Data Compression

COMP.SGN.240-2020-2021-1 Advanced Signal Processing Laboratory (Laboratory work)

Submitted to: Emre Kaya (emre.kaya@tuni.fi).

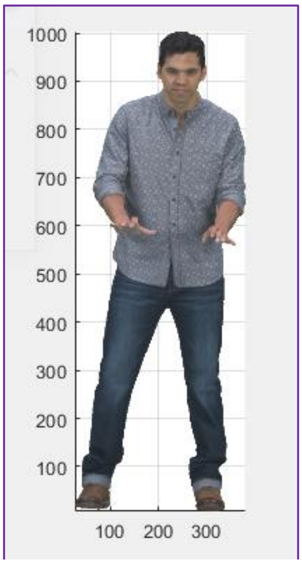
Submitted by: Pragya Singh (293206)
Aleksei Sapozhnikov (293200)

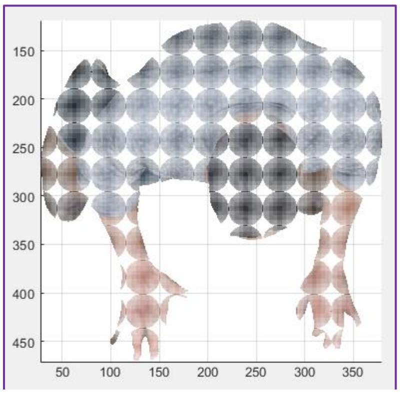
Introduction

Data compression is the process of reducing the amount of data needed for the storage or transmission of a given piece of information. We need to find ways to store and transmit our data in a more effective way, sometimes by compensating for losing some information (lossy compression), and in some cases, without losing any information (lossless compression). In this exercise we are studying the compression of image data using simple causal predictor and a lossless encoding algorithm (Golomb-Rice coding).

Task 1

We extracted 3 major projections (Front, Top and 1 Side Projection) as depicted in *Figure 1 from a* point cloud format file,





extracted assigned top/bottom projections as

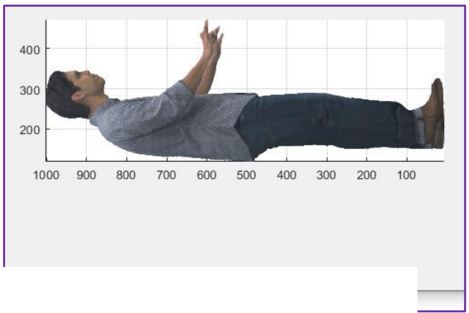


Figure 1 *Major projections*

depicted in *Figure 2*,

Figure 2 *Assigned Top/Bottom projections*

extracted grayscale conversion of top/bottom projections depicted in *Figure 3.*

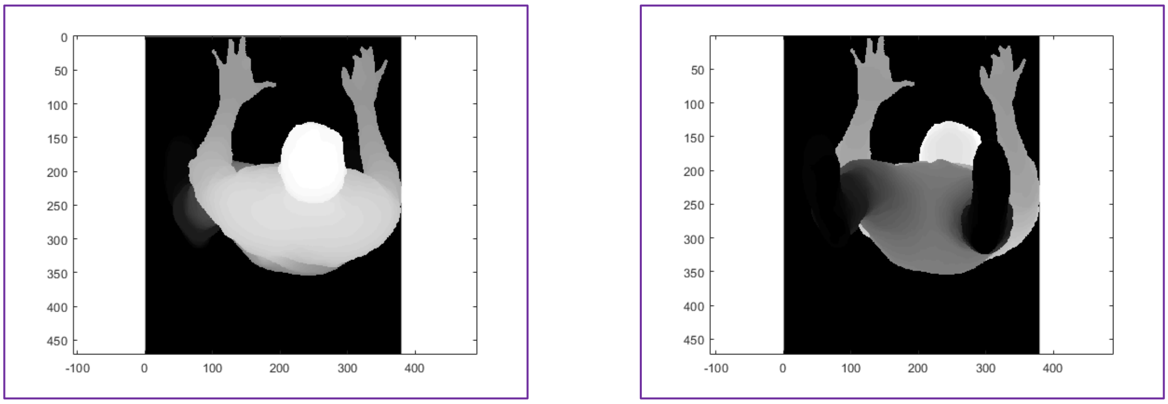


Figure 3 *Grayscale conversion of top/bottom projection*

We next Transform Acolor into a grayscale image **A** depicted in **Figure 4**.

Figure 4 *Grayscale image* ***A***

Task 2

Histogram of the values found in the matrix A. We used Semilogy(X,Y) function of MATLAB which plots x- and y-coordinates using a linear scale on the x-axis and a base-10 logarithmic scale on the y-axis.

Entropy equals to 2.4781.

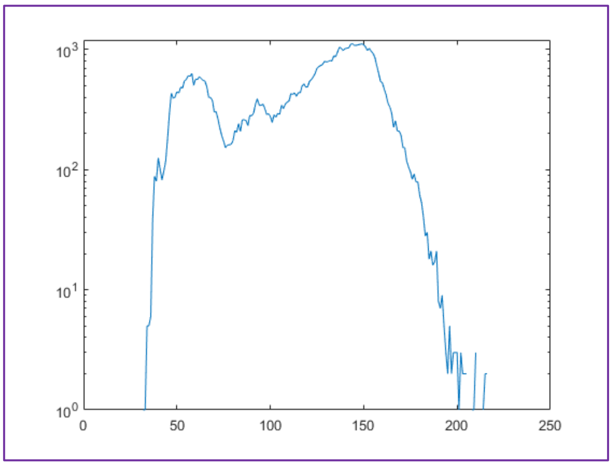


Figure 5 *Semilog plot of histogram values*

Task 3

Residual matrix **E** depicted in *Figure 6.*

Figure 6 *Residual matrix* ***E***

We have compressed the image by estimating values and calculating the residual matrix. Then the predictor estimation is represented as:

$$\hat{z} = \text{median}\{n, w, n + w - nw, w - n + ne, w + \frac{ne - nw}{2}\}$$

where n = A$(i-1,j)$, w = A$(i,j-1)$, nw = A$(i-1,j-1)$, ne = A$(i-1, j+1)$.

