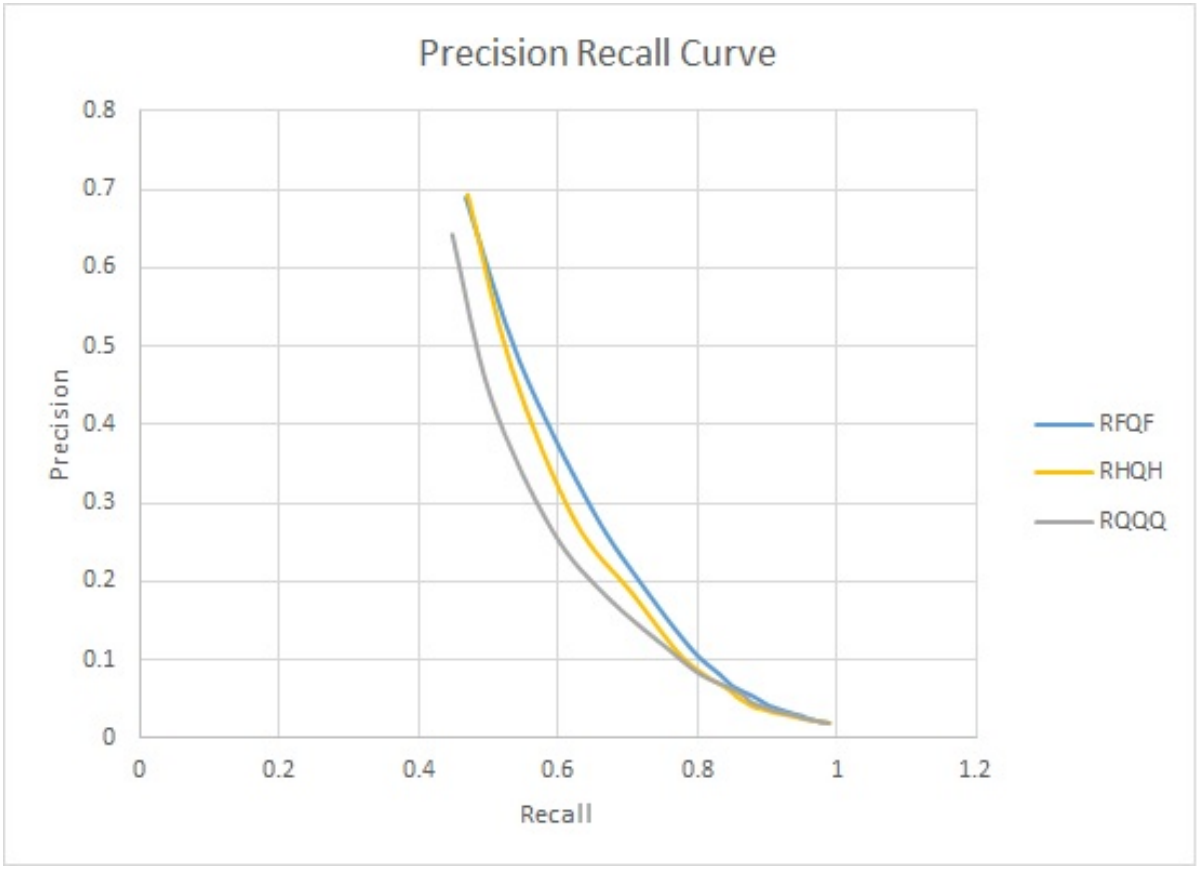

Figure 4.4: Precision-recall curve of Data state descriptor. There are 3 different versions: full resolution and high quality(RFQH), half resolution and medium quality(RHQM), and quarter resolution and low quality(RQQL)

As the resolution and quality decrease, ANMRR increases, which means the performance is getting worse. It seems the reduction of resolution has more effect on ANMRR, because when resolution keeps in the same level, the reduction of quality influences ANMRR. However, this influence is small compared with the increase of ANMRR caused by resolution reduction when quality is in the same level. Finally, context generally has a lower ANMRR than data state, which means context is better in terms of ranking.

Mid bit-planes

It is mentioned in 3.4.1 that mid bit-planes are important for analysis as they contain

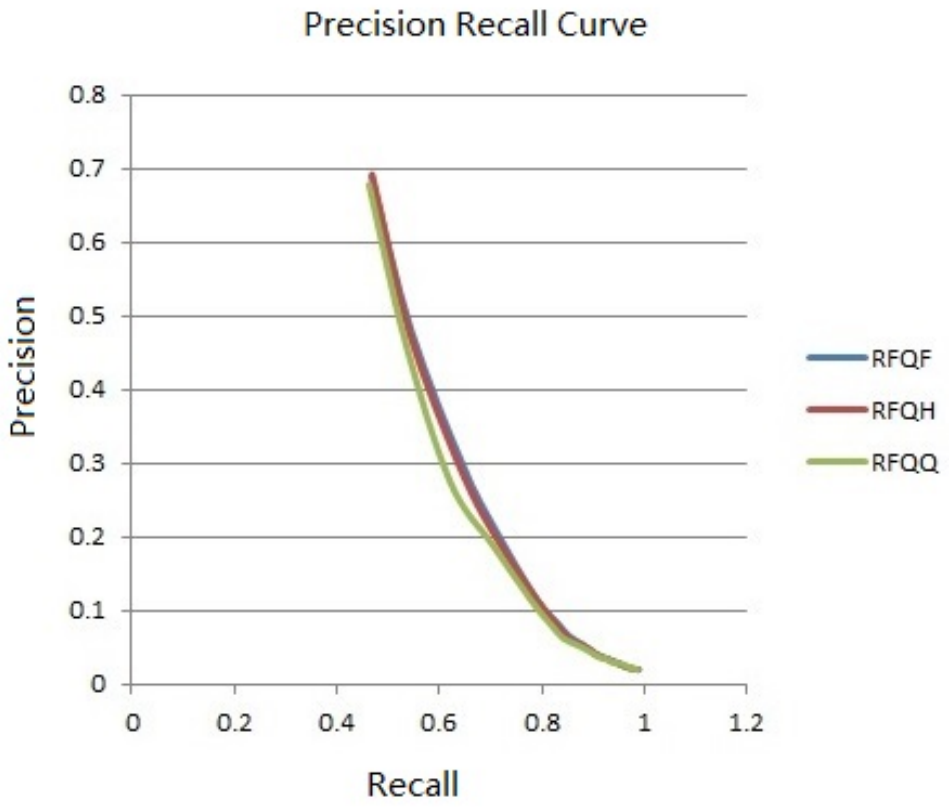

Figure 4.5: Precision-recall curve of data state descriptor. The resolution keeps full, but there are 3 different quality versions, i.e., full resolution and high quality(RFQH), full resolution and medium quality(RFQM), and full resolution and low quality(RFQL)

adequate information. Thus, experiments are conducted with only mid bit-planes. By investigating Figure 3.14 and Figure **??**, the third, fourth and fifth bit-planes are the ideal selection, as higher bit-planes do not contain enough information in the majority of subbands, and lower bit-planes have too much information. Precision-recall curves are shown from Figure 4.15 to Figure 4.17, and ANMRR are listed in Table 4.3.

Figure 4.15 is a comparison between data state and context with only mid bit-planes. Three resolution are chosen: full, half and quarter. It is clear that context descriptor outperforms data state one. Context overwhelms data state when only mid bit-planes are used,

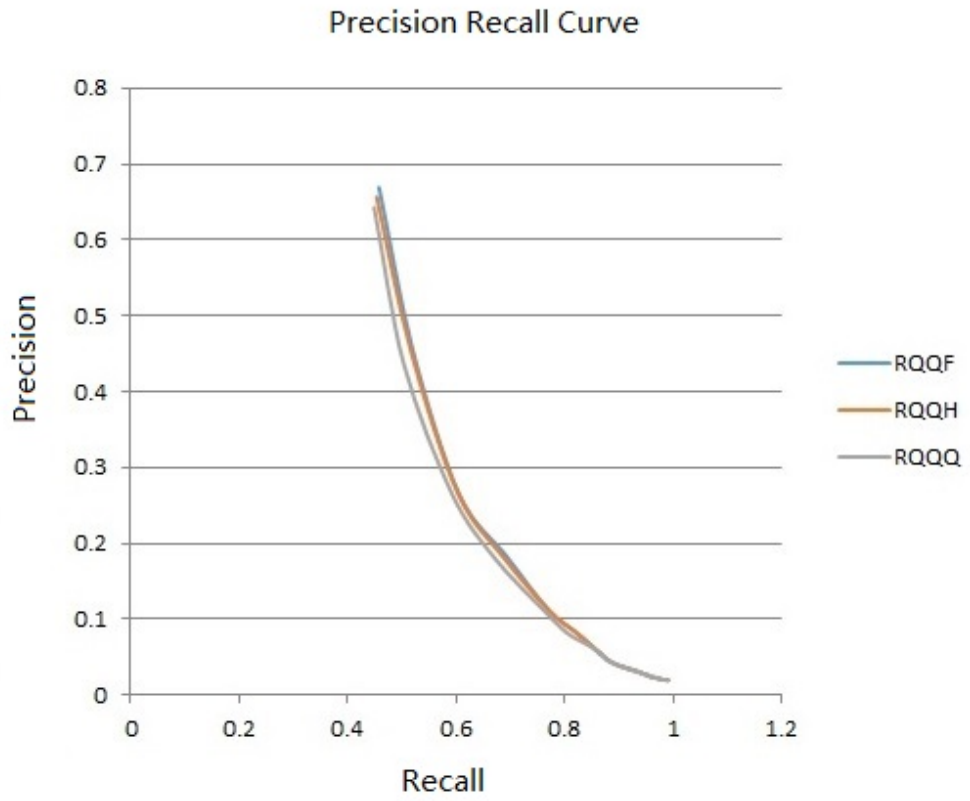

Figure 4.6: Precision-recall curve of data state descriptor. The resolution keeps quarter, but there are 3 different quality versions, i.e., quarter resolution and high quality(RQQH), quarter resolution and medium quality(RQQM), and quarter resolution and low quality(RQQL)

because even the best data state(full resolution version) cannot beat the worst context(quarter resolution).

In Figure 4.16, data state with only mid bit-planes are compared with data state with full resolution high quality, half resolution medium quality and quarter resolution and low quality. The performances of data state with only mid bit-planes are worse than any combination of resolution and quality. The results indicate that the retrieval performance of data state descriptor decreases when only mid bit-planes are engaged.

In Figure 4.17, the situation of context is different. The performance of context with

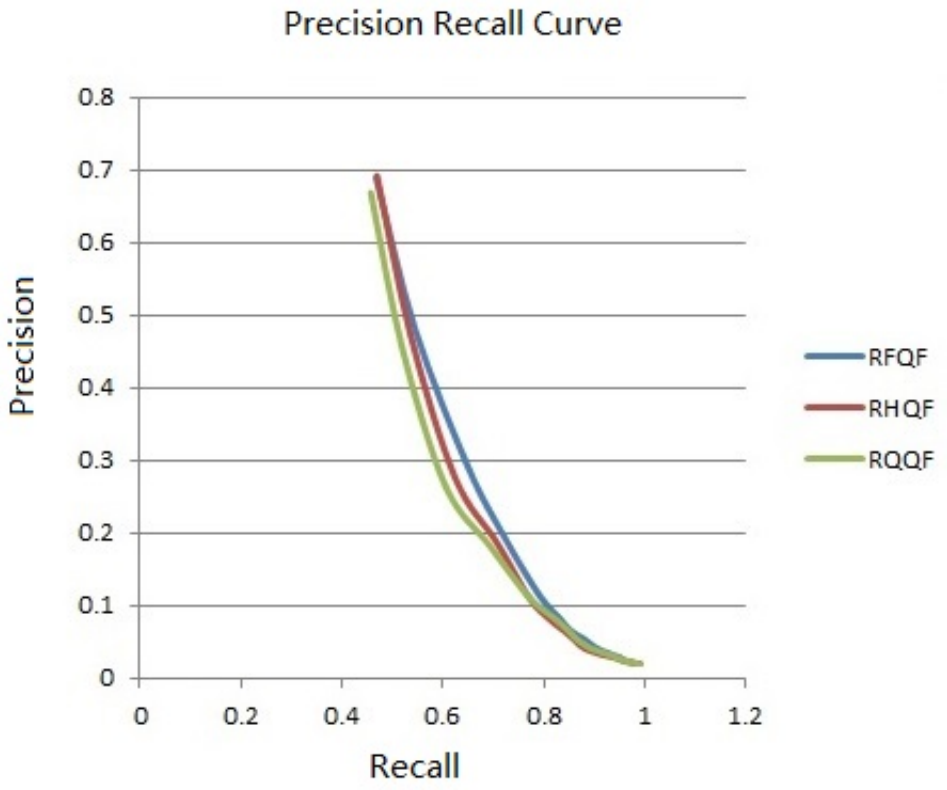

Figure 4.7: Precision-recall curve of data state descriptor. The quality keeps full, but there are 3 different resolution versions, i.e., full resolution and high quality(RFQH), half resolution and high quality(RHQH), and quarter resolution and high quality(RQQH)

only mid bit-planes are worse when the resolution is full and half. In quarter resolution, the performance with only mid bit-planes and the least quarter significant bit-planes are very similar.

Table 4.2 includes ANMRR of data state with mid bit-planes and context with mid bit-planes. In terms of ANMRR, context with mid bit-planes outperforms data state in any resolution level. Compared with regular resolution quality combination, context with mid bit-planes are closer in all resolutions.

In all, the performances for both data state and context descriptors drop when only mid bit-planes are engaged for retrieval. Context performs better in all resolution levels. When

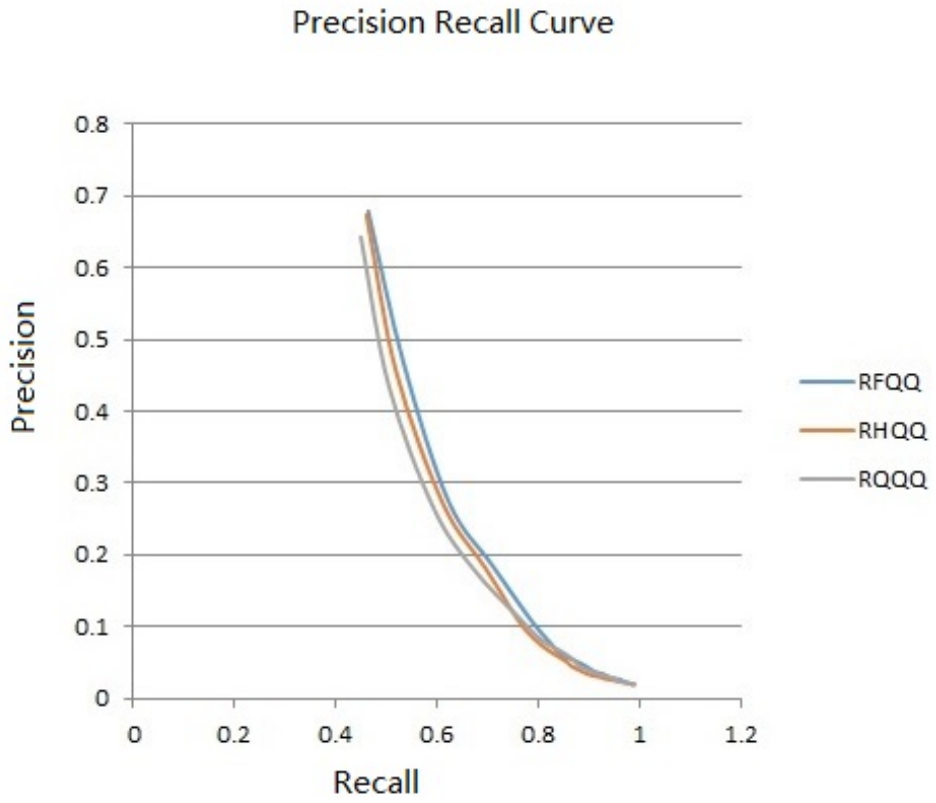

Figure 4.8: Precision-recall curve of data state descriptor. The quality keeps quarter, but there are 3 different resolution versions, i.e., full resolution and low quality(RFQL), half resolution and low quality(RHQL), and quarter resolution and low quality(RQQL)

resolution is full and half, context with only mid planes are not as good as all bit-planes(high quality) and half bit-planes(medium quality) respectively. When it comes to quarter resolution, however, mid bit-planes has almost the same performance with quarter bit-planes(low quality). Therefore, context is a good choice if only mid bit-planes are used for retrieval and analysis.

