CHAPTER 6

OBJECT BASED LOSSLESS COMPRESSION OF IMAGES

6.1 GENERAL

Traditional image compression methods like the JPEG and JPEG 2000 are pixel based, and do not consider the content of the images while encoding them. The region of interest coders and object based coders perform image compression based on the image contents. However, they obtain a lesser bit rate than a direct coder when used for lossless image compression. This Chapter introduces an object based lossless coder (OBLC) with an improved bit rate.

6.2 OBJECT BASED CODING

Object based coding is an image compression scheme that uses a segmentation approach to separate the object from the background, and encode the object and the background regions at different bit rates. The objective is to improve the overall compression ratio. Object based coding is useful for tele-medicine applications, which have bandwidth and memory constraints.

Memory is an important design constraint while compressing the images. Any algorithm which expands the image during the compression process requires more memory at the encoder end. The Contourlet transform is an example of an expansive algorithm. One of the memory efficient,
transform based encoding techniques is the DCT, which processes the image as 8x8 blocks, and uses a small buffer to hold the image data. On the other hand, the wavelet transform is computed for the whole image, and the memory requirement is also high. Techniques to reduce the memory required for the wavelet domain compression scheme, is an area which requires more attention. In this context, object based coding can be considered as a method for reducing the memory requirements in wavelet based image coding.

6.2.1 Basic Concepts

Object based coding is more suitable for medical images due to their structure. Most of the medical images contain a large object region with a background. The background region is generally not important for the diagnosis. Hence, the general theme is to preserve the diagnostically important regions without loss, and apply lossy compression to the other regions. The coefficients associated with the object region can be transmitted before the coefficients of the background region. Therefore, it is necessary to identify the object coefficients in advance. This identification is generally performed, with the help of an object mask or region of interest (ROI) mask. The mask is a map which points to the coefficients that belong to the ROI.

Let I be the image in the wavelet domain and P \( \subseteq \) I be the ROI. Then, the function \( P(x) \) is defined as (Doukas & Maglogiannis 2007)

\[
P(x) = \begin{cases} 
1 & \text{if } x \in P \\
0 & , \text{otherwise}
\end{cases}
\]  
(6.1)

The object region in an image is generally random shaped. In some of the object based approaches, a rectangular region surrounding the object is identified, thus giving a shape to the random object. The straightforward approach to code the rectangular shaped object is to first find the bounding
box enclosing the object, and code the pixels inside and outside the box. For encoding an arbitrary shaped object, shape extraction and shape encoding methodologies are required.

6.2.2 Shape Adaptive Discrete Wavelet Transform

The Shape Adaptive Discrete Wavelet Transform (SA-DWT) is an improvisation of the SA-DCT coding technique. In the SA-DWT scheme, the object region is first segmented from the background. It is essentially a one dimensional DWT that is applied to each row of the ROI. The rows in the ROI are of random length. Hence, before encoding the ROI, each row in the ROI region is extended to the initial row length of the image. The ROI region is encoded without loss, while the background is quantised and encoded. Since the object is of an arbitrary shape, the shape information is also required by the decoder for reconstruction. Chain coding is used to send the shape information of the object.

One feature of the SA-DWT is that, the number of coefficients after the application of the transform is equal to the number of pixels in the object region. This is an advantage in comparison with the JPEG2000 ROI coding, which generates more coefficients than the number of pixels in the original image. Due to its high computational complexity, the SA-DWT approach is not preferred for object based coding.

6.2.3 The MAXSHIFT Method

The MAXSHIFT based ROI coding can be treated as a variation of the object coding approach. This method has been included as a part of the JPEG2000 standard for the ROI based coding facility. The ROI coding in the JPEG2000 uses three mechanisms, namely, tiling, code block selection and coefficient scaling.
Tiling is the process of dividing the original image into rectangular sub-images so that, each tile can be coded separately. The tiling operation is performed mainly for reducing the memory requirements. The tiling process also helps in accessing the ROI areas in the tile and codes them at different bit rates. However, the compression ratio decreases with a decrease in the tile size.

