## INDEX

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>TOPIC</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Motivation</td>
<td>3</td>
</tr>
<tr>
<td>(2)</td>
<td>Abstract</td>
<td>3</td>
</tr>
<tr>
<td>(3)</td>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>(4)</td>
<td>Mixed Raster Content (MRC) Imaging Model</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Decomposition and Compression Analysis</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>RD plot Modification</td>
<td>8</td>
</tr>
<tr>
<td>(5)</td>
<td>Resolution Enhanced Rendering</td>
<td>10</td>
</tr>
<tr>
<td>(6)</td>
<td>Iterative Pre-and Post processing for MRC layers</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Edge Sharpening</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Data-filling</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Edge softening</td>
<td>18</td>
</tr>
<tr>
<td>(7)</td>
<td>JPEG matched MRC compression of compound documents</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Pictorial and Text compression comparison</td>
<td>22</td>
</tr>
<tr>
<td>(8)</td>
<td>Shape Primitive Extraction and Coding (SPEC)</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>(1) The System</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>(2) Segmentation</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>(3) Coding of pixels</td>
<td>28</td>
</tr>
<tr>
<td>(9)</td>
<td>References</td>
<td>30</td>
</tr>
<tr>
<td>(10)</td>
<td>Acronyms</td>
<td>32</td>
</tr>
</tbody>
</table>
Submitted to:  Dr. K. R. Rao

Submitted by: Pritesh Shah
(UTA ID: 1000555858; E-mail: priteshrasiklal.shah@mavs.uta.edu)

Title: Implementation of Mixed Raster Content (MRC) for Compound Image Compression

Motivation:

In today’s world it is impossible to imagine a day without information or without information exchange in digital form over internet. With new advances in data processing systems and scanning devices, documents are present in a wide variety of printing systems. Documents in digital form are easy to store and edit, and can also be transmitted within seconds. These documents may contain text, graphics and pictures. So their storage requires huge memory and also transmission requires high compression and bit rates to avoid expenses and delay.

Abstract:

Mixed raster content (MRC) is described in the following report. It is a compression method which will be described here for compression of compound images (images containing both, continuous tone images and binary text) [1]. It uses a multi-layer, multi-resolution representation of a compound document. Instead of using a single algorithm, it uses multiple compression algorithms, including the ones specifically developed for text and images. So it can combine the best of new or existing algorithms and can provide various quality-compression ratio tradeoffs. Also the problems faced by MRC compression are described along with its solutions and related topics. These topics include the problems with soft edges, edge sharpening, data-filling (with algorithms) [6], edge softening [4], resolution enhanced rendering (RER) [2] and optimizing block threshold segmentation for MRC compression [5], [9]. Also a comparison is made with JPEG standard [7], [8]. Tables and images after using different
compression techniques are shown. Most of the algorithms use the standard MRC with a 3-layer representation which has some image processing overhead due to the management of multiple algorithms and the imaging model, and is also computationally expensive. A hybrid image coding scheme, shape primitive extraction and coding (SPEC) is also described which provides an accurate segmentation algorithm that separates the image into two classes of pixels: text/graphics and pictures [10], [11]. SPEC also provides a lossless coding scheme with low complexity algorithm for segmentation and good compression ratio along with excellent visual quality.

Introduction:

With the new advancement in scanning devices and data processing systems, documents are now present in variety of digital forms. These documents may contain pictures, text and graphics. In many cases documents with images are stored in high resolution and so their sizes are invariably large and commonly consume large amounts of storage space. Also these days information transform has been inevitable. So transferring such huge amount of data is difficult and expensive. So data compression becomes inevitable. There are different algorithms proposed and being used for different applications. There is no such single best algorithm for all image types or applications [1].

Different types of images require different coding fidelities. The preservation of edges and shapes of characters is important to facilitate reading. After the text is binarized, it has a lossless compression since coding errors in text can be easily noticed. But as far as continuous tone images are concerned, the human visual system works differently due to the richness of patterns and frequency contents. Text requires much higher resolution than pictures.