Comparison between proposed descriptors and MPEG-7 descriptors

Precision-recall curves are shown in Figure 4.18, Figure 4.19 and Figure 4.20. There are five curves in each figure, where three of them are SCD, CLD and EHD, and proposed data

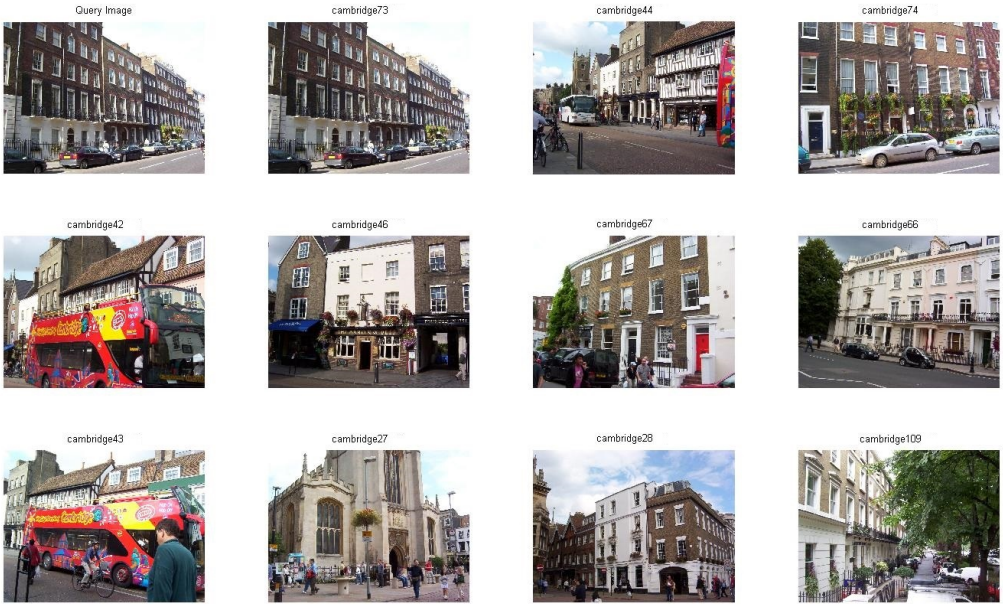

Figure 4.9: Retrieval results of query image(full resolution, high quality) cambridge73.

state and context descriptors are also listed for comparison.

In Figure 4.18, the query image is full resolution and high quality. Data state and context descriptors outperform CLD and EHD, but SCD has a better performance. The performance of CLD and EHD are alike. In Figure 4.19, the query image is half resolution and medium quality. SCD still performs the best, and the performance of data state and context descriptors are steady. CLD outperforms EHD, and EHD is the worst one. In Figure 4.20, the query image is quarter resolution and low quality. The situation is obviously different. Data state and context descriptors still have steady performance and become the best overall. However, the curve of SCD drops dramatically which means this descriptor is sensitive to the reduction of resolution and quality, and performs badly when compression rate is high. The second worst one is EHD which is not robust against resolution and quality variance as well. Although CLD is scalable to resolution and quality, its performance is not as good as either data state or context descriptors.

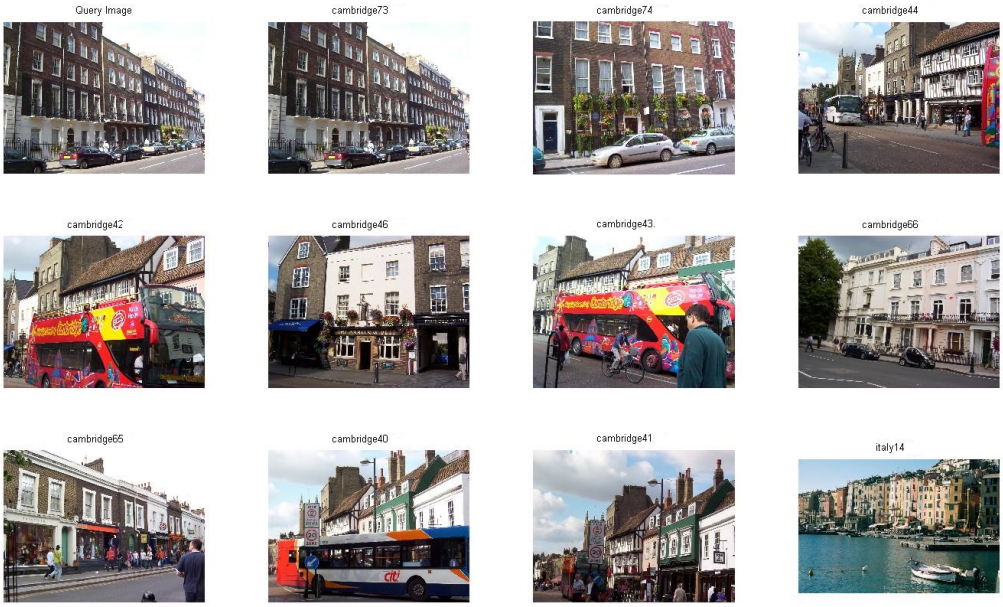

Figure 4.10: Retrieval results of query image(half resolution, medium quality) cambridge73.

Table 4.3 presents ANMRR of data state, context, SCD, CLD, EHD. Similarly, SCD has the best performance when RFQH and RHQM. However, its ANMRR increases significantly when RQQL which becomes the worst over all descriptors. Generally, data state and context has the most stable and best performance over all three resolution and quality levels.

Comparison between proposed descriptors and SIFT

The proposed method is compared with the famous state of art SIFT method. Precision-recall curves are shown in Figure 4.21, Figure 4.22 and Figure 4.23, where the query image is full resolution and high quality, half resolution and medium quality and quarter resolution and low quality, respectively. In all three cases, either data stat and context descriptors outperform SIFT.

Table **??** is the ANMRR of data state, context and SIFT. Context has a better performance than state in full resolution half quality and high resolution medium quality cases. Moreover, both proposed methods are superior to SIFT in terms of ANMRR.

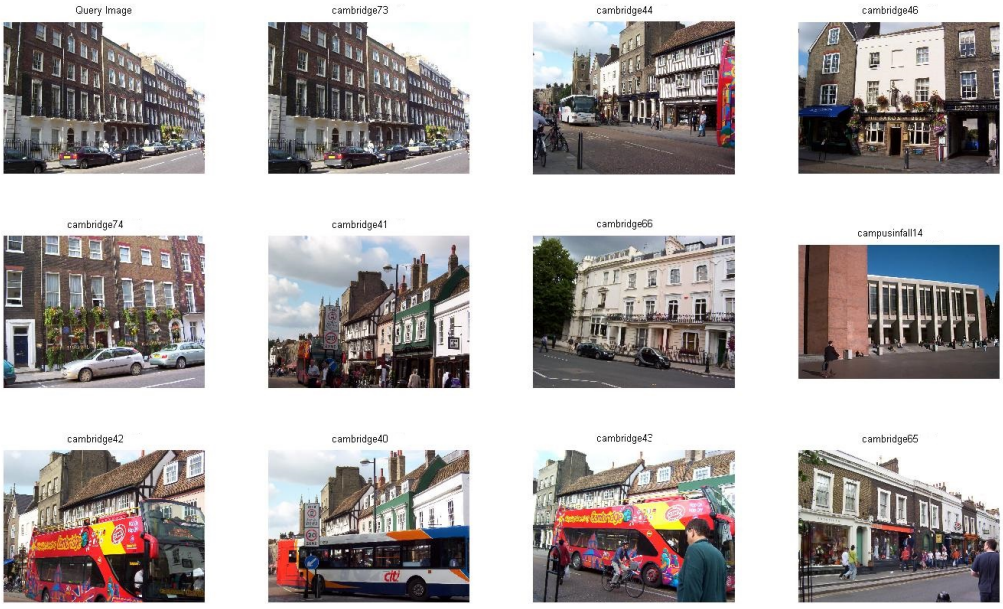

Figure 4.11: Retrieval results of query image(quarter resolution, low quality) cambridge73.

4.2.5 Discussion

In this chapter, a retrieval system is built up based on proposed data state and reliable descriptors, and experimental work are conducted to analyse and evaluate proposed methods. Precision-recall curve and ANMRR are used for performance evaluation, and proposed descriptors are compared with state-of-art MPEG-7 descriptors.

First, either data state descriptor or context descriptor is robust and steady for image retrieval as they provide excellent precision-recall curve and ANMRR results in various conditions. Second, the proposed descriptors are robust against the change of resolution and quality. They perform steadily even in low resolution and quality level. Other MPEG-7 descriptors might have good performance when the resolution and quality are high, but the outcomes drops dramatically when resolution and quality are low. The curves of CLD are relatively more stable than others, but they are not as good as proposed descriptors.

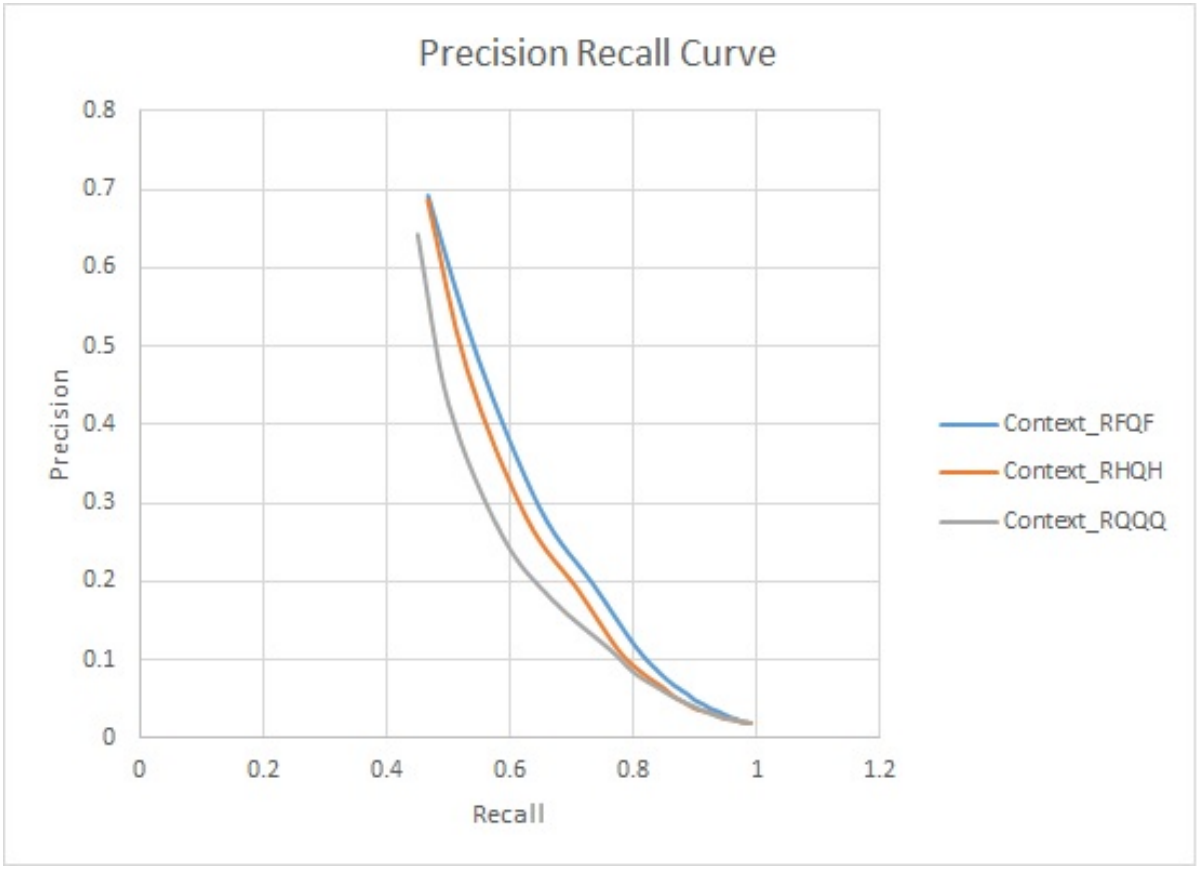

Figure 4.12: Precision-recall curve of Context descriptor. There are 3 different versions: full resolution and high quality(RFQH), half resolution and medium quality(RHQM), and quarter resolution and low quality(RQQL)

One thing need to be notice is that MPEG-7 descriptors extract features from multiple channels. For instance, SCD adopts a HSV colour space and there are a specific ratio among these components. The colour space CLD employs YCrCb colour space. In this research, however, only Y component(grayscale image) is used so far, but it is able to outperform CLD and EHD in all compression levels, and outperform SCD in terms of resolution and quality scalability. The remaining colour component will be used for analysis in the future which will probably achieve better result.

EHD is texture descriptor that concentrate on directional edges. There are five types of

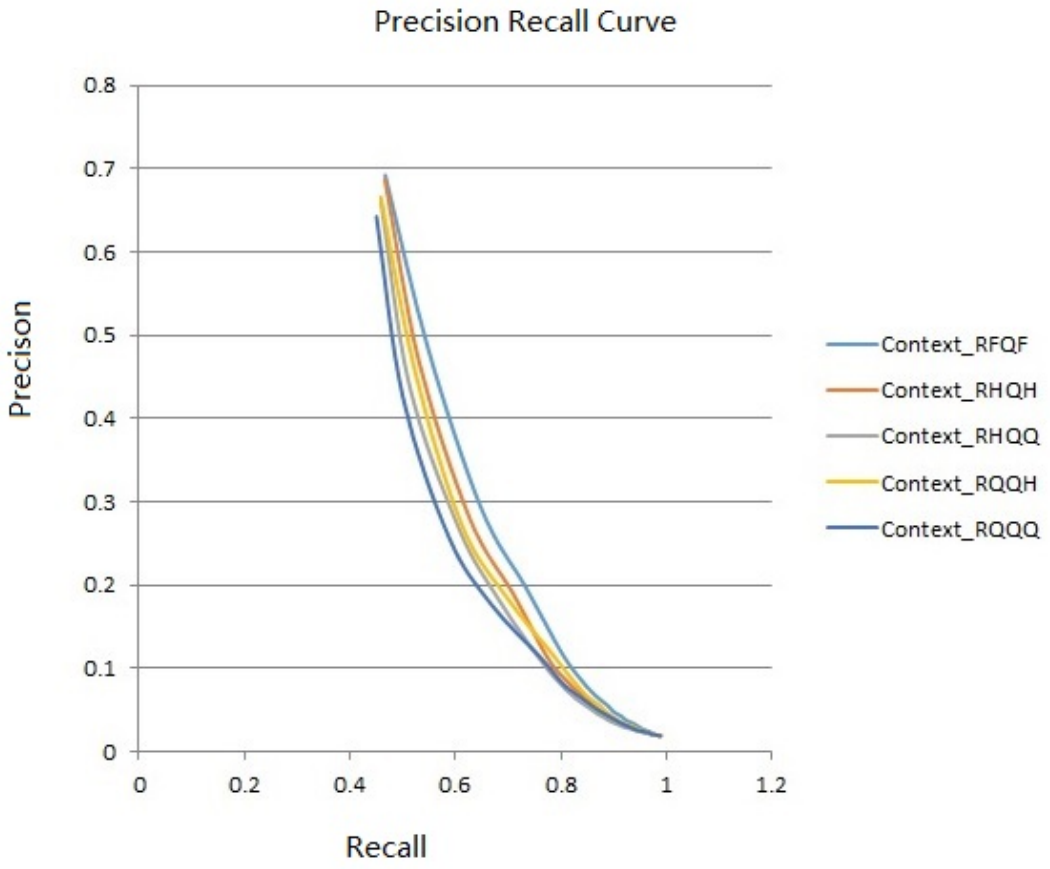

Figure 4.13: Precision-recall curve of Context descriptor. There are 5 versions: The first version is full resolution and high quality(RFQH). The next two versions keep resolution half, and quality is medium(RHQM) and low(RHQL) respectively. The last two versions keep resolution half, and quality is medium(RQQM) and quarter(RQQL) respectively.

edges: horizontal, vertical, 45° diagonal, 135° diagonal and isotopic. As wavelet transform are adopted in the JPEG 2000 compression standard, different subbands are directional [73]. HL subband has horizontal directionality, while LH are vertical and HH are diagonal, which is similar to EHD. From this point of view, proposed descriptors have something in common with EHD. Both of them are directional descriptors. In terms of retrieval performance, data

Table 4.1: ANMRR of data state(S), context(C). There are nine compression versions: full resolution high quality(RFQH), full resolution medium quality(RFQM), full resolution low quality(RFQL), half resolution high quality(RHQH), half resolution medium quality(RHQM), half resolution low quality(RHQL), quarter resolution high quality(RQQH), quarter resolution medium quality(RQQM) and quarter resolution low quality(RQQL)

	RFQH	RFQM	RFQL	RHQH	RHQM	RHQL	RQQH	RQQM	RQQL
S	0.281	0.288	0.302	0.302	0.303	0.317	0.317	0.318	0.329
C	0.275	0.285	0.304	0.287	0.297	0.322	0.295	0.307	0.329

state descriptor or context descriptor are better choices.