The tiling process is followed by the code block selection. After applying the DWT to the sub-images, the wavelet coefficients that belong to the ROI are determined, by using a ROI mask. The mask is a map which has a non-zero value in the ROI domain, and a zero value in the non-ROI region. The wavelet coefficients belonging to the ROI are identified by mapping the transform coefficients to the spatial domain samples. Let \( x(2n) \) and \( x(2n+1) \) be two spatial domain samples that belong to the ROI (Tahoces et al 2008). For a 5/3 wavelet transform, these samples are mapped using Equations (6.2) and (6.3).

\[
X(2n) = L(n) - \frac{H(n-1) + H(n)}{4} \tag{6.2}
\]

\[
X(2n+1) = \frac{L(n) + L(n+1)}{2} + \frac{H(n-1) + 6H(n) - H(n+1)}{8} \tag{6.3}
\]

Therefore, the wavelet sub band coefficients required to reconstruct the original samples \( x(2n) \) and \( x(2n+1) \) are \( H(n) \), \( H(n-1) \), \( H(n+1) \), \( L(n) \) and \( L(n+1) \).

After identifying the ROI coefficients, a scaling value, \( S \) is found. The scaling value is the magnitude of the largest wavelet coefficient in the background region. All the background coefficients (\( W_{BG} \)) are now scaled
down by dividing them with \((S + \Delta)\), where \(\Delta\) is a small constant. The effect of scaling is that, all the background coefficients will have a magnitude of less than one. Hence, no extra shape information is required. The scaling value \(S\) is also written into the bit stream. At the decoder end, all the background coefficients will be scaled up by using the \((S + \Delta)\) value. After the scaling process, the most significant ROI bit planes are placed in the bit stream before the background bit planes. This process ensures that the ROI is decoded before the background. The scaling process is shown in Figure 6.1.

![Diagram of scaling process](image)

(a) Before scaling

(b) After scaling

**Figure 6.1** Scaling of the background in the MAXSHIFT method
However, the MAXSHIFT based encoding reduces the compression ratio. This is due to the multiple coding of the boundary of the object region. The boundary is coded first during the ROI coding with a zero background and again during the background coding with a zero ROI. For multiple ROIs, the coding overhead further increases. If both the ROI and the background are coded without loss, then there is an increase of 5% in the bit rate.

In addition, with the MAXSHIFT algorithm, there is a possibility of over-coding of the ROI, as the ROI is coded till the LSB, before encoding the background coefficients. It is also observed that, in a JPEG2000 coder, the compression ratio achieved with direct lossless coding is higher than the value obtained using the ROI based lossless coder. This coding overhead problem of the ROI based lossless coder is addressed in this Chapter.

6.3 THE OBLC CODER

Some of the issues that need to be addressed by an object based coder are the detection of the object, the processing of the non-object region and the complexity of the encoding algorithm. The MAXSHIFT algorithm is not efficient for the lossless coding of the object and the background region. In addition, the ROI coefficients are identified through a complex mapping process between the pixels and the sub band coefficients.

In this work, the mapping process is eliminated by identifying the ROI as an object region in the spatial domain itself. The DPCM technique is chosen for encoding the object region, owing to its simplicity and fast compression and decompression capability. The DPCM method does not have the multi-resolution feature of the wavelet transform, which is generally used for progressive transmission. The absence of this feature is not of serious concern, as object based coding is based on a two step process. The non-
object area is encoded using the same DPCM method, after a split and merge operation. Figure 6.2 shows the block diagram representation of the encoder.

![Block Diagram](image)

**Figure 6.2 The OBLC Encoder**

### 6.3.1 Edge Detection

The object detection and extraction process begins with an edge detection operation. Since the edges characterize the boundaries of the object, they can be used to segment the object region. The most commonly used edge detection techniques are based on either the gradient method, which uses the first derivative, or the Laplacian method, which uses the second derivative. The usage of the second derivative makes the Laplacian method very sensitive to noise, and is not further considered. Among the gradient based operators, the Sobel operator has the advantage of a smoothing effect on a noisy image. Hence, the Sobel edge detector is chosen for detecting the edge information.

Let \( x(m,n) \) be an input pixel for the edge detection algorithm. If the pixel value exceeds a pre-fixed threshold value \( k \), it is considered as an edge point. All such edge points together make an edge map \( d(m,n) \), which is defined as (Gonzalez & Woods 2002),
\[
    d(m,n) = \begin{cases} 
    1 & \text{if } x(m,n) > k \\
    0 & \text{otherwise}
    \end{cases} \quad (6.4)
\]

The edge map can be used to trace the boundary of the object. The edge detected image obtained from a sample head image is shown in Figure 6.3.

(a) Original image  (b) After edge detection

Figure 6.3 Edge Detection

### 6.3.2 Object Extraction

If the objects are random in shape, then it is necessary to send the shape information along with the encoded image coefficients, which increases the bit rate. However, if a rectangular object region is considered, the coordinates of the corner pixels alone are required, thereby saving the bits spent on the shape information. The object extraction process begins with the formation of the rectangular bounding box.

