Batch: B2	Roll Number: 16010121045
Experiment No.: 10	

Title : Image compression by lossless technique (Run Length Coding).

Objective: To understand image compression by lossless technique (Run Length Coding).

Expected Outcome of Experiment :

Course Outcome	Description
CO-5	Design and develop applications based on 1-D and 2-D digital signals.

Books / Journals / Websites referred :

- http://www.mathworks.com/support/
- www.math.mtu.edu/~msgocken/intro/intro.html
- www.mccormick.northwestern.edu/docs/efirst/matlab.pdf
- A.Nagoor Kani “Digital Signal Processing”, 2nd Edition, TMH Education.

❖ **Pre-Lab Prior concepts :**

Variable length code can be used to remove coding redundancy. One of the way to remove the inter pixel redundancy is run length coding. When inter pixel redundancy is removed by using run length coding the mapper transforms the input data in to usually non visual format. This operation is reversible and may or may not reduce directly the amount of data required to represent the image. Run length coding is the example of a mapping that directly results in data compression in the initial stage of overall source encoding process.

ALGORITHM:

Compression:

- Read the binary (monochrome) Image.
- Write the runs of the pixel in a text file.
- Use the text file created in a step 2 and writes each run by ● using 8 bits in output file to create compressed image.
- Use the compress image.
- Expand the runs to create decompressed image.

SOMAIYA VIDYAVIHAR UNIVERSITY
K. J. Somaiya College of Engineering, Mumbai -77

❖ Implementation Steps with Screenshots :

```
% Read the binary image
img = imread("binary_image.png");

% Convert to grayscale if the image has three channels
if size(img, 3) == 3
    img_gray = rgb2gray(img);
else
    img_gray = img;
end

% Initialize variables
encoded_data = zeros(1, 2*numel(img_gray));
count = 0;
prev_value = -1;
idx = 1;

% Encode the image data using run-length encoding
for i = 1:numel(img_gray)
    if img_gray(i) ~= prev_value
        if count > 0
            encoded_data(idx) = count;
            encoded_data(idx+1) = prev_value;
            idx = idx + 2;
        end
        prev_value = img_gray(i);
        count = 1;
    else
        count = count + 1;
    end
end

% Store the last count and value
encoded_data(idx) = count;
encoded_data(idx+1) = prev_value;
encoded_data = encoded_data(1:idx+1);

% Display encoded data
disp('Encoded data:');
disp(encoded_data);

% Decode the encoded data
decoded_img_gray = zeros(size(img_gray));
index = 1;

for i = 1:2:numel(encoded_data)
    count = encoded_data(i);
    value = encoded_data(i+1);
    decoded_img_gray(index:index+count-1) = value;
    index = index + count;
end

decoded_img_gray = reshape(decoded_img_gray, size(img_gray));

% Display original and decoded images
figure;
subplot(1, 2, 1);
imshow(img_gray);
title('Original Image');

subplot(1, 2, 2);
imshow(decoded_img_gray, []);
title('Decoded Image');
```

```
>> Exp10
  Columns 132,331 through 132,360

    1   223     1   228     2   230     1   229     1   227     1   228     2   229     3   228     4   227     3   226     3   225     4   224     3   223     4   222     4   221

  Columns 132,361 through 132,390

    3   220     4   219     3   218     4   217     4   216     2   215     4   214     3   213     4   212     4   211     3   210     4   209     3   208     1   207     3   206

  Columns 132,391 through 132,420

    1   204     1   192     1   158     2   240     2   241     1   242     3   243     1   244     1   245     1   247     4   248     2   249     2   250     2   251     5   252

  Columns 132,421 through 132,450

    1   250     1   202     1   193     1   231     1   253     3   254     4   253     2   254    67   255     2   254   106   255     2   250     1   255     1   230     1   194

  Columns 132,451 through 132,480

    1   199     1   197     1   116     1    60     1    46     1    50     1    34     1     5     1    26     1    67     1    66     1    43     1     8     1     3     1    30

  Columns 132,481 through 132,510

    1   127     1   228     1   230     1   234     1   217     1   125     1    56     1    30     2     6     1     4     1     1   196     0     1     1     1     6     1     5

  Columns 132,511 through 132,540

    1     0     1     2     1     8     1    47     1    62     1   146     1   194     1   209     1   228     2   230     2   228     2   229     3   228     4   227     3   226

  Columns 132,541 through 132,570

    3   225     4   224     3   223     4   222     4   221     3   220     4   219     3   218     4   217     4   216     2   215     4   214     3   213     4   212     4   211

  Columns 132,571 through 132,600

    3   210     4   209     3   208     1   207     2   206     1   205     1   203     1   191     1   157     2   240     2   241     1   242     3   243     1   244     1   245
```

Original Image

Decoded Image

Conclusion : By eliminating coding and inter pixel redundancy lossless compression is achieved. Here by applying run length coding interpixel redundancy is removed and relative data redundancy is calculated as: $R_D = 1 - 1/C_R$ where C_R is the compression ratio.

SOMAIYA VIDYAVIHAR UNIVERSITY
K. J. Somaiya College of Engineering, Mumbai -77

Post Lab Questions

1) Compare Lossy and lossless compression

Lossy and lossless compression are two methods used to reduce the size of digital files, particularly for media such as images, audio, and video. Here's a comparison between the two:

1. **Definition**:

- **Lossy Compression**: This method reduces file size by permanently eliminating some of the data. It achieves higher compression ratios but at the cost of losing some quality. It's commonly used for multimedia files like images, audio, and video.

- **Lossless Compression**: This method reduces file size without sacrificing any data. It achieves compression by identifying and eliminating redundant information within the data. Lossless compression is commonly used for text files, executable programs, and archival purposes.

2. **Quality**:

- **Lossy Compression**: The compression process results in a loss of data, which can lead to a reduction in quality. The degree of quality loss depends on the compression algorithm and the level of compression applied. While lossy compression can significantly reduce file size, it may introduce noticeable artifacts in the compressed data.

- **Lossless Compression**: No data is lost during the compression process, so the quality of the compressed file is identical to the original. It preserves all the original information, making it ideal for scenarios where maintaining fidelity is essential, such as in professional image editing or archival storage.

3. **Compression Ratios**:

- **Lossy Compression**: Typically achieves higher compression ratios compared to lossless compression. This means that lossy compression can significantly reduce file sizes, making it suitable for applications where minimizing storage or transmission bandwidth is critical.

- **Lossless Compression**: While it may not achieve compression ratios as high as lossy compression, it still offers substantial reductions in file size without sacrificing quality. Lossless compression is preferred in scenarios where maintaining data integrity is paramount, even if the reduction in file size is not as dramatic.

4. **Applications**:

- **Lossy Compression**: Widely used in scenarios where a certain level of quality loss is acceptable or even imperceptible to users. Common applications include multimedia streaming, web graphics, and digital photography.

- **Lossless Compression**: Preferred for applications where preserving every detail of the original data is necessary, such as medical imaging, legal documents, and scientific data analysis.

In summary, while lossy compression offers higher compression ratios and smaller file sizes, it does so at the expense of some data loss and quality degradation. Lossless compression, on the other hand, preserves all original data without any loss in quality, making it suitable for applications where data integrity is paramount. The choice between the two methods depends on the specific requirements of the application and the importance of maintaining data fidelity.

Date: **Signature of faculty In-charge**