The proposed descriptors outperform state of art SIFT method in high, medium and low resolution and quality levels in terms of precision recall curve and ANMRR. SIFT has advantages in situations like object detection [6, 29]. However, when it comes to scalable coding and image retrieval with adapted content, the performance and consistency is not competitive with proposed methods.

Mid bit-planes only is a good selection when there is a limitation of quality, but the trade-off is loss of performance. Instead of data state, context is a good choice for only mid bit-planes as it provides better and stable performance. The performance of context with mid bit-planes only is not as good as either the performance of full bit-planes or half bit-planes, but quite similar to quarter bit-planes.

The computational complexity of one query image is $O(n)$, where n is the number of images in the database. When a query image is inputted, the system selects the corresponding part of feature vector from each image in the database for matching. On other words, feature vectors of irrelevant bit-planes and subbands are discarded. The first step has the complexity of $O(1)$. Query image feature vector is then compared one by one with those of images in database, and then L2 norm distance are calculated. The complexity of these procedures is $O(n)$.

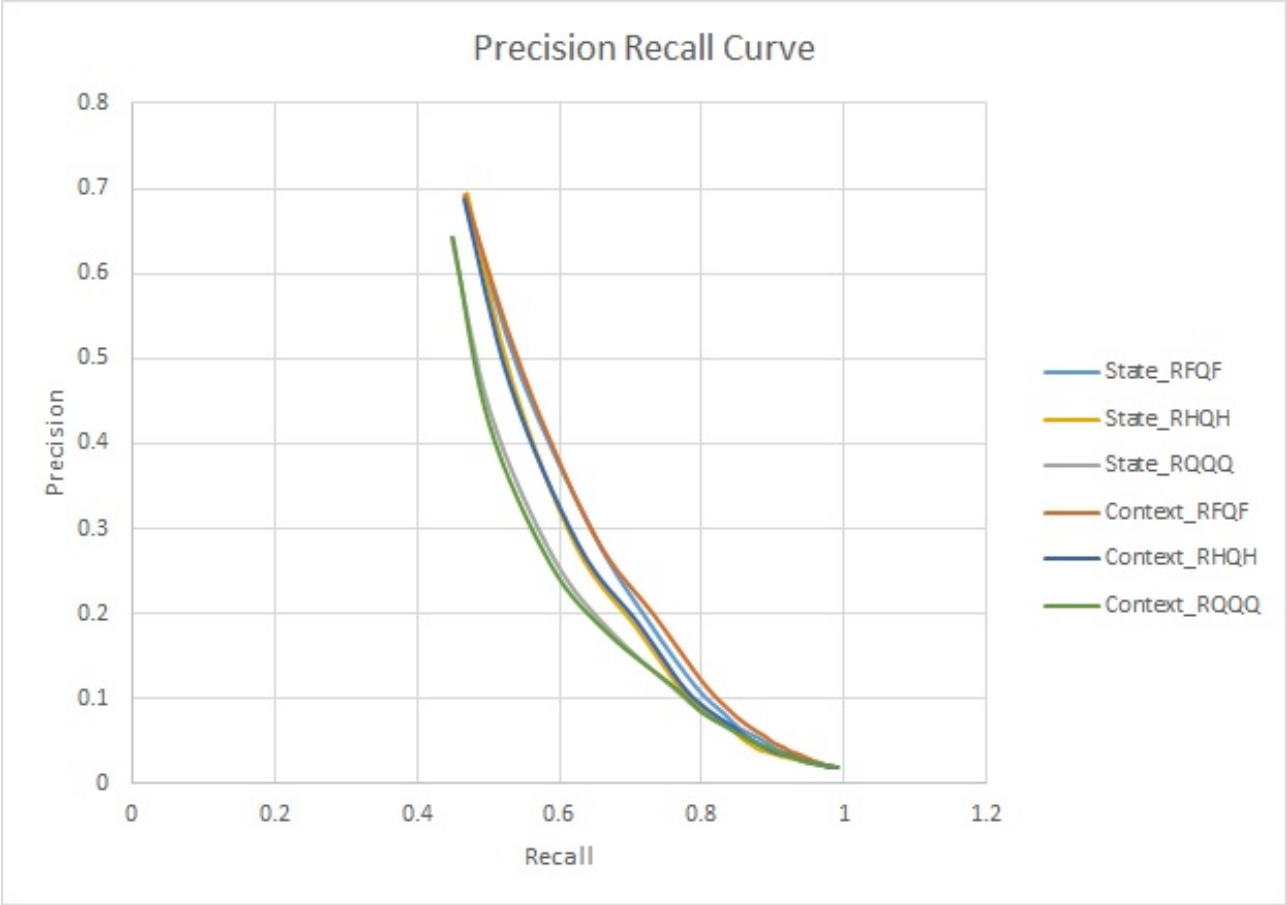

Figure 4.14: Precision-recall curve: A comparison between data state descriptor and context descriptor. Both of them have three different versions: full resolution and high quality(RFQH), half resolution and medium quality(RHQM), and quarter resolution and low quality(RQQL)

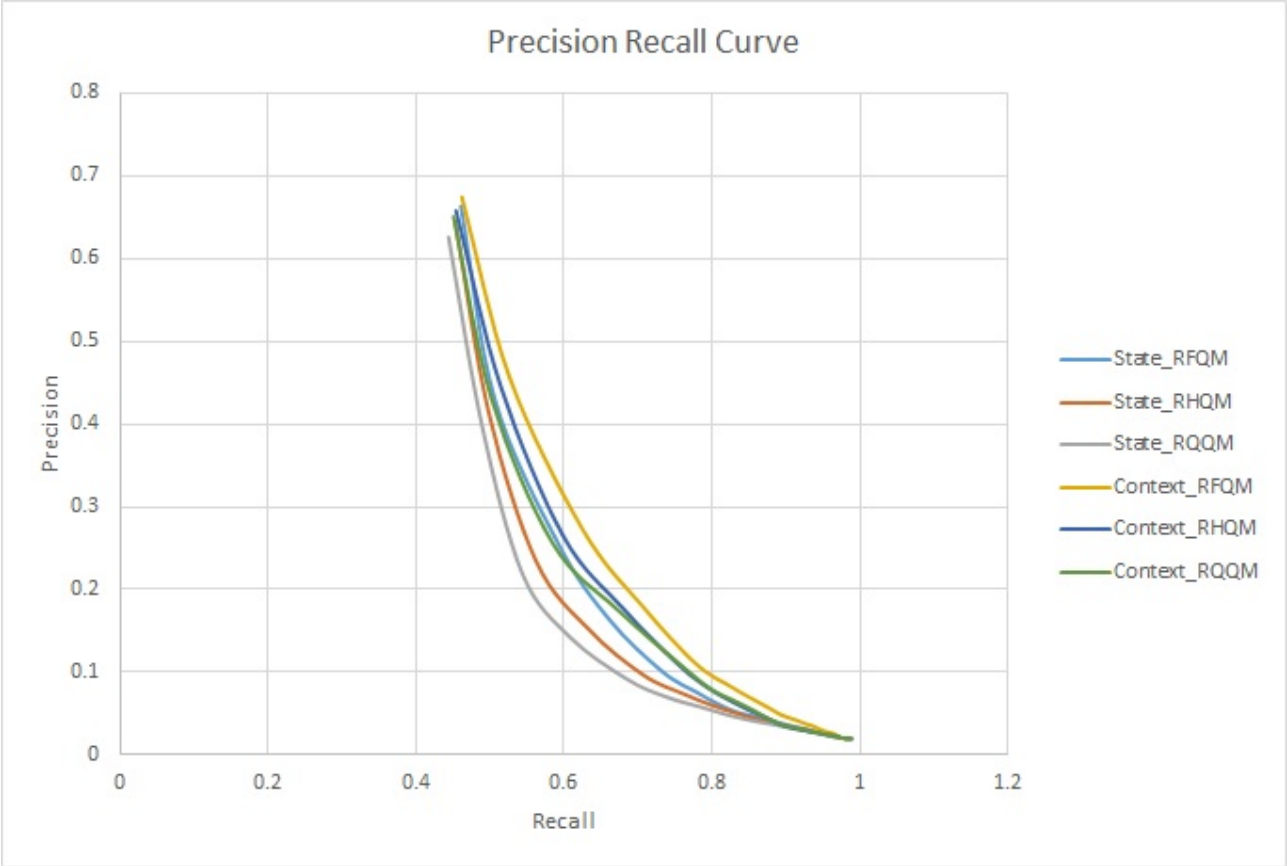

Figure 4.15: Precision-recall curve: data state and context with only mid bit-planes. There are three resolution versions: full resolution, half resolution, and quarter resolution.

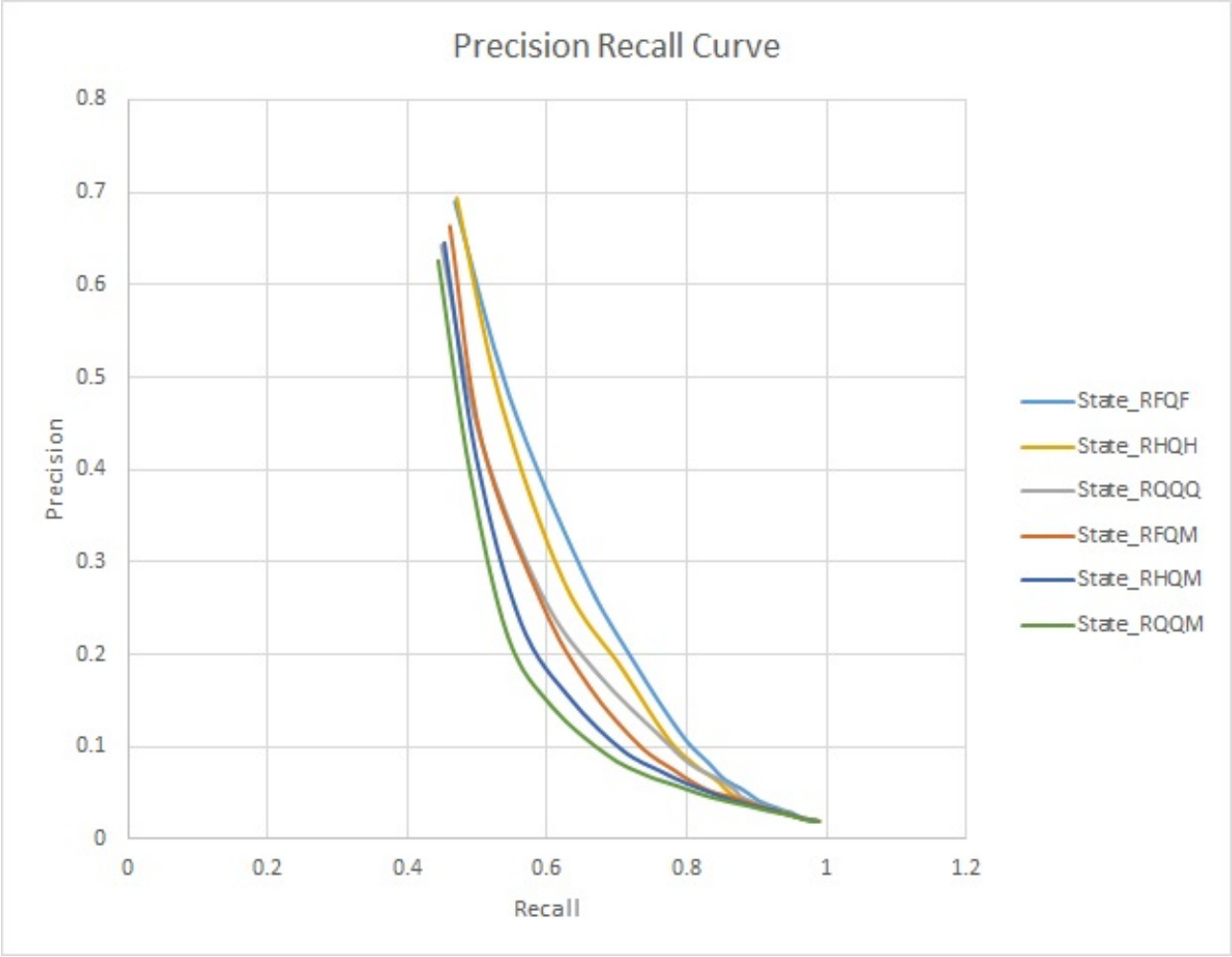

Figure 4.16: Precision-recall curve: data state with full resolution high quality(RFQH), half resolution medium quality(RHQM) and quarter resolution and low quality(RQQL) are compared with data state with only mid bit-planes. The latter has three resolution versions: full resolution, half resolution, and quarter resolution.