The estimation error is processed as matrix **E**(i, j) which is called residual matrix. As from the figure above we can analyze that the higher or lower values are mostly at the edge of object. Since value is close to zero, it results in sharing the same encoding region which can be encoded in fewer bits for each value, therefore, improves the compression ratio.

Task 4 and 5

We have further divided the matrix E into blocks (b x b). Then we apply Golomb-Rice coding with the number of least significant bits *p* belonging to the range [0, 8]. We applied closest value to $\log_2$ of the mean of the absolute value of a given block values. After that, each value within a block is encoded with same *p* value.

Task 6

We have further used Function lf_compress() which takes as arguments the input matrix A, and the compressed file path, the block size for the Golomb-Rice coding (b, in the range 32 . . . 128). At the start of every block's bit vector placed its p value in 8 bits representation. The number of block rows, block columns, padding pixels of vertical and horizontal axes are also saved at the beginning of bitstream.
The second step is lf decompress() which takes as an argument the source file (compressed file) and returns the reconstructed matrix. We have decompressed the image by reconstructing the residual matrix from the data in encoded file and then further reconstructing the image by summing up the residual matrix E and the estimation.

Task 7 and 8

Results of compression showed that compression rate is more than 50 percent, and restoring images we had no loss in information. We calculated mean absolute error for all compressed images, MAE equals to 0 for all the images.

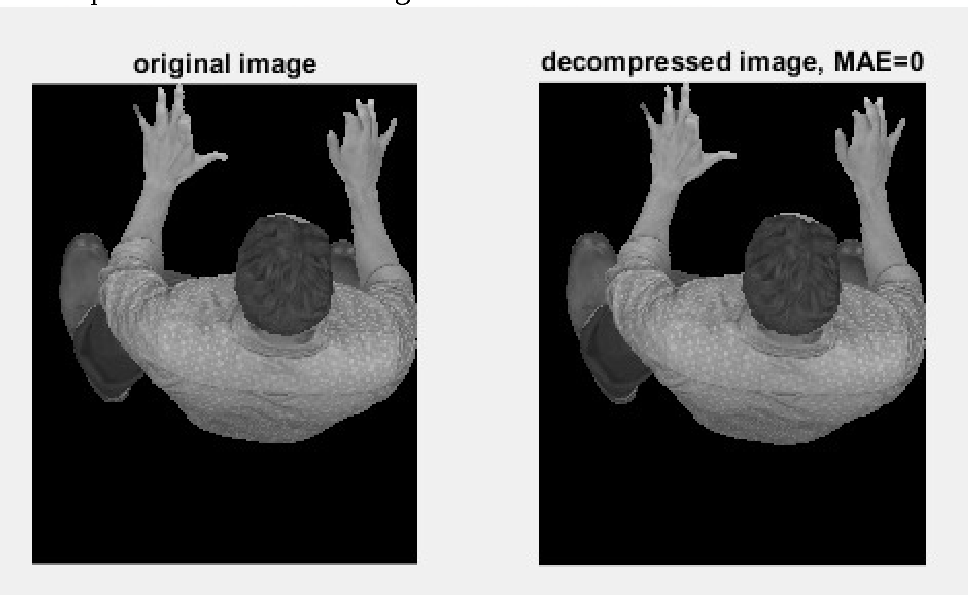


Figure 7. Result for the depth image

The compressed file size increases with greater b parameter. The best compression obtained is from the smallest block size 32. The cause for this is the fact that p value is more optimized for the smaller block size. We can also see that when b is small the computational time increase.

Also, from the graph we can deduce that file size fluctuates more with larger block size. Visualization is in the graph below.

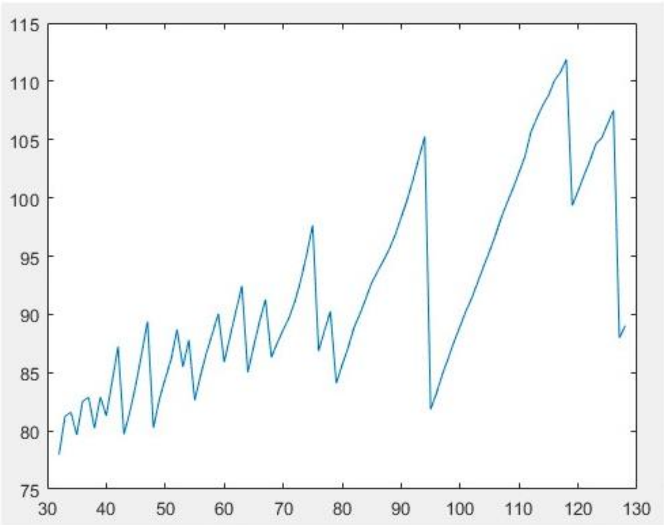


Figure 8. Compressed file size corresponding to the block size

Discussion

We have established the fact that other advanced compression algorithms such as JPEG-2000's, PNG’s are much more effective as they exploit the redundancies in both spatial and frequency domain. We have noticed we can also decrease the number of regions further if we group the values based on 8-connectivity relationship in block-based segmentation algorithm. Also the bitstream can be reduced to 4 bits if we know beforehand that f our image values fall in the range of [0, 255].