When compressing a compound image (scanned image containing both text and pictures), the differences between text and continuous-tone images becomes significant. For pictures, lossy
continuous-tone compression while for text, lossless binary compression is suitable. MRC allows both to be used within a single raster image. The raster is decomposed into several image layers with different image classes. So the different layers can be compressed differently with individual efficient compression systems.

**Mixed Raster Content (MRC) Imaging Model**

Encoding of compound images in MRC is carried out using a multi-layered, multi-resolution imaging model. The 3-layer MRC model contains 2 colored image layers (foreground (FG) and background (BG)) and one binary image layer (mask) [1]. Mask layer is the decisive layer in reconstructing the image from the FG and BG. When the pixel value in the mask layer is 1, the corresponding pixel from the FG is selected and when it is 0, the corresponding pixel from the BG is selected and the final image is reconstructed as shown in Fig. 1. The final image can be thought as the FG poured onto the BG through the Mask layer. This is MRC’s most common form. Also there can be more FG planes poured through different Mask planes onto the same BG plane and this process can be repeated several times.

![Fig. 1: Basic 3 plane configuration for MRC imaging model. Foreground plane is poured into background through the mask plane [2].](image-url)

**Mixed Raster Content for Compound Image Compression**
After the decomposition of the original image into layers, each layer can be individually processed and compressed using different algorithms as per Fig. 3. Resolution change or color mapping can be a part of this image processing technique. For good compression and reduced distortion visibility, the resolution and compression algorithm used for a particular layer would be matched to the layer’s content [1]. The compressed layers after being packaged in a format like TIFF-FX or ITU-T MRC data-stream are delivered to the decoder where the individual planes are retrieved, decompressed, processed and the image is composed using MRC imaging model.

Depending on its content, a page may be represented as one, two, three or more layers. A page containing black and white text can use FG and BG layers defaulted to black and white along with the mask layer.
Compression algorithms such as JPEG or wavelet encoders are designed for compression of natural images while they yield very poor quality/bit rate tradeoffs for typical document images. This is observed when the document contains pictures, graphic content, text, and line art. Since last few years, MRC is being used for document encoding as it improves image quality or reduces bit rate to a great extent when compared to conventional picture encoders.

Decomposition and compression analysis

It can be seen from Fig. 3 that how the decomposition process and the compressors and their associated parameters for each plane provide a good degree of freedom in the MRC based compression. The operation of the encoder is affected by decomposition, while the decoder remains unaffected. Fig. 4 describes the decomposition process. The basic approaches are: region classification (RC) and transition identification (TI). In case of RC decomposition, the regions containing graphics and text are represented in a separate FG plane. Everything is represented in FG plane including the spaces in between letters and other blank spaces. The mask as shown in the figure is uniform with large patches, clearly differentiating the text and the graphic regions and the background contains the document background, complex graphics and/or continuous tone pictures [1]. TI decomposition is quite similar to RC decomposition as can be seen from Fig. 4. However, in case of TI, mask and FG planes represent graphics and text in a different manner. The FG plane pours the ink of text and graphics through the mask onto the BG plane. So the mask should have the necessary text contours. Hence the mask layer

\[
\text{FG} = \text{M} + \text{BG}
\]

\[
\text{FG} = \text{M} + \text{BG}
\]

**Fig. 4: Typical decomposition approaches yielding the same reconstructed image (in the absence of processing or compression) [1].**
contains text characters, line art and filled regions, while the FG layer contains colors of the text letters and graphics.

In RC decomposition, the mask layer is very uniform and very well compressible. But the FG can contain edges and continuous tone details. So it cannot be compressed well with typical continuous tone coders such as JPEG [1]. As the mask layer in case of TI contains text objects and edges, it can be efficiently encoded using standard binary coders such as JBIG and JBIG-2 [1]. The FG plane can be very efficiently coded even with coders such as JPEG because it contains large uniform patches. Also the FG plane can be sub-sampled without much loss in image quality [1].