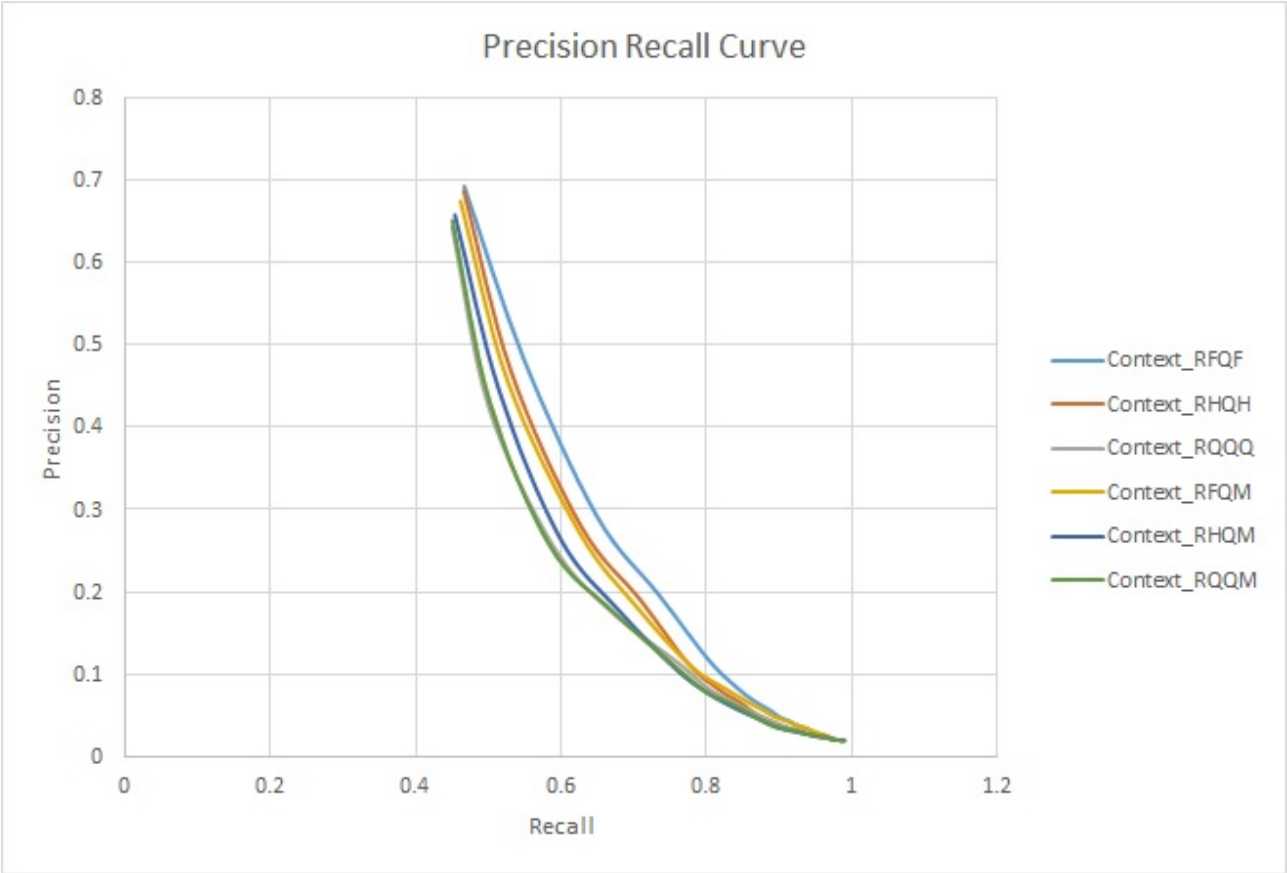

Figure 4.17: Precision-recall curve: Context with full resolution high quality(RFQH), half resolution medium quality(RHQM) and quarter resolution and low quality(RQQL) are compared with context with only mid bit-planes. The latter has three resolution versions: full resolution, half resolution, and quarter resolution.

Table 4.2: ANMRR of data state, context, data state mid and context mid. There are three compression versions: full resolution high quality(RFQH), half resolution medium quality(RHQM) and quarter resolution low quality(RQQL). Data state mid and context mid are data state and context with only mid bit-planes respectively.

ANMRR	RFQH	RHQM	RQQL
Data state	0.281	0.303	0.329
Context	0.275	0.297	0.329
	full resolution	half resolution	quarter resolution
Data state Mid	0.344	0.367	0.386
Context Mid	0.297	0.324	0.334

Table 4.3: ANMRR of data state, context, SCD, CLD, EHD. There are three compression versions: full resolution high quality(RFQH), half resolution medium quality(RHQM) and quarter resolution low quality(RQQL).

ANMRR	RFQH	RHQM	RQQL
Data state	0.281	0.303	0.329
Context	0.275	0.297	0.329
MPEG-7 SCD	0.216	0.253	0.870
MPEG-7 CLD	0.347	0.353	0.360
MPEG-7 EHD	0.338	0.388	0.642

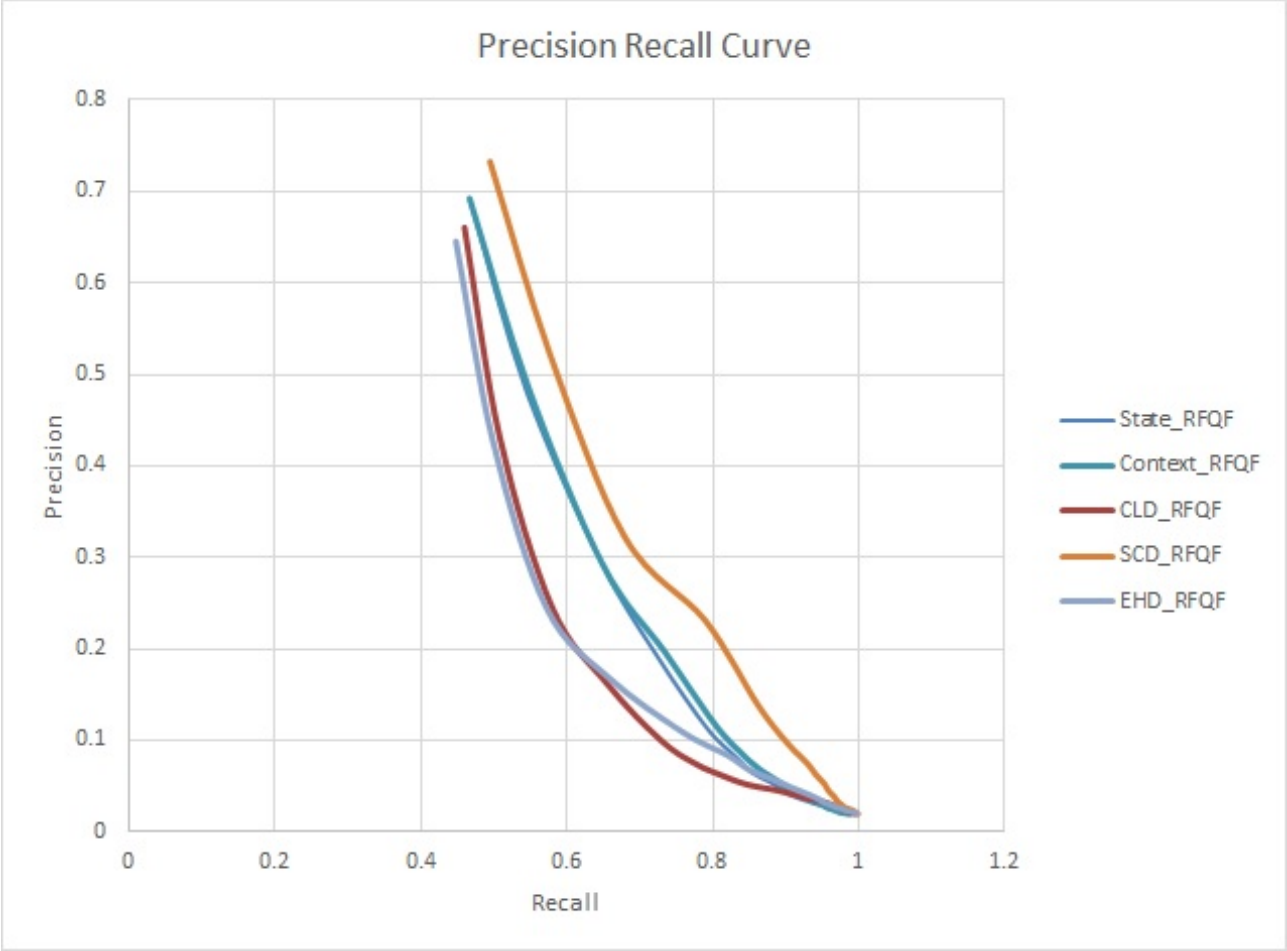

Figure 4.18: Precision-recall curve: proposed descriptors and MPEG-7 descriptors(CLD, SCD and EHD), full resolution high quality

Table 4.4: ANMRR of data state, context, SIFT. There are three compression versions: full resolution high quality(RFQH), half resolution medium quality(RHQM) and quarter resolution low quality(RQQL).

ANMRR	RFQH	RHQM	RQQL
Data state	0.281	0.303	0.329
Context	0.275	0.297	0.329
SIFT	0.394	0.436	0.490

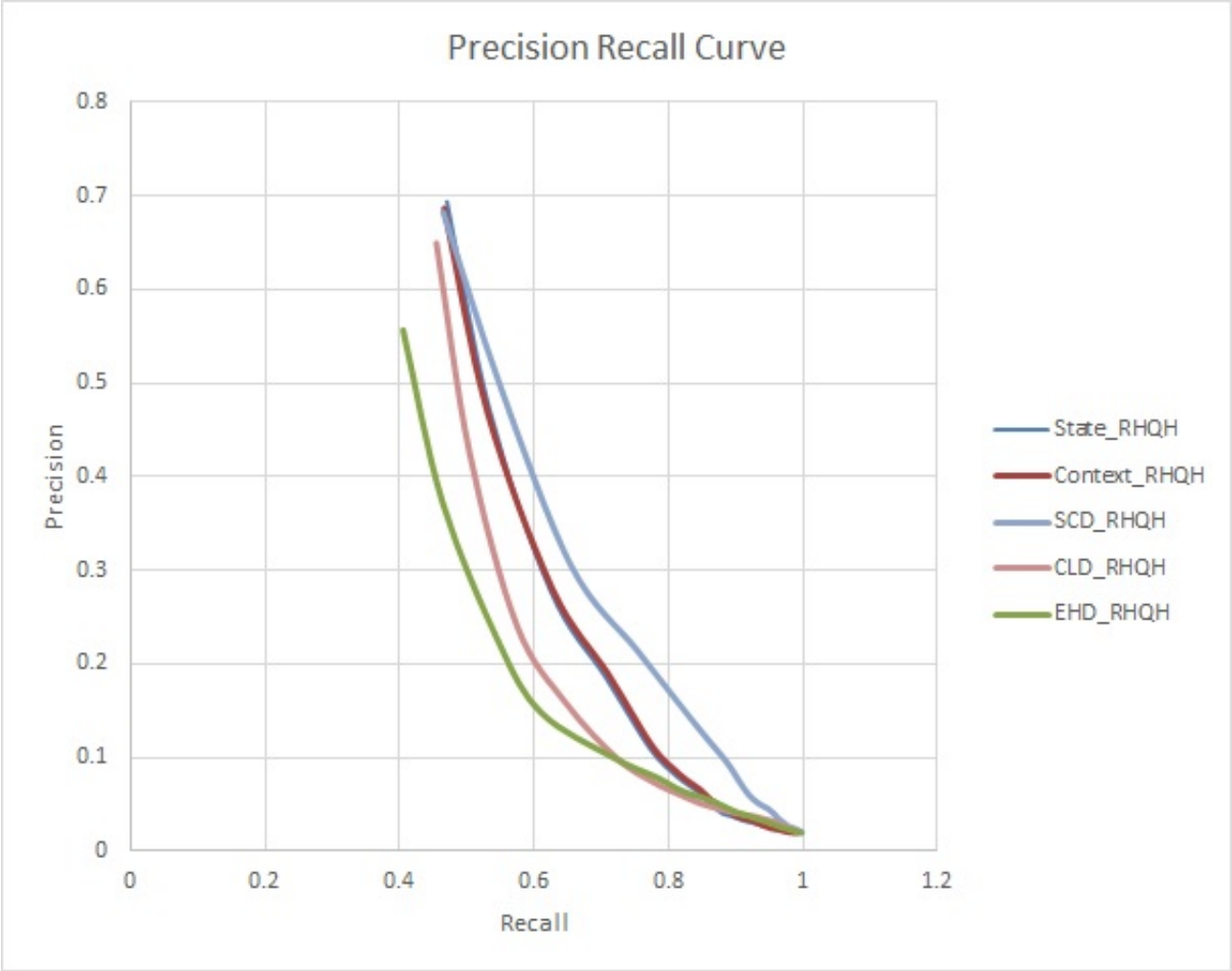

Figure 4.19: Precision-recall curve: proposed descriptors and MPEG-7 descriptors(CLD, SCD and EHD), half resolution medium quality

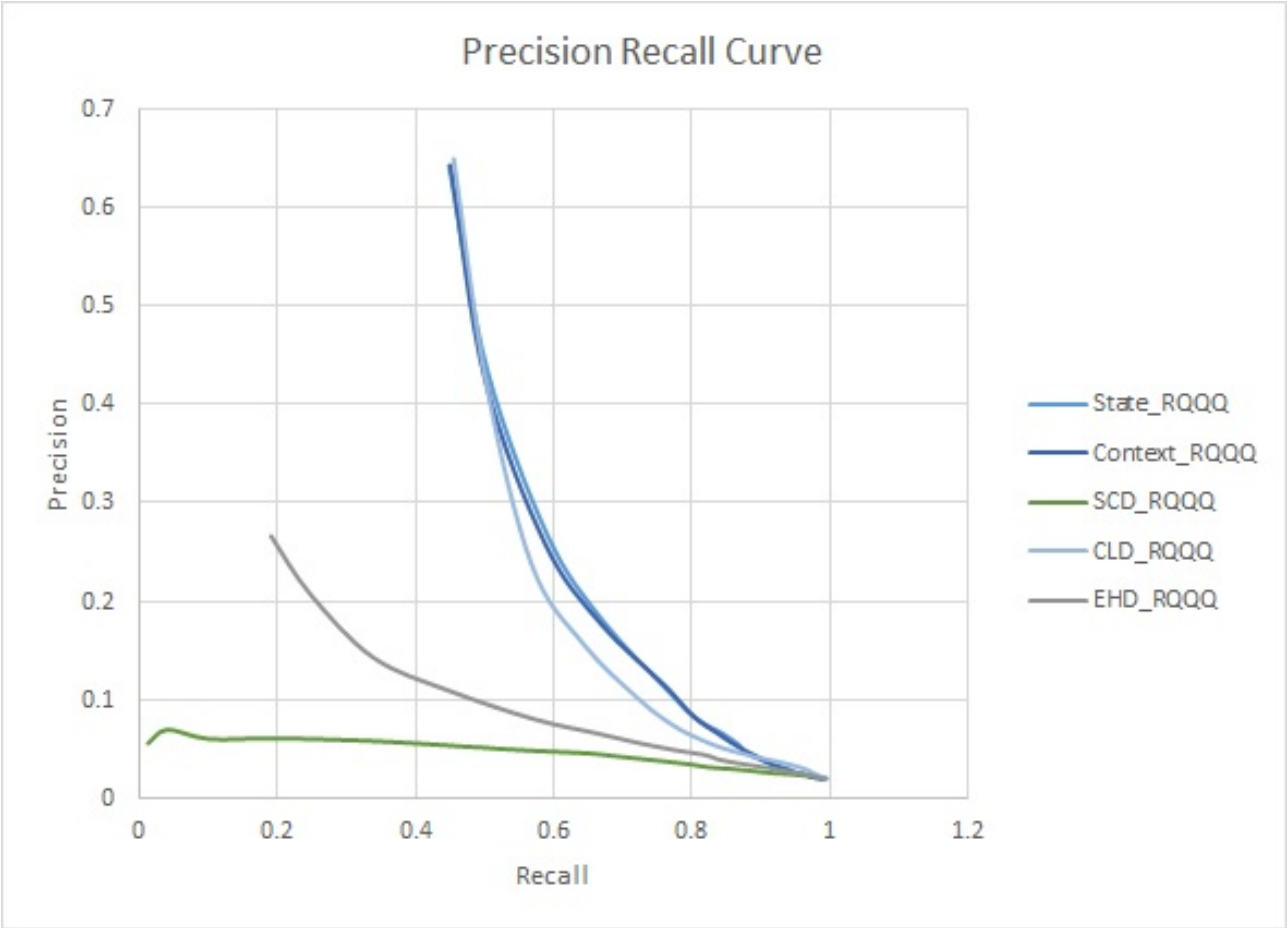

Figure 4.20: Precision-recall curve: proposed descriptors and MPEG-7 descriptors(CLD, SCD and EHD), quarter resolution low quality