In order to find the bounding box for the object region, the positions of the farthest rows and columns in the edge image are required. The algorithm for finding the co-ordinates of the bounding box is given below:-
1. Read the edge detected image

2. Scan the image row wise

3. Find all the rows and columns with an edge.

4. Arrange the row numbers in the ascending order. Find the first row number and the last row number

5. Arrange the column numbers in the ascending order. Find the first column number and the last column number

6. Specify the co-ordinates of the bounding box as the first row number, first column number, last row number and the last column number

The rows and columns intersect to form the bounding box surrounding the object. The coordinates of the rectangular bounding box are used to extract the object region. This approach makes the object identification an automatic process; there is no need to generate the complex ROI mask, as in the MAXSHIFT algorithm. The object region extracted using the bounding box co-ordinates for the brain image, is shown in Figure 6.4. Once the object region is extracted, it is encoded using the block based DPCM.

Figure 6.4 The object region of the brain image
6.3.3 Background Modelling

After identifying and encoding the object region, the background, also has to be encoded without loss. In order to optimise the encoding performance, the background region is pre-processed, by splitting it into four regions A, B, C and D. The regions A and D are full length column blocks, whereas the regions C and B are short column blocks. Similar to the object region, these regions are also rectangular in shape, and can be identified by the co-ordinates of the bounding box itself.

![Background region](image)

(a) Background region

![Background after concatenation](image)

(b) Background after concatenation

**Figure 6.5 Concatenation of the background region**

The labelled background regions are encoded using the DPCM coder. The output of the DPCM coders is converted into 1-D vectors and concatenated. The original background and the concatenated background are shown in Figure 6.5. The concatenated vectors are subsequently sent to the Huffman encoder, to generate the final bit stream. The block diagram representation of the background modelling process is shown in Figure 6.6.
After encoding the object and the background regions, the additional bytes required by the decoder are the size of the image and the coordinates of the rectangles corresponding to the object and the background. This information requires a few bytes only, and hence, does not affect the overall bit rate. The complete encoding process is explained in the flow chart shown in Figure 6.7.

Inverse processes are applied at the decoder end to generate a decompressed image. Since the object region is transmitted prior to the background in the bit stream, it is decoded before the rest of the image.

6.4 SIMULATION RESULTS

In order to evaluate the OBLC encoder, different medical images like the brain, coronary and elbow are selected. Test images are selected based on their object content.
When the size of the object region is more than 50% of the total image size, it is considered as a large object. If the object size is less than 25%, then the object is considered as small. Any size between 25% and 50% is taken as medium. The object region is smaller than the background region for the MRI brain images, whereas the coronary and the CT head images contain large object regions. The elbow image has a medium object region. The sample images from the test datasets are shown in Figure 6.8.
The performance evaluation of the OBLC encoder is conducted by applying it to the selected set of medical images. Within the same modality, images having objects of different sizes are selected. For comparison purposes, the test images are also compressed using the object based DPCM coder, object based wavelet coder and the direct coders. The 5/3 transform is used in the wavelet coders, as it is the standard wavelet used by the JPEG2000 coder for lossless compression. Table 6.1 shows the comparison of the bit rate obtained, using different object based coders.
Table 6.1  Bit rate comparison for different coders