**RD plot Modification**

The benefit of using MRC model for compression can be observed by analysing its rate-distortion (RD) characteristics. As shown in Fig. 5, if image (a) is compressed with a generic coder A with fixed parameters except for a compression parameter, it will operate under a RD plot as shown in (b). It is seen that another coder B under the same circumstances is found to perform better than coder A if its RD plot is shifted to the left as shown in (c). The logic for MRC is to split the image into multiple planes as shown in (d), and to apply coders C, D and E to each plane with RD plots similar to that of coder B. Thus the equivalent coder may have better RD plots than A, but there would be an overhead associated with a multi-plane representation.
Fig. 5: Interpretation of the multi-plane approach as a means to modify the RD characteristics of coding mixed images. (a) Mixed image; (b) RD plot for given coder A; (c) modified RD plot for resulting coder B; (d) coder B may be achieved by plane decomposition, where each plane undergoes a better tailored coder, thus achieving better RD plots [1].
Resolution Enhanced Rendering

Conventional MRC Encoder:
There is some inefficiency in MRC model because all the layers are compressed independently and so the layers which are not used are also encoded. The FG and BG layers are compressed at lower resolutions with lossy image coders such as JPEG or JPEG2000 and mask layer is encoded with JBIG or JBIG2[2].

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Associated Company</th>
<th>Subsampling Factor</th>
<th>FG/BG Encoder</th>
<th>Mask Encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digipaper</td>
<td>Xerox</td>
<td>N/A</td>
<td>FG: Color Tags</td>
<td>Token-based</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BG: JPEG</td>
<td></td>
</tr>
<tr>
<td>DjVu</td>
<td>LizardTech, AT&amp;T</td>
<td>6</td>
<td>Wavelet-based</td>
<td>Modified JBIG2</td>
</tr>
<tr>
<td>LuraDocument</td>
<td>LuraTech</td>
<td>3</td>
<td>JPEG2000</td>
<td>CCITT G4</td>
</tr>
<tr>
<td>JPEG-matched MRC</td>
<td>HP</td>
<td>2</td>
<td>JPEG</td>
<td>JBIG</td>
</tr>
<tr>
<td>JPEG2000-matched MRC</td>
<td>HP</td>
<td>2</td>
<td>JPEG2000</td>
<td>JBIG</td>
</tr>
</tbody>
</table>

Resolution enhanced rendering method:

The RER algorithm [2] functions as described below. The FG and BG are segmented using an adaptive error diffusion method. The binary mask is dithered along the edges of the characters to represent the gradual transition of the scanned text characters. The true value of the document pixels are then estimated by the RER decoder with the help of mask, FG and BG layers. This estimation is done using a non-linear tree structured predictor. This encoder is trained to identify the characteristic patterns of the RER encoder. So it can estimate the true pixel values more accurately.
Fig. 6: Training model of the optimized encoder and decoder. Once training is complete, the encoder and decoder function independently [2].

Fig. 6 shows the minimization of distortion of the decoded document carried out jointly by the encoder and decoder. Both the encoder and decoder have parameters which can be trained to produce very good results.

During each particular process, the parameters of either the encoder or decoder are fixed (alternatively) and the parameters of the other are optimized. Two disjoint sets of documents are used for training the encoder and decoder. The simulation results of the above process have shown that parameters have been reached which can reduce the distortion of the decoded document. Also it has been found that optimization of both encoder and decoder together performs better than the individual optimization of the two [2].
In Fig. 7(b), the RER encoding module creates the dithered mask \( D_s \) as the output by taking in the FG, BG, binary mask layer, and the original document as the input. The \( D_s \), FG and BG are separately compressed by using the binary image encoder and continuous-tone encoder that are used for the MRC encoder.

In the encoding module, an edge detection procedure is performed on the binary mask layer to determine the pixels that lie on the boundary between FG and BG, to compute dithered mask. If \( K_s \) is 1, then \( s \) is a boundary pixel or it is a non-boundary pixel. \( \lambda_s \) is obtained by Linear Projection. Then at each boundary pixel \( D_s \) is generated from \( \lambda_s \) by switching on the error diffusion procedure. The error diffusion procedure is switched off at each non-boundary pixel and the binary mask value, \( M_s \), is the output, \( D_s \). Thus the binary output image of the dithered mask is produced.
In Fig. 7, $X_s$ is the pixel color in the raster document at location $s$. $F_s$ and $B_s$ are the decimated FG and BG colors at $s$. $M_s$ and $D_s$ are the binary and dithered masks respectively. $F_s - (\tilde{F}_s)$ and $B_s - (\tilde{B}_s)$ be the interpolated FG and BG colors at $s$. $K_s$ is the binary output of the boundary detection procedure. $\lambda_s$ is the output of the linear projector [2]. ‘t’ in equation (1) stands for transpose.