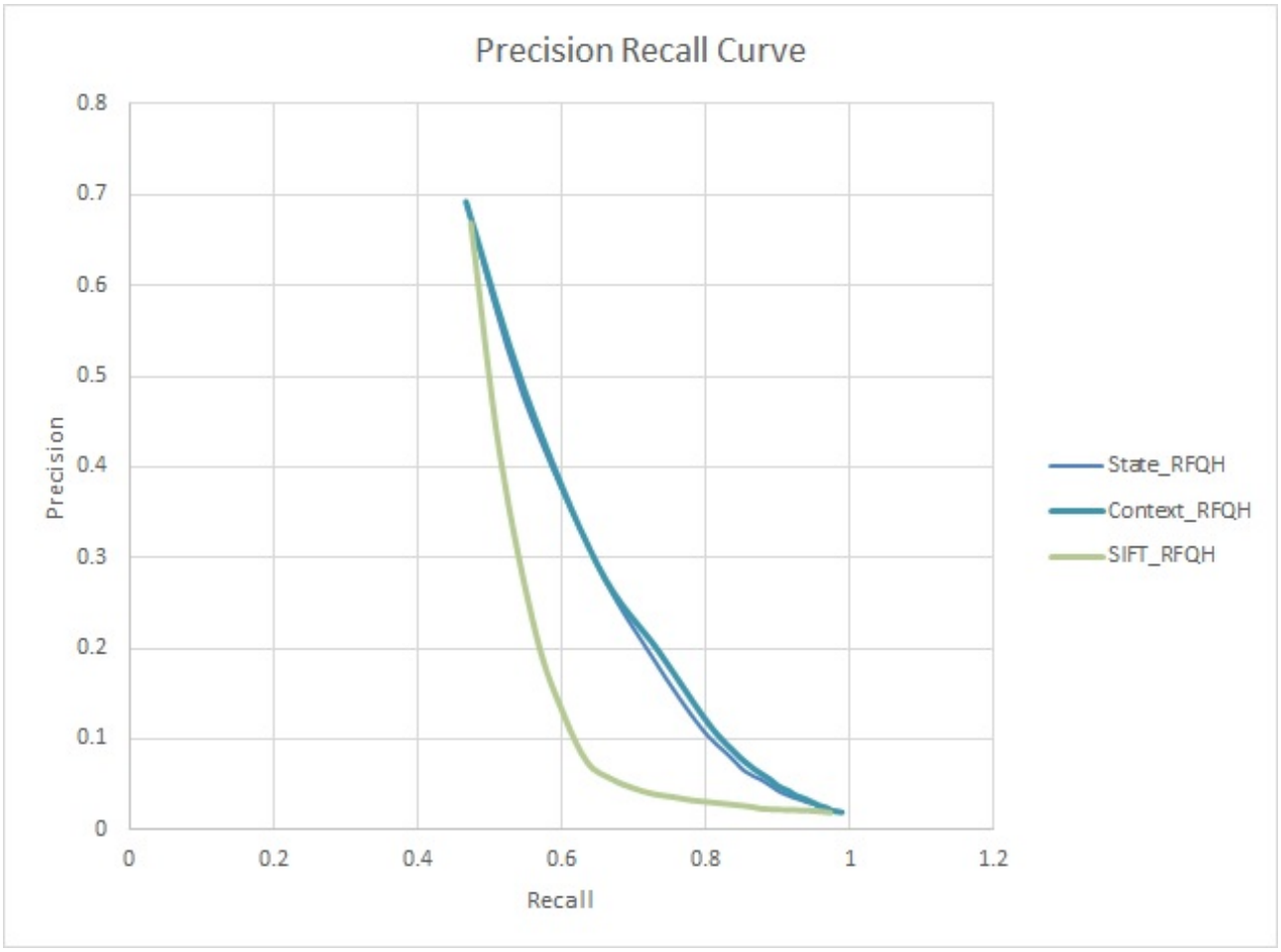

Figure 4.21: Precision-recall curve: proposed descriptors and SIFT, full resolution high quality

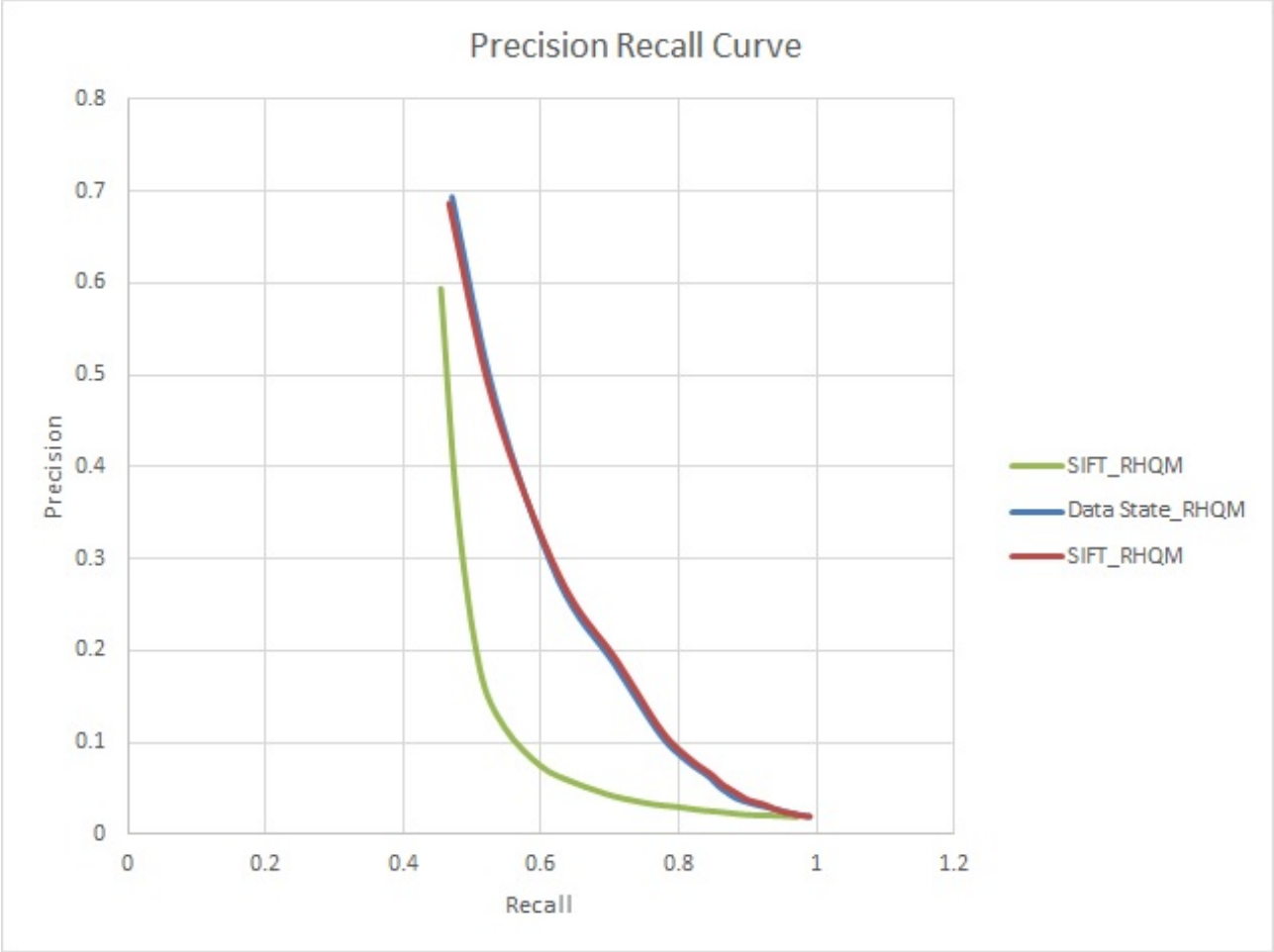

Figure 4.22: Precision-recall curve: proposed descriptors and SIFT, half resolution medium quality

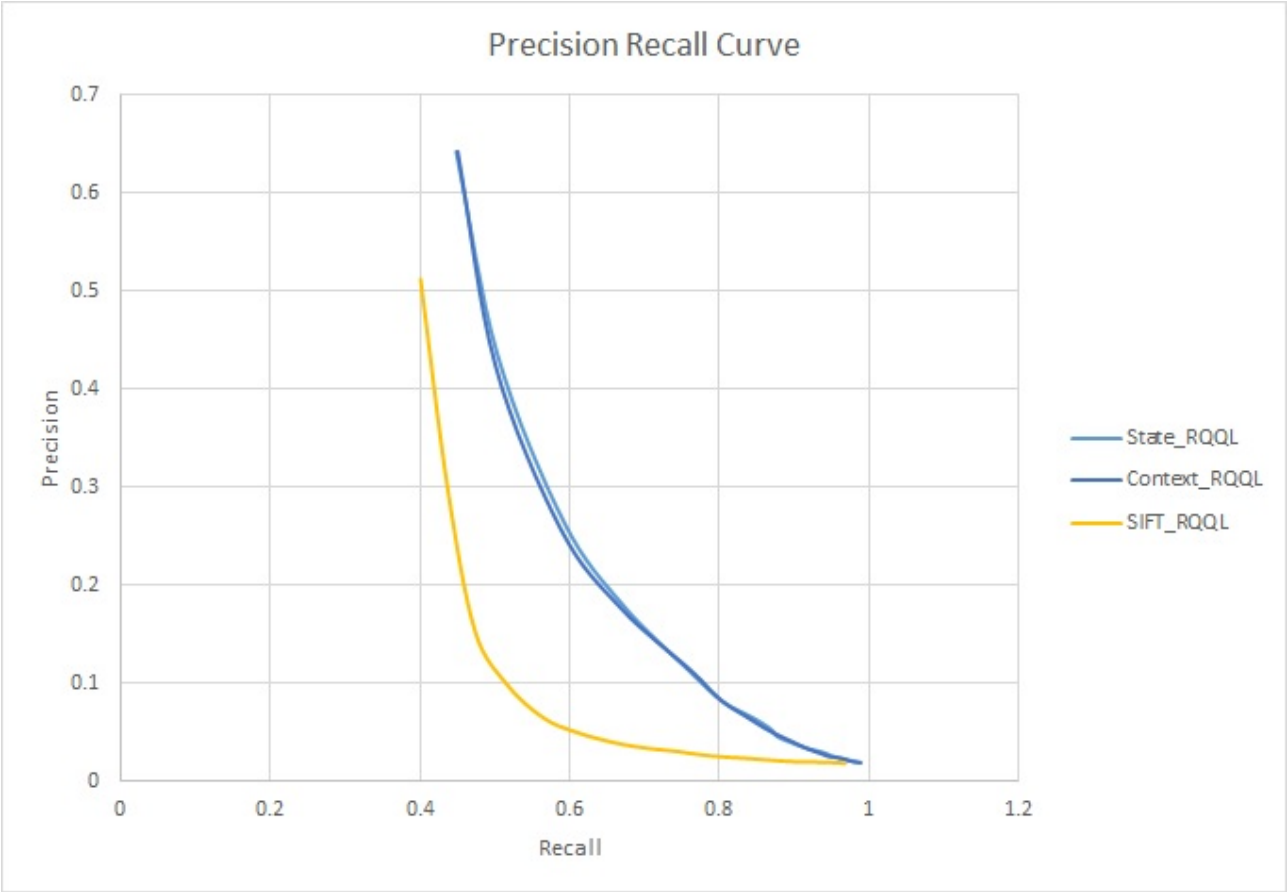

Figure 4.23: Precision-recall curve: proposed descriptors and SIFT, quarter resolution loq quality

Chapter 5

Conclusions and Future Work

In this thesis, research on compressed domain low level visual descriptors has been presented. The majority of low level visual descriptors have been reviewed, and image compression and image retrieval based on this domain have been investigated. Different descriptors have their own advantages, but they are not able to satisfy the need in this research. This research has developed new descriptors based on JPEG 2000 standard, and a retrieval system is built up to run experimental work. The performance has been evaluated and compared with state-of-art MEPG7 descriptors, the following remarks can be concluded from this research. After that, the future work that can be continued beyond this research is listed.

5.1 Conclusions

The retrieval results in chapter 4 shows that proposed descriptors, data state and context, have good performances. In terms of precision-recall curve and ANMRR, they outperform CLD and EHD when there is no compression or mildly compression; when images are highly compressed, their performances are the best.

When it comes to resolution and quality scalability, data state and context descriptors have excellent performances. They have been compared with state-of-art MPEG-7 descriptors, and their results are better than CLD and EHD in various resolution and quality situation.

Although SCD performs better in full resolution and quality and half resolution and quality situation, its performance drops significantly when resolution and quality are quarter. It can be concluded that both data state and context have a stable and better performance when resolution and quality decreases. The results of mid-level compression are still quite close to results of original images, which indicates higher level subbands and lower bit-planes contains most information.

It is the structure of bit stream and coding process of JPEG 2000 standard that make resolution and quality scalability possible. In this research, data state and context are collected and generated during coding process and feature vectors are extracted based on the structure of bit stream. In practical retrieval, each image in the database only has one version (like full resolution and high quality). No matter what resolution and quality a query image is, feature vectors in the database can achieve the same resolution and quality level by discarding unused subbands and bit-planes. If other descriptors like MPEG-7 descriptors want to achieve the same goal, images in the database first need to be compressed to get the same resolution and quality level as query image, and then feature vectors are extracted from each image to measure similarity, which costs considerable time and computation. It can be concluded that the proposed descriptors are time-saving and computation-saving in compression domain.

In chapter 4, it has been explained that only one colour channel, i.e., Y component, is used in descriptors generation and feature extraction. However, it has the best results in terms of scalability and performance compared with other MPEG-7 descriptors, which uses multiple colour channels. As Y component is actually a grayscale version of original image, there are still potential enhancements if more channels with colours are involved.

As wavelet transform is employed in JPEG 2000 standard, features extract from different subbands have directionality. Thus, the proposed data state and context are directional descriptors, which similar to some texture descriptors, such as EHD. In terms of retrieval results, thus, both data state and context are more suitable and robust than other directional texture descriptors in compressed domain.

The proposed descriptors outperform state of art SIFT method in high, medium and low resolution and quality levels in terms of precision recall curve and ANMRR. SIFT has advantages in situations like object detection [6, 29]. However, when it comes to scalable coding and image retrieval with adapted content, the performance and consistency is not competitive with proposed methods.

When there is a limitation of bandwidth, an alternative solution is to keep only mid bit-planes (3 or 4 bit-planes in the middle) and discard the rest ones. As there may be 8 or even more bit-planes of an image, the size of data can be reduced significantly in this way. There is a trade-off between the loss of retrieval performance and the size of images. Although the performance is, in this research, it is actually the most significant three bit-planes are kept in low quality situation, whose performance

5.2 Future Work

As only one colour channel Y has been used in this research, more channels can be engaged for feature extraction, which brings colour information in addition to grayscale component.

Current research has not employed any machine learning techniques. In the future, either supervised or unsupervised learning can be brought in with a huge database. Experiments with more situations can be implemented to further investigate the potential of proposed descriptors.

Two proposed descriptors, data state and context, have their own characteristics, a further study can be conducted on the difference and similarity between these two descriptors. A combination of them can be developed where each of them may contribute its advantages.

This research can be extended into video analysis. New descriptors with resolution and quality scalability can be developed for video domain.