<table>
<thead>
<tr>
<th>Image</th>
<th>Size of the image</th>
<th>Size of the Object</th>
<th>Direct IWT</th>
<th>Object Based IWT</th>
<th>Direct DPCM</th>
<th>Object Based DPCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR-1</td>
<td>256 x 256</td>
<td>small</td>
<td>2.07</td>
<td>1.99</td>
<td>2.01</td>
<td>1.91</td>
</tr>
<tr>
<td>MR-2</td>
<td>256 x 256</td>
<td>small</td>
<td>1.97</td>
<td>1.91</td>
<td>1.93</td>
<td>1.84</td>
</tr>
<tr>
<td>MR-3</td>
<td>256 x 256</td>
<td>small</td>
<td>2.25</td>
<td>2.20</td>
<td>2.14</td>
<td>2.05</td>
</tr>
<tr>
<td>MR-4</td>
<td>256 x 256</td>
<td>small</td>
<td>1.82</td>
<td>1.77</td>
<td>1.78</td>
<td>1.71</td>
</tr>
<tr>
<td>MR-5</td>
<td>256 x 256</td>
<td>small</td>
<td>1.93</td>
<td>1.87</td>
<td>1.89</td>
<td>1.80</td>
</tr>
<tr>
<td>CT-1</td>
<td>256 x 256</td>
<td>large</td>
<td>1.94</td>
<td>1.94</td>
<td>1.73</td>
<td>1.73</td>
</tr>
<tr>
<td>CT-2</td>
<td>256 x 256</td>
<td>large</td>
<td>1.97</td>
<td>1.97</td>
<td>1.80</td>
<td>1.80</td>
</tr>
<tr>
<td>CT-3</td>
<td>256 x 256</td>
<td>large</td>
<td>2.25</td>
<td>2.25</td>
<td>2.05</td>
<td>2.01</td>
</tr>
<tr>
<td>CT-4</td>
<td>256 x 256</td>
<td>large</td>
<td>1.94</td>
<td>1.94</td>
<td>1.73</td>
<td>1.72</td>
</tr>
<tr>
<td>CT-5</td>
<td>682x756</td>
<td>large</td>
<td>1.95</td>
<td>1.90</td>
<td>1.80</td>
<td>1.75</td>
</tr>
<tr>
<td>CT-6</td>
<td>682x743</td>
<td>medium</td>
<td>1.39</td>
<td>1.34</td>
<td>1.35</td>
<td>1.30</td>
</tr>
<tr>
<td>CT-7</td>
<td>512 x 512</td>
<td>medium</td>
<td>1.55</td>
<td>1.55</td>
<td>1.45</td>
<td>1.43</td>
</tr>
<tr>
<td>CT-8</td>
<td>674x1080</td>
<td>medium</td>
<td>0.97</td>
<td>0.97</td>
<td>0.93</td>
<td>0.91</td>
</tr>
<tr>
<td>Elbow</td>
<td>512 x 512</td>
<td>medium</td>
<td>0.85</td>
<td>0.85</td>
<td>0.745</td>
<td>0.73</td>
</tr>
<tr>
<td><strong>Average bpp</strong></td>
<td></td>
<td></td>
<td><strong>1.77</strong></td>
<td><strong>1.75</strong></td>
<td><strong>1.66</strong></td>
<td><strong>1.62</strong></td>
</tr>
</tbody>
</table>

Between the two direct coders, the DPCM based coder has a lesser bit rate than the IWT based direct coder. Similar results are observed for the object based coders also. Further, the performance of the object based coder is related to the size of the object region. For all object regions, the IWT based object coder shows similar or slightly better bit rates than the direct coder. However, the DPCM coder behaves differently. For large object regions, there is not much improvement, whereas for small and medium object regions, considerable reduction in the bit rate is obtained. The difference in
the performance of the object based DPCM coder is due to the modelling of the background region.

Another significant observation is the bit rate achieved by the object based coders. For all the sample images, both object based coders demonstrated a lesser bit rate than the direct coders. This is in contrast to the MAXSHIFT algorithm, which was gave a better bit rate for the ROI based approach, when used for lossless coding. The average bit rates achieved by the direct coders and the object based coders are shown in Figure 6.9.

![Figure 6.9 Average bpp for various lossless coders](image)

It is obvious from Figure 6.9 that the least average bit rate is obtained by the object based DPCM coder. The reduction in the bit rate of the object based DPCM coder is about 2% compared to the direct DPCM, and 9% compared to the direct wavelet coder. Although the reduction is small for a single image, it will have a significant impact if applied to volumetric images.
6.5 CONCLUSION

A novel OBLC coder for the lossless encoding of medical images is explained in this Chapter. A rectangular object region is extracted from the image using segmentation techniques, and encoded using a block based DPCM. The background region is split into four rectangular regions and encoded using the DPCM. Afterwards, the DPCM coded regions are converted into 1-D vectors, concatenated and entropy coded. The only source of coding overhead is the information about the co-ordinates of all the rectangles, which is minimal. At the decoder end, the object and the background regions are combined back to reconstruct the image.

The object based processing eliminates the ROI mask requirement, which is required in the MAXSHIFT algorithm based coding approach. The OBLC coder is tested with different types of medical images and the bit rates are calculated. The bit rates are also calculated for the object based wavelet coders and direct coders. The simulation results show that the DPCM based OBLC coder outperforms the direct coders and the wavelet based OBLC coder in terms of the bit rate. A high compression ratio for the lossless compression of images is thus achieved, through the OBLC technique.

This method is effective only when the images contain a dominant object region different from the background. Brain images, face images are examples for such images. Otherwise, the whole image will be considered as the object and there will be no background region, effectively reducing the technique to a normal DPCM.