The solution of $\lambda_s$ is,

$$\lambda_s = \frac{(X_s - \tilde{B}_s)^t(\tilde{F}_s - \tilde{B}_s)}{(\tilde{F}_s - \tilde{B}_s)^t(\tilde{F}_s - \tilde{B}_s)}$$

(1)

Fig. 8: (a) Structure of an MRC decoder. (b) Basic structure of a RER-enhanced MRC decoder [2].
Fig. 8(a) and 8(b) show the working of an MRC and RER decoder respectively. In an RER decoder, first, the binary mask, Mₜₛ, is input to a boundary detection module, which works exactly the same way as in the RER encoder. It is assumed that the decoded FG and BG layers have lower resolution than the original. Also any interpolating algorithm can be applied to interpolate the two layers. Let Fₛ⁻ and Bₛ⁻ be the decoded FG and BG layers that have been interpolated respectively, both having the same resolution as before. Xₛ⁻ (Xₛ) is the reconstructed color pixel for each non-boundary pixel and is chosen between the FG and the BG colors, according to the binary mask as in the normal MRC. To show the reconstructed pixel color if the pixel is on the boundary, a nonlinear prediction algorithm is performed [2]. It works by computing, λₛ⁻ (Xₛ), a minimum mean-squared error (MSE) estimate of λₛ. Using this estimate, we can have the following relation,

\[ \hat{X}_s = \hat{\lambda}_s \hat{F}_s + (1 - \hat{\lambda}_s) \hat{B}_s \]  

(2)

**Fig. 9: RER predictor using tree-based nonlinear estimation [2].**
Fig. 9 contains a tree-based classifier and a set of class-dependent least-square estimators. A binary mask is extracted in a 5 x 5 window about the pixel in question. The binary vector formed, \( Z_s \), is then input to a binary regression tree predictor known as tree-based resolution synthesis (TBRS). A two step process, classification and prediction, is used by the TBRS predictor to estimate the value of \( \lambda_s \). Firstly it classifies the vector \( Z_s \) into one of the M classes using a binary tree predictor. Each class, then has a corresponding linear prediction filter to estimate the value of \( \lambda_s \) from \( Z_s \) by using the following relation

\[
\hat{\lambda}_s = A_m z_s + \beta_m
\]  

where, \( m \) is the determined class of vector \( Z_s \); \( A_m \) and \( \beta_m \) are the corresponding linear prediction parameters of class \( m \).

**Fig.10**: Example of binary tree prediction model (a) The regression tree structure \( e_i \) and \( \mu_i \) specify the binary decision rule for splitting node \( i \). \( A_m \) and \( \beta_m \) are the linear filter parameters of least-square estimation for leaf (or class) \( m \). (b) The recursive classification procedure of the binary tree in the input vector space \( Z_m \) [2].
Fig. 10 shows the structure of the binary tree prediction model and its classification procedure. In Fig. 10(a), the tree is to be traversed down from the root which is at the top. A binary decision is to be made at each non-terminal node that the input vector \( Z_s \) passes through [2]. The termination is at one of the leaves at the bottom, where each leaf represents a distinct class of the input vector space \( Z_s \). At any non-terminal node \( i \), the input vector is determined to go to the left child or the right child using the specific splitting rule

\[
 e_i^T (z_s - \mu_i) \geq 0
\]  

where, \( e_i \) is a pre-computed vector specifying the orientation of a hyperplane for splitting, \( e_i^T \) is the transpose of \( e_i \) and \( \mu_i \) is a reference point on the hyperplane. If the projection of \( z_s - \mu_i \) is negative, then \( z_s \) falls into the left side of the hyperplane passing through the reference point \( \mu_i \) and perpendicular to the vector \( e_i \) and consequently goes down to the left child of node \( i \). Or, it falls to the right side of the hyperplane and goes down to the right child.