Bibliography

[1] D. S. Taubman and M. W. Marcellin. *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[2] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262 – 282, 2007.

[3] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, May 2008.

[4] Dengsheng Zhang, Md. Monirul Islam, and Guojun Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346 – 362, 2012.

[5] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, and A Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715, Jun 2001.

[6] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[7] *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[8] Gerald Schaefer. Pixel domain and compressed domain image retrieval features. 2013.

[9] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG2000 still image compression standard. *IEEE Signal Processing Magazine*, pages 36–58, 2001.

[10] T. Sikora. The mpeg-7 visual standard for content description-an overview. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):696–702, Jun 2001.

[11] MichaelJ. Swain and DanaH. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[12] Anil K. Jain and Aditya Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233 – 1244, 1996.

[13] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(7):729–736, Jul 1995.

[14] John R. Smith and Shih fu Chang. Tools and techniques for color image retrieval. pages 426–437, 1996.

[15] Markus A. Stricker and Markus Orengo. Similarity of color images. *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 2420:381–392, 1995.

[16] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, number 9 in MULTIMEDIA '96, pages 65–73. ACM, 1996.

[17] G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. In *Applications of Computer Vision, 1996. WACV '96., Proceedings 3rd IEEE Workshop on*, pages 96–102, Dec 1996.

[18] Andrzej Materka and Michal Strzelecki. Texture analysis methods – a review. Technical report, INSTITUTE OF ELECTRONICS, TECHNICAL UNIVERSITY OF LODZ, 1998.

[19] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding

to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, June 1978.

[20] Soo Beom Park, Jae Won Lee, and Sang Kyoon Kim. Content-based image classification using a neural network. *Pattern Recognition Letters*, 25(3):287 – 300, 2004.

[21] George R. Cross and Anil K. Jain. Markov random field texture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5(1):25–39, Jan 1983.

[22] A. Speis and G. Healey. An analytical and experimental study of the performance of markov random fields applied to textured images using small samples. *Image Processing, IEEE Transactions on*, 5(3):447–458, Mar 1996.

[23] Nirupam Sarkar and B.B. Chaudhuri. An efficient approach to estimate fractal dimension of textural images. *Pattern Recognition*, 25(9):1035 – 1041, 1992.

[24] B.B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1):72–77, Jan 1995.

[25] Kuen-Long Lee and Ling-Hwei Chen. An efficient computation method for the texture browsing descriptor of mpeg-7. *Image and Vision Computing*, 23(5):479 – 489, 2005.

[26] J.Z. Wang, Jia Li, and G. Wiederhold. Simplicity: semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(9):947–963, Sep 2001.

[27] M. Unser. Texture classification and segmentation using wavelet frames. *Image Processing, IEEE Transactions on*, 4(11):1549–1560, Nov 1995.

[28] P. Wu, B.S. Manjunath, S. Newsam, and H.D. Shin. A texture descriptor for browsing and similarity retrieval. *Signal Processing: Image Communication*, 16(12):33 – 43, 2000.

[29] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[30] Jurandy Almeida, Ricardo da S Torres, and Siome Goldenstein. Sift applied to cbir. *Revista de Sistemas de Informacao da FSMA n*, 4:41–48, 2009.

[31] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[32] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1 – 19, 2004.

[33] R. Mehrotra and J.E. Gary. Similar-shape retrieval in shape data management. *Computer*, 28(9):57–62, Sep 1995.

[34] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962.

[35] M. Bober. Mpeg-7 visual shape descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):716–719, Jun 2001.

[36] Lei Zhang, Fuzong Lin, and Bo Zhang. Support vector machine learning for image retrieval. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 721–724 vol.2, Oct 2001.

[37] Rui Shi, Huamin Feng, Tat-Seng Chua, and Chin-Hui Lee. An adaptive image content representation and segmentation approach to automatic image annotation. In Peter Enser, Yiannis Kompatsiaris, NoelE. OConnor, AlanF. Smeaton, and ArnoldW.M. Smeulders, editors, *Image and Video Retrieval*, volume 3115 of *Lecture Notes in Computer Science*, pages 545–554. Springer Berlin Heidelberg, 2004.

[38] ChristopherJ.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[39] S.D. MacArthur, C.E. Brodley, and Chi-Ren Shyu. Relevance feedback decision trees in content-based image retrieval. In *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, pages 68–72, 2000.

[40] Ishwar K. Sethi, Ioana L. Coman, and Daniela Stan. Mining association rules between low-level image features and high-level concepts. volume 4384, pages 279–290, 2001.

[41] Wanjun Jin, Rui Shi, and Tat-Seng Chua. A semi-naïve bayesian method incorporating clustering with pair-wise constraints for auto image annotation. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 336–339, New York, NY, USA, 2004. ACM.

[42] N. Vasconcelos and Andrew Lippman. Library-based coding: a representation for efficient video compression and retrieval. In *Data Compression Conference, 1997. DCC '97. Proceedings*, pages 121–130, Mar 1997.

[43] A Vailaya, M.AT. Figueiredo, AK. Jain, and Hong-Jiang Zhang. Image classification for content-based indexing. *Image Processing, IEEE Transactions on*, 10(1):117–130, Jan 2001.

[44] Daniela Stan and Ishwar K. Sethi. Mapping low-level image features to semantic concepts. volume 4315, pages 172–179, 2001.

[45] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 11–, New York, NY, USA, 2004. ACM.

[46] *Hierarchical clustering algorithm for fast image retrieval*, volume 3656, 1998.

[47] Xia Wan and C.-C.J. Kuo. A new approach to image retrieval with hierarchical color clustering. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):628–643, Sep 1998.

[48] Subhasis Saha. Image compression - from dct to wavelets : A review, 2000.

[49] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, 41(12):3445–3462, Dec 1993.

[50] A Said and W.A Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(3):243–250, Jun 1996.

[51] D. Taubman. High performance scalable image compression with ebcot. *Image Processing, IEEE Transactions on*, 9(7):1158–1170, Jul 2000.

[52] M.K Mandal, F Idris, and S Panchanathan. A critical evaluation of image and video indexing techniques in the compressed domain. *Image and Vision Computing*, 17(7):513 – 529, 1999.

[53] Gerald Schaefer. Content-based retrieval of compressed images. In *DATESO*, pages 175–185. Citeseer, 2010.

[54] D. Edmundson and G. Schaefer. An overview and evaluation of jpeg compressed domain retrieval techniques. In *ELMAR, 2012 Proceedings*, pages 75–78, Sept 2012.

[55] M. Shneier and M. Abdel-Mottaleb. Exploiting the jpeg compression scheme for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):849–853, Aug 1996.

[56] J.A. Lay and Ling Guan. Image retrieval based on energy histograms of the low frequency dct coefficients. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3009–3012 vol.6, Mar 1999.

[57] Gerald Schaefer. Jpeg image retrieval by simple operators, 2001.

[58] J. Jiang, A. Armstrong, and G.C. Feng. Direct content access and extraction from {JPEG} compressed images. *Pattern Recognition*, 35(11):2511 – 2519, 2002.

[59] Guocan Feng and Jianmin Jiang. {JPEG} compressed image retrieval via statistical features. *Pattern Recognition*, 36(4):977 – 985, 2003.

[60] M. Eom and Y. Choe. Fast extraction of edge histogram in dct domain based on mpeg7.

[61] Zhe-Ming Lu, Su-Zhi Li, and Hans Burkhardt. A content-based image retrieval scheme in jpeg compressed domain. *International Journal of Innovative Computing, Information and Control*, 2(4):831–839, 2006.

[62] T. Chang and C.-C.J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *Image Processing, IEEE Transactions on*, 2(4):429–441, Oct 1993.

[63] M.K. Mandal, T. Aboulnasr, and S. Panchanathan. Image indexing using moments and wavelets. *Consumer Electronics, IEEE Transactions on*, 42(3):557–565, Aug 1996.

[64] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 277–286, New York, NY, USA, 1995. ACM.

[65] James Ze Wang, Gio Wiederhold, O. Firschein, and Sha Xin Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *Digital Libraries, 1997. ADL '97. Proceedings., IEEE International Forum on Research and Technology Advances in*, pages 13–24, May 1997.

[66] Kai-Chieh Liang and C.-C.J. Kuo. Waveguide: a joint wavelet-based image representation and description system. *Image Processing, IEEE Transactions on*, 8(11):1619–1629, Nov 1999.

[67] Lin Ni. A novel image retrieval scheme in jpeg2000 compressed domain based on tree distance. In *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, volume 3, pages 1591–1594 vol.3, Dec 2003.

[68] M. Lamard, G. Cazuguel, G. Quellec, L. Bekri, C. Roux, and B. Cochener. Content based image retrieval based on wavelet transform coefficients distribution. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4532–4535, Aug 2007.

[69] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux. Wavelet optimization for content-based image retrieval in medical databases. *Medical Image Analysis*, 14(2):227 – 241, 2010.

[70] M. Lamard, W. Daccache, G. Cazuguel, C. Roux, and B. Cochener. Use of a jpeg-2000 wavelet compression scheme for content-based ophtalmologic retinal images retrieval. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 4010–4013, Jan 2005.

[71] C. Abhayaratne. Content adaptation resilient low-level visual features for content based image retrieval. *IET Conference Proceedings*, pages 164–169(5), January 2008.

[72] Charith Abhayaratne and Farooq Muzammil. Robust visual content representation using compression modes driven low-level visual descriptors. In Benoit Huet, Alan Smeaton, Ketan Mayer-Patel, and Yannis Avrithis, editors, *Advances in Multimedia Modeling*, volume 5371 of *Lecture Notes in Computer Science*, pages 322–332. Springer Berlin Heidelberg, 2009.

[73] C. Christopoulos, A Skodras, and T. Ebrahimi. The jpeg2000 still image coding system: an overview. *Consumer Electronics, IEEE Transactions on*, 46(4):1103–1127, Nov 2000.

[74] D.S. Taubman and M.W. Marcellin. Jpeg2000: standard for interactive imaging. *Proceedings of the IEEE*, 90(8):1336–1357, Aug 2002.

[75] Robert Buckley. Jpeg 2000 as a preservation and access format for the wellcome trust digital library. 2009.

[76] T. Saidani, M. Atri, and R. Tourki. Implementation of jpeg 2000 mq-coder. In *Design and Technology of Integrated Systems in Nanoscale Era, 2008. DTIS 2008. 3rd International Conference on*, pages 1–4, March 2008.

[77] D. S. Taubman. High performance scalable image compression with ebcot. *IEEE Trans. on Image Processing*, 9:1158–1170, 2000.

[78] Qihui Wang and Changsheng Yang. Sign coding and estimation of zero-quantized coefficients in the reformed ezw algorithm. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 3010–3015. IEEE, 2004.

[79] University of Washington. Object and concept recognition for content-based image retrieval: Ground truth database.

[80] R. J. Qian, P. J. L. Van Beek, and M. I. Sezan. Image retrieval using blob histograms. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 125 –128 vol.1, 2000.

[81] Mathias Lux.

Chapter 6

Appendix

The original image, Barbara, is presented in Figure 6.1. As it is mentioned in chapter 3, the original size versions of data states and contexts of each bit-plane of wavelet coefficients are presented as following:

The original size versions of data States of each bit-plane are presented from Figure 6.2 to Figure 6.9. The original size versions of contexts of each bit-plane are presented from Figure 6.10 to Figure 6.17.

Figure 6.1: The original image, Barbara.