Fig. 10(b) illustrates an example of the recursive classification procedure in the input vector space \( Z_s \). Each polygon region at the bottom level represents a different class of \( Z_s \). The distinct regions of the document corresponding to mask edges of different shapes and orientations can be separated by the classification step [2].

Thus RER leads to minimum document image distortion. Simulation results indicate that at a fixed bit rate, RER can reduce distortion to a great extent. Also RER is fully compatible with the standard MRC encoders and decoders.

**Iterative Pre-and Post processing for MRC layers**

The binary nature of the mask layer makes it difficult for MRC compression to deal with scanned data and soft edges. “halo” is observed at the object edges when MRC model is used. This is because the edge transitions do not belong fully to either FG or BG [4].
In MRC after the original single resolution image is decomposed into layers, they are processed and compressed using different algorithms. Fig. 11 shows how the original document X is input to the pre-processor which yields output Y (3-layer MRC model). It also constructs an edge sharpening map and estimates the original edge “softness”. This information is termed as side information. This data is then encoded and then the encoded data goes to the decoder, where the reconstructed data Y’, along with the side information is used by the post-processor to assemble the reconstructed version X’ of the original document.

**Fig. 11: Working of MRC with scanned data [4].**

**Problem with soft edges**

In the mask layer in MRC, the size and shape of the text is same as in the imaged text. The edges of text and graphics, when scanned are not very sharp and they are referred to as “soft edges”. The selector plane (mask) is binary and the edge transitions are smooth, so it is not possible to contain all the BG in one plane and all the FG in the other [4]. This causes halo around the text in FG/BG planes which is very damaging to compression.

**Edge Sharpening**

This halo cannot be removed with data-filling, forcing to change the data itself. It cannot be assumed that all image edges can cause problems. Also not all mask transitions are going to cause halo in the scanned material [4]. This occurs only when the edges of the image and the transition of the mask coincide.
Data-filling

There may be unused pixels in FG or BG layers because the final selected pixels can be from either layer depending on the mask. So in order to increase compression ratio, these pixels can be replaced by any color. This can be seen in the Fig. 12 [4], [6]. Once the pre-processor and input are fixed, segmenter finds a binary mask, with which the pre-processor can derive the output layers according to the input.

![Diagram](image.png)

*Fig. 12: MRC decomposer (based on a segmenter and plane filling algorithm [6].)*

Edge softening
Although intentionally blurring an image is not a common practice since the edges in the image were originally soft, soft edges at the decoder can be expected from a good compression system. The edges are sharpened to accommodate the mask, so an estimation of the image edge transition is necessary [4].

A few parameters about the soft edges coinciding with the mask are to be sent to the decoder, so that it can soften the image around the mask borders. One parameter can be the width of the edge, or how soft the edges are while the other can give an idea of the offset of the mask border with the image edges and is related to the thresholding parameters used to find the mask. Also a linear filter can be used. The encoder has access to the recovered and the original unsharpened image. The encoder can find the optimal filter coefficients say by least squares, and send all filter parameters to the decoder at a low cost.

**JPEG matched MRC compression of Compound documents**

The Fig. 13 shows the MRC encoder. In order to maintain tractable run-time memory requirement, the algorithm works on independent stripes of image data rather than a full image [7]. For each of the stripes there are practically ample number of possible decompositions and associated stripe encoding parameters. But we need to obtain a near-optimal decomposition in terms of good compression and reconstructed image.
Fig. 13: Schematic of MRC encoder [7].

The analysis algorithm should analyze the input stripe thoroughly and also should consider the characteristics of the particular coders that are to be used to code the decomposed layers after the analysis.

JPEG [7] has been chosen for coding the FG and BG layers as it is omnipresent and an efficient image codec. The decomposition scheme developed is matched to the way JPEG operates. The mask layer is coded with JBIG [7].
Fig. 14: (a) Original 300 dpi full color National Geographic scan, and (b) Reconstructed document, with file sizes, along with (c) the binary Mask Layer. The MRC codec uses 2x2 sub-sampling of image layers [7].