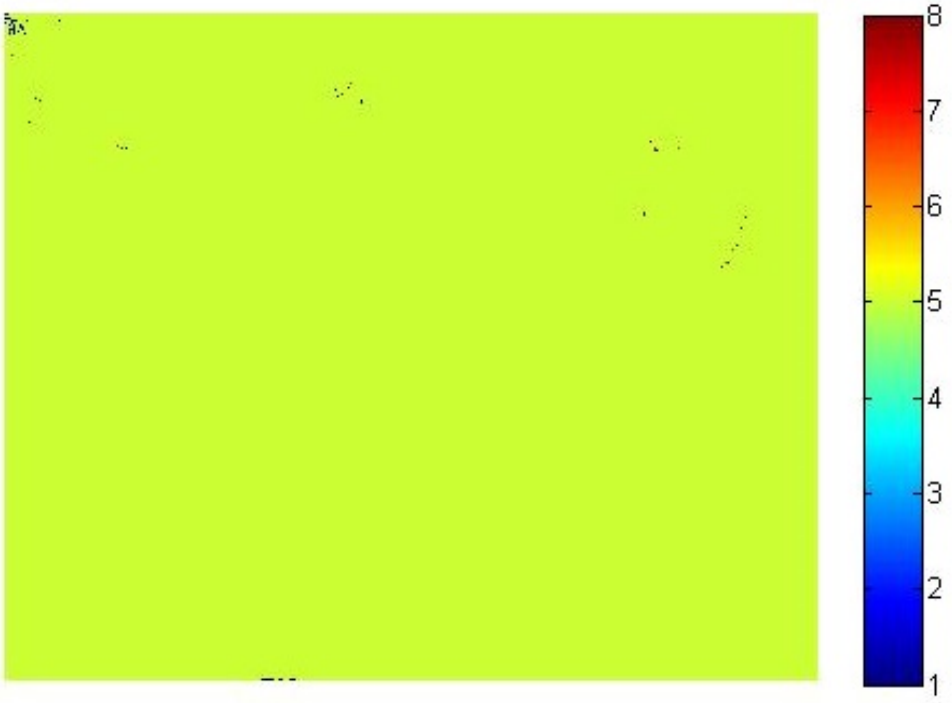

Figure 6.2: Data States: Most significant bit-plane

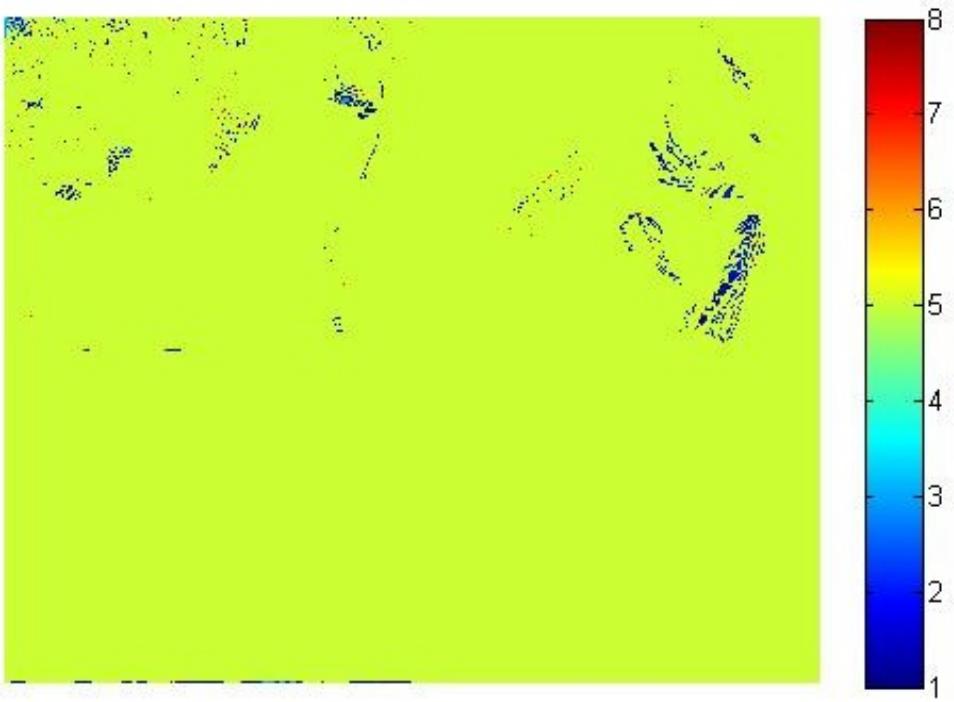

Figure 6.3: Data States: Bit-plane 6

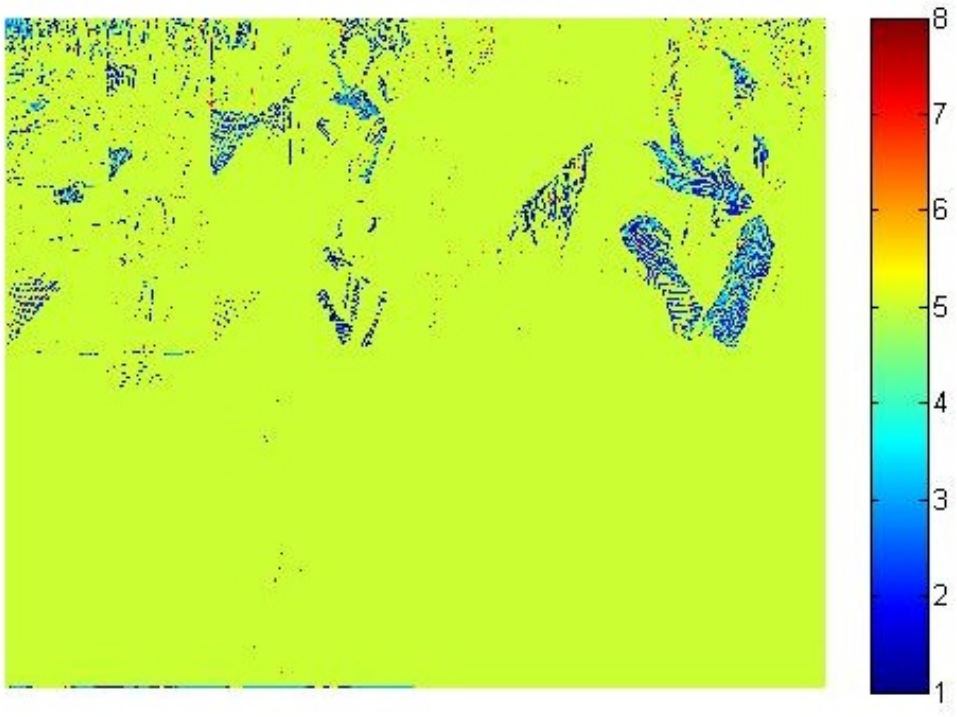

Figure 6.4: Data States: Bit-plane 5

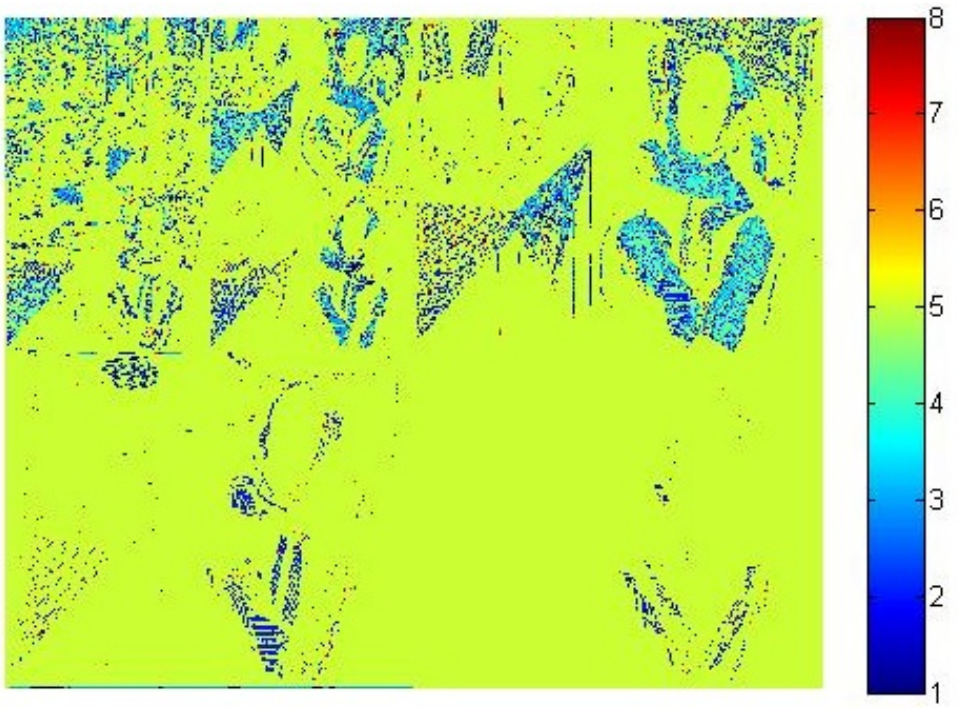

Figure 6.5: Data States: Bit-plane 4

Figure 6.6: Data States: Bit-plane 3

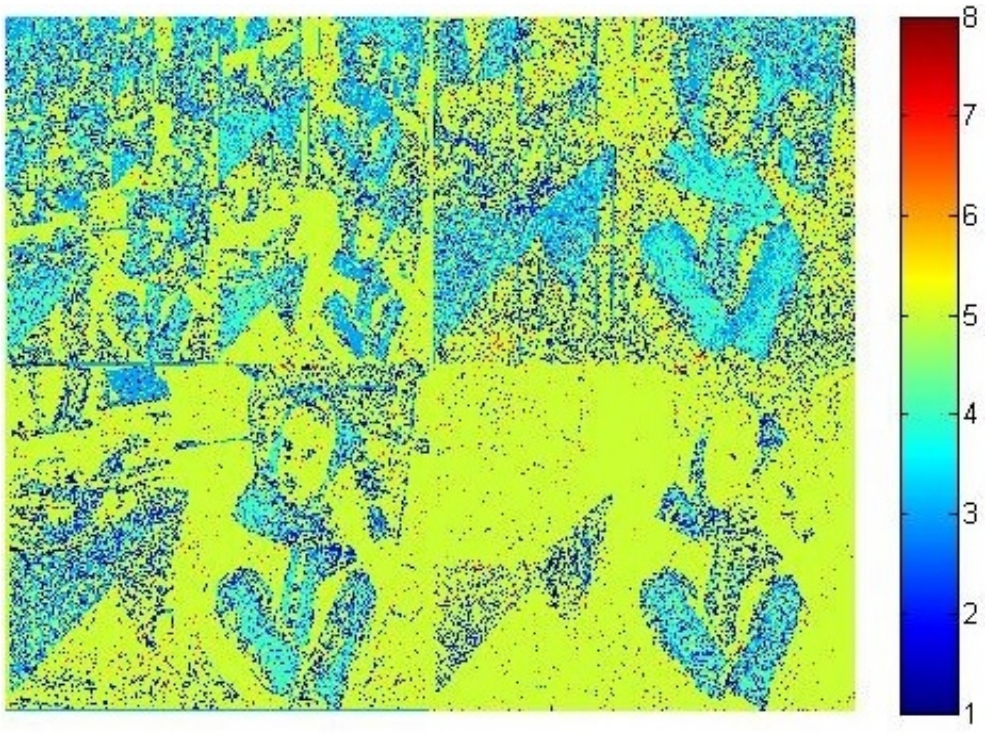

Figure 6.7: Data States: Bit-plane 2

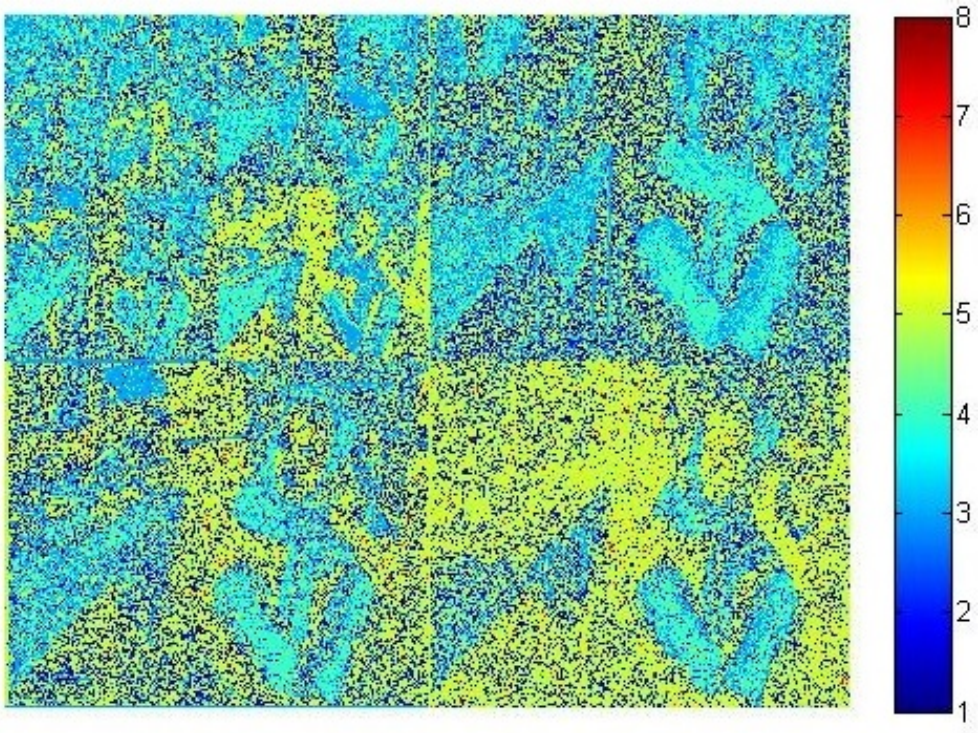

Figure 6.8: Data States: Bit-plane 1

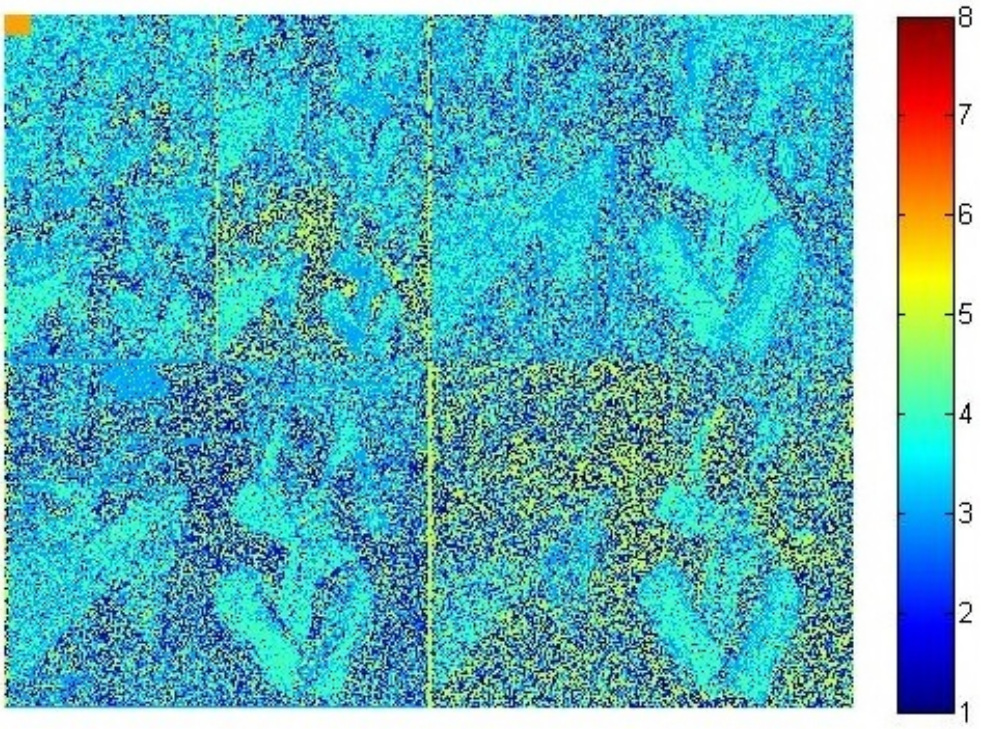

Figure 6.9: Data States: Least Significant bit-plane