Fig. 14 represents the compression results for a 300 dpi scan from the National Geographic. The mask layer as can be seen performs a nice job distinguishing text and other features from the document. Thus this full-color compound document codec is compliant with the ITU standard T.44 [7]. JPEG was used for FG and BG layers and JBIG for the mask layer. This approach makes the encoding complexity low enough to make scan-to-email, scan-to-web, or scan-and-distribute type applications feasible [7]. To enable viewing documents on machines with varying capabilities, resolution scalability is implemented at the decoder.
Fig. 15: Images representing the magnitude of the error after decompression (scaled × 30, white = zero): (a) original compound image (600 dpi page); (b) after JPEG compression; (c) after MRC-JPEG compression [8].

Table II
Results of different coding methods applied to compound image ‘Toy Store’ (Fig.11a) [8].

<table>
<thead>
<tr>
<th>Coding method</th>
<th>Compression ratio</th>
<th>Encoding/decoding time</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>24.6:1</td>
<td>5 s / 4 s</td>
</tr>
<tr>
<td>WinZip</td>
<td>21.0:1</td>
<td>18 s / 7 s</td>
</tr>
<tr>
<td>JPEG LS</td>
<td>21.5:1</td>
<td>4 s / 7 s</td>
</tr>
<tr>
<td>MRC-JPEG</td>
<td>166.3:1</td>
<td>16 s / 11 s</td>
</tr>
</tbody>
</table>

Fig.15 shows the evaluation of the visual quality by showing the error after decompression [8]. These results correspond to the compression ratios in Table II. It can be seen from this figure that JPEG produces large errors around the text whereas MRC decomposition and coding can improve both the compression ratios and quality to a great extent.
Pictorial and Text compression comparison between JPEG and JPEG matched MRC compression

Fig. 16 and Fig. 17 will show the comparison of the JPEG and JPEG matched MRC images along with the original image. It shows the comparison both in text and pictures. The simulation results show how JPEG matched MRC is better than JPEG. This is true in both, the text and the pictorial case.

(a)   (b)
Fig. 16: Enlarged portion of the text region of original and reconstructed images at a compression ratio of 70:1.
(a) Original, (b) JPEG, and (c) MRC using JPEG + MMR [12] + JPEG [1].
SPEC (Shape Primitive Extraction and Coding)

In case of a compound image compression, most of the algorithms use the standard 3-layer MRC model. But this model is computationally expensive and has some image-processing overhead to manage multiple algorithms and the imaging model. SPEC [10] decomposes the image into two sets of pixels: text/graphics and pictures. Its unique features are that segmentation and coding are tightly integrated in the SPEC algorithm. Also shape and color serve as two basic features to the segmentation and the lossless coding.

SPEC Algorithm

(1) The System:
Fig. 18 shows the proposed SPEC algorithm consisting of two stages – segmentation and coding. Firstly, the algorithms segments 16 x 16 non-overlapping blocks of pixels into text/graphics class and picture class [11]. Then the text/graphics pixels are compressed with a new lossless coding algorithm and the rest of the pixels are compressed with lossy JPEG.

SPEC uses four types of shape primitives: isolated pixels, horizontal lines, vertical lines and rectangles. These shape primitives are represented by a color tag and its position information, i.e., (x, y) is for an isolated pixel, (x, y, w) for a horizontal line, (x, y, h) for a vertical line, and (x, y, w, h) for a rectangle, respectively.

(2) **Segmentation:**

The two steps that comprise segmentation are block classification and refinement segmentation as shown in Fig. 19. In step one, the 16 x 16 non-overlapping blocks are scanned to count the number of different colors. The block is classified as the picture block if the color
number is larger than a certain threshold value $T_1$ ($T_1=32$ is used for SPEC) [11]. Else it is classified as text/graphics block. This classification can be extremely fast.