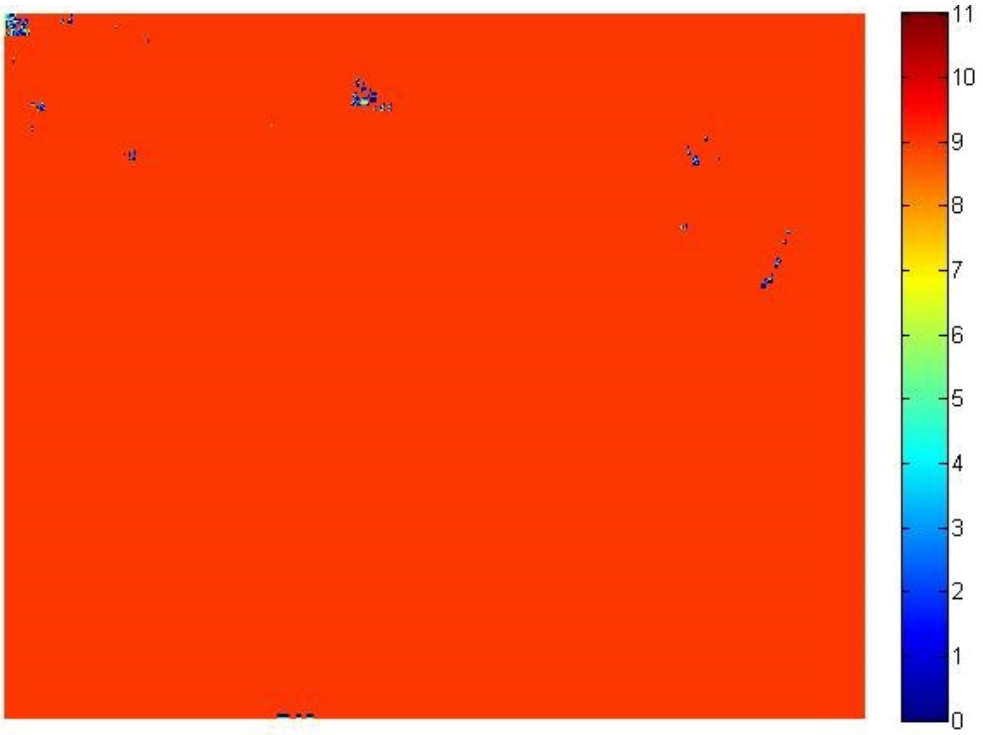

Figure 6.10: Contexts: Most significant bit-plane

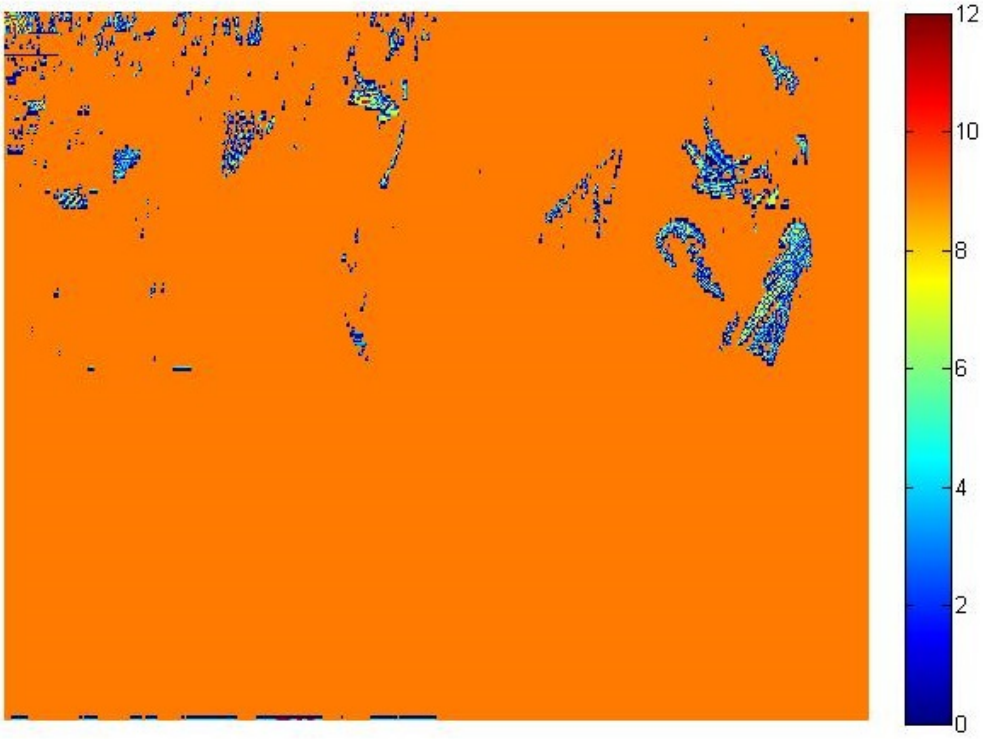

Figure 6.11: Contexts: Bit-plane 6

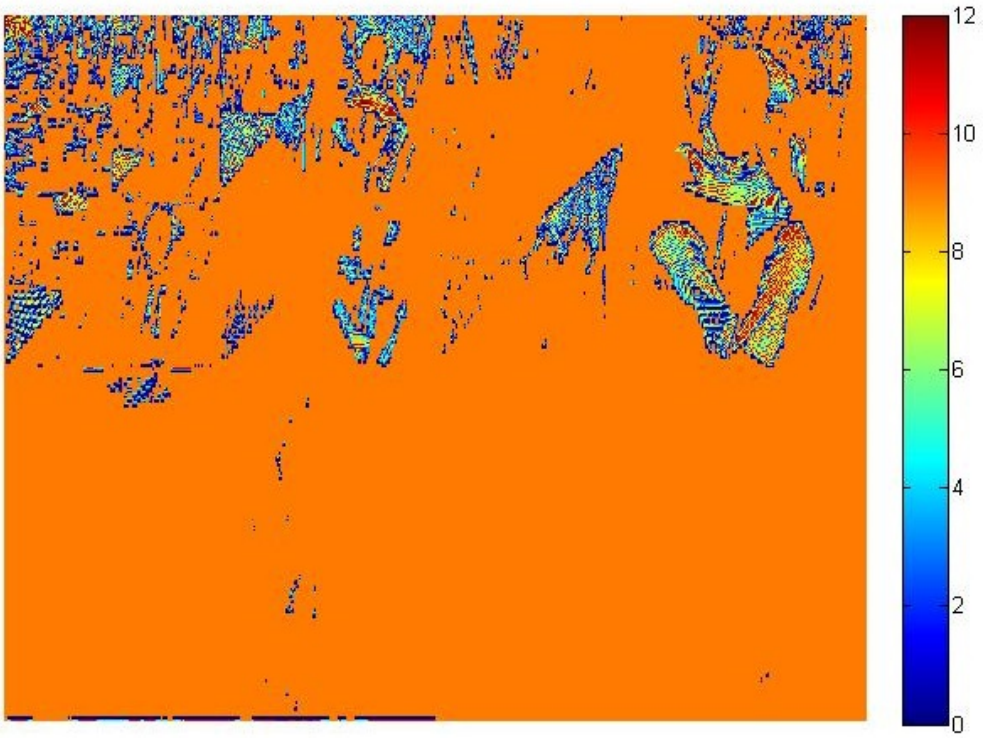

Figure 6.12: Contexts: Bit-plane 5

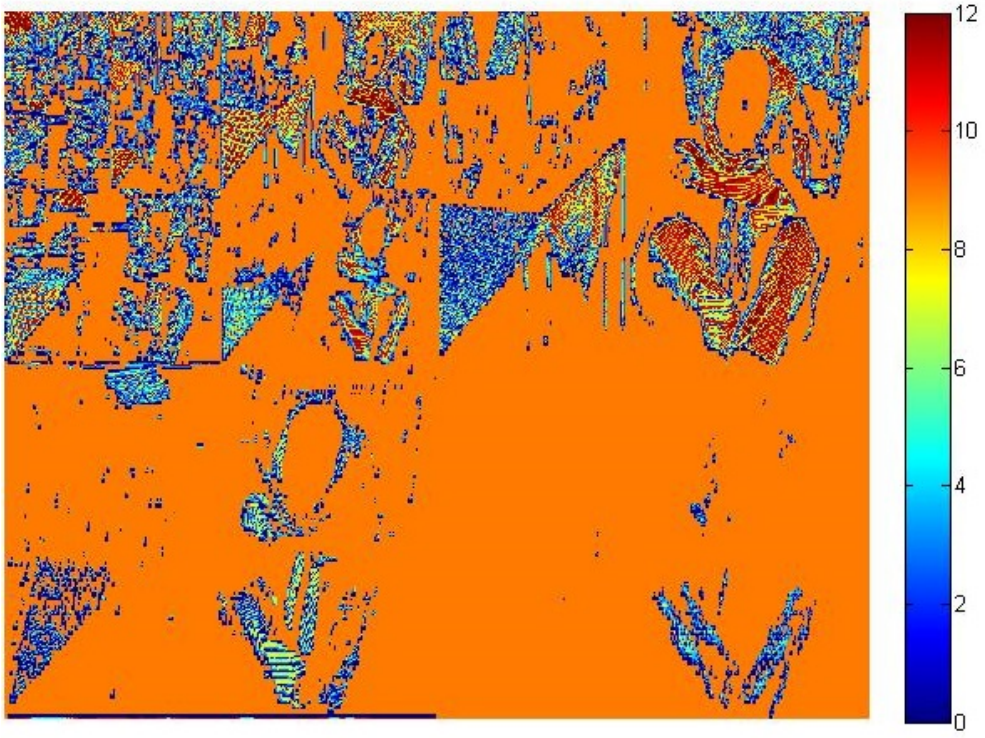

Figure 6.13: Contexts: Bit-plane 4

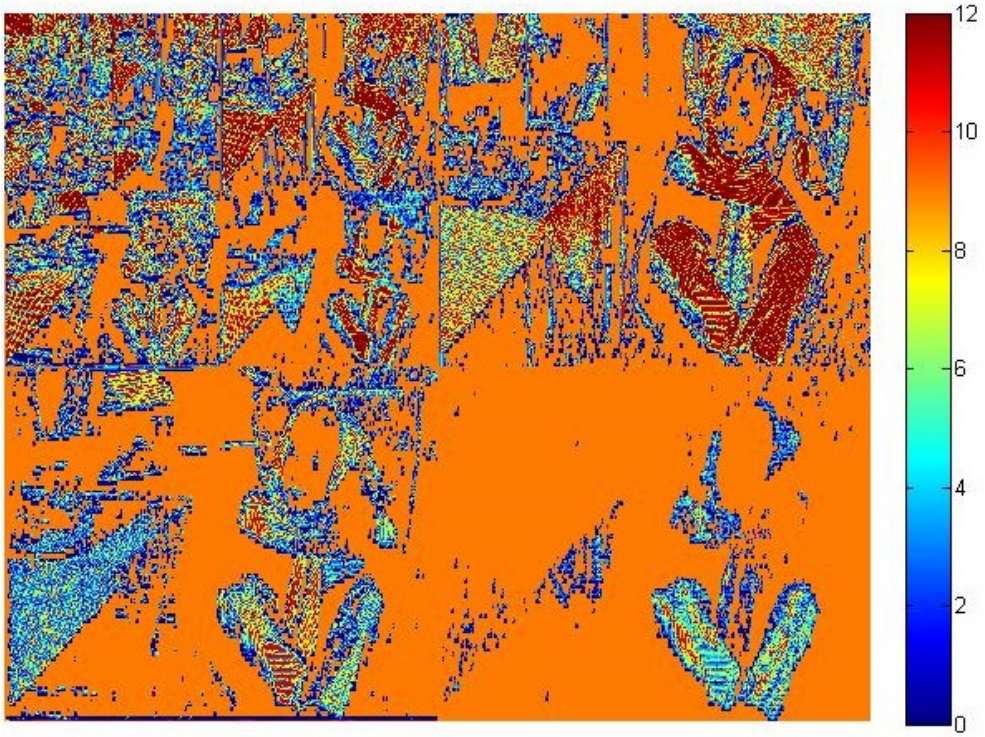

Figure 6.14: Contexts: Bit-plane 3

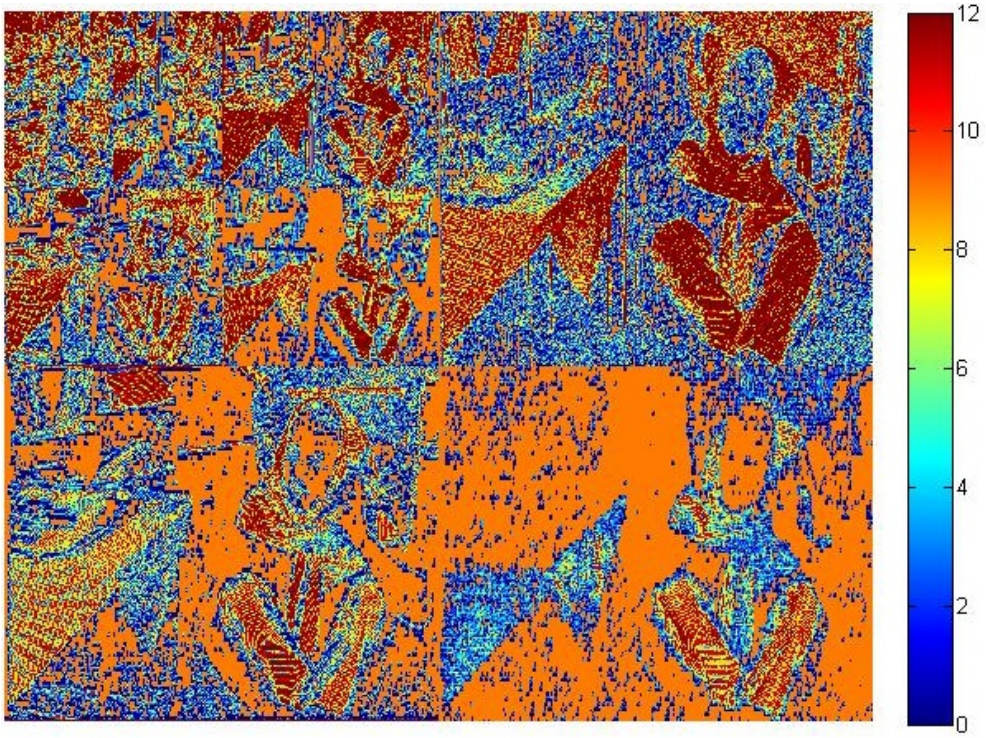

Figure 6.15: Contexts: Bit-plane 2

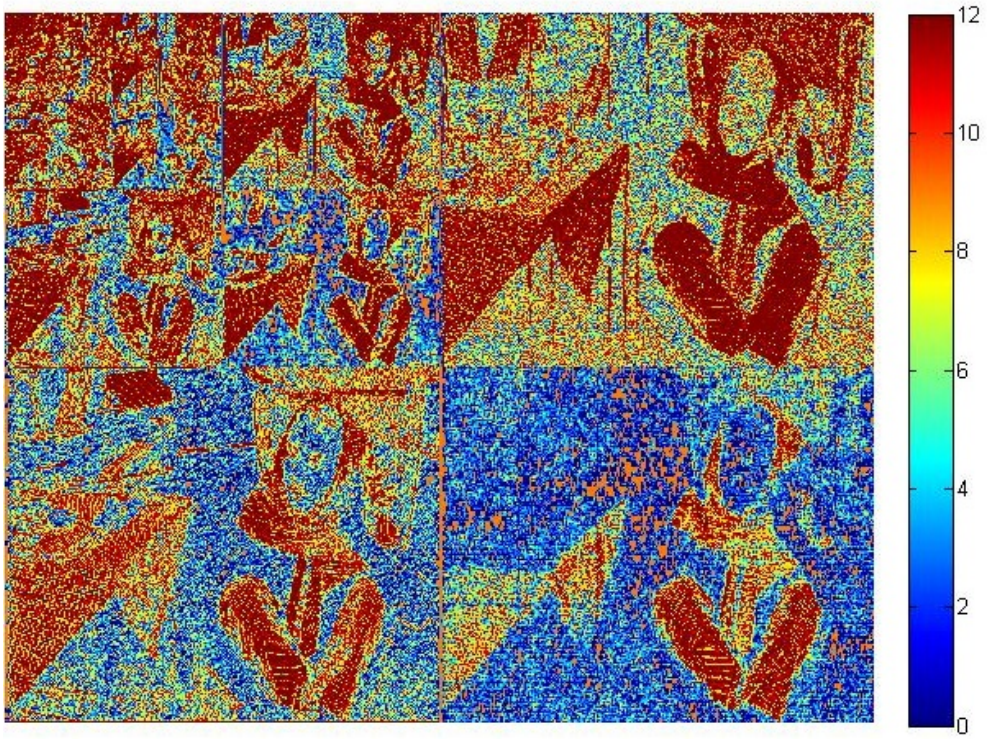

Figure 6.16: Contexts: Bit-plane 1

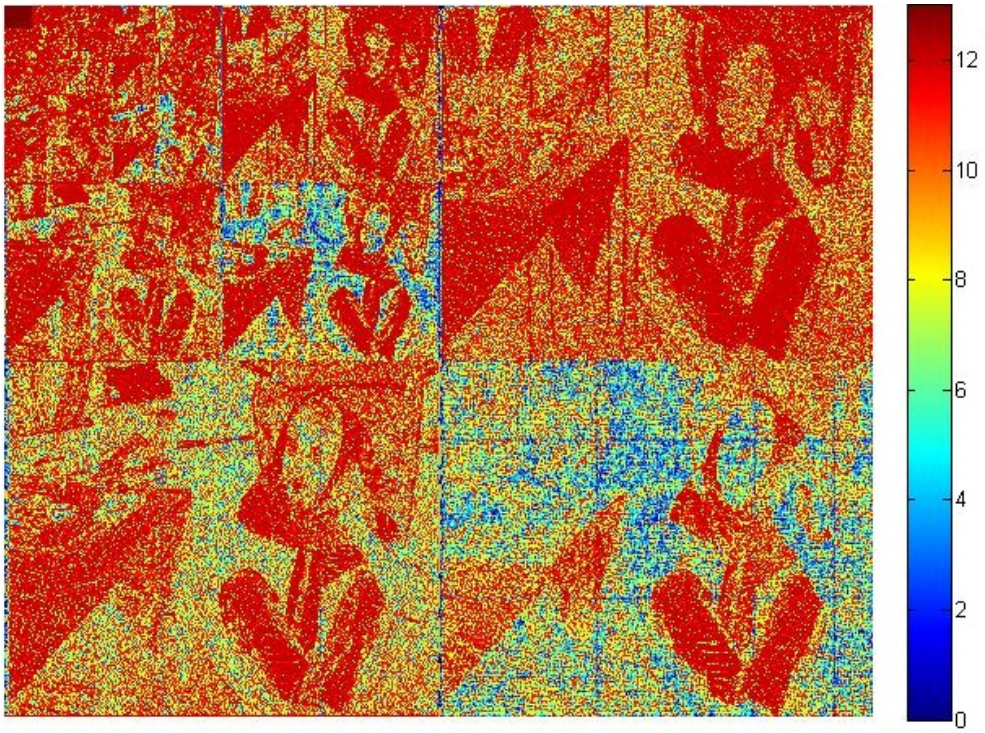

Figure 6.17: Contexts: Least Significant bit-plane