To increase the performance refinement segmentation is done to extract text and graphics pixels, as the first step is coarse. Its implementation is carried out by scanning each picture block in order to extract four types of shape primitives. The policy implemented here is size-first. The classification of a shape primitive depends on its size and color. The shape primitive is extracted as text/graphics pixels if its size is larger than the specified threshold $T_2$ ($T_2=3$ for SPEC). Else its color is compared to a dynamic palette of recent text/graphics. The shape primitive is considered as text/graphics if an exact color match is found [11]. A shape primitive’s color is put into the dynamic color palette if its size is larger than a threshold $T_3$ ($T_3=5$ for SPEC). The implementation of the dynamic color palette is carried out with a first-in first-out buffer and eight entries. It is of small size for computational efficiency. A shape primitive with pure black or pure white color is directly classified into text/graphics. Thus the segmentation process segments the image into two parts – text/graphics pixels (shape primitive pixels and all pixels of text/graphics) and pictorial pixels (the remaining pixels in the picture blocks).
Fig. 19: Flow-chart of segmentation [10].

Fig. 20: Flow-chart of coding [10].
(3) **Coding of Pixels:**

The SPEC coding process is shown in Fig. 20. The pictorial pixels are compressed with the lossy JPEG algorithm, while the text/graphics pixels are compressed with a new lossless coding algorithm which is based on shape primitives.

**Lossless Coding of text/graphics pixels**

The shape primitives are extracted first to compress the text/graphics blocks. The extraction procedure is similar to that of the picture blocks. The encoding scheme that represents the counts four types of shape primitives and their corresponding values is used for each color. Too many small shape primitives in a complex block makes shape-based coding inefficient. So palette-based coding is a good alternative. The most shape complex color is palette-based coded whereas the other colors are coded shape based in order to attain minimal code length. When all colors are shape-based coded, the most shape-complex color is the color that generates the maximum coding length.

A color table reuse technique is applied by SPEC to represent the colors of shape primitives. In general, most colors in the color tables of two consecutive blocks are the same. If a color of the current block is found to match the color of the previous block, it is represented by a 1-byte index. Else it is represented by a 3-byte (R, G, B) format.

**Pictorial Pixel coding with JPEG**

A simple JPEG coder is used to compress pictorial pixels in picture blocks in SPEC. Text/graphics pixels in the picture block are removed before the JPEG coding in order to reduce ringing artifacts and to achieve higher compression ratio. These pixels are coded by a lossless algorithm. Their values can be arbitrarily chosen, but its preferable that these values are similar to the neighbor pictorial pixels to produce a smooth picture block. So these holes are filled in with the average color of pictorial pixels in the block.
Thus an accurate algorithm is developed to separate text/graphics from pictures and a lossless coding method is designed for text/graphics compression.
References:


### Acronyms:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG</td>
<td>Background</td>
</tr>
<tr>
<td>CCITT, ITU-T</td>
<td>Telecommunication Standardization Sector of the International Telecommunications Union</td>
</tr>
<tr>
<td>Dpi</td>
<td>Dots per inch</td>
</tr>
<tr>
<td>FG</td>
<td>Foreground</td>
</tr>
<tr>
<td>HP</td>
<td>Hewlett-Packard</td>
</tr>
<tr>
<td>ICIP</td>
<td>International Conference on Image Processing</td>
</tr>
<tr>
<td>ITU-T</td>
<td>Telecommunication Standardization Sector (ITU-T) coordinates telecommunication standards on behalf of ITU (International Telecommunication Unit)</td>
</tr>
<tr>
<td>JBIG</td>
<td>Joint bi-level image experts group</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group</td>
</tr>
<tr>
<td>M</td>
<td>Mask</td>
</tr>
<tr>
<td>MMR</td>
<td>Modified Modified read</td>
</tr>
<tr>
<td>MRC</td>
<td>Mixed raster content</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean square error</td>
</tr>
<tr>
<td>RC</td>
<td>Region classification</td>
</tr>
<tr>
<td>RD</td>
<td>Rate distortion</td>
</tr>
<tr>
<td>RER</td>
<td>Resolution enhanced rendering</td>
</tr>
<tr>
<td>SPEC</td>
<td>Shape primitive extraction and coding</td>
</tr>
<tr>
<td>TBRIS</td>
<td>Tree-based resolution synthesis</td>
</tr>
<tr>
<td>TI</td>
<td>Transition identification</td>
</tr>
<tr>
<td>TIFF-FX</td>
<td>Tagged image file format fax extended</td>
</tr>
</tbody>
</table>