Image Compression based on Non-Parametric Sampling in Noisy Environments

Kishan Narotam & Nitesh Nana — School of Electrical & Information Engineering, University of the Witwatersrand

Objectives

To develop a robust scheme for image compression involving the creation of holes, encoding, transmitting and consequently receiving, decoding and filling these holes.

- Creating holes in an image
- Encode and transmit the image
- Simulate a noisy channel in order to introduce errors
- Identifying and filling the holes appropriately.

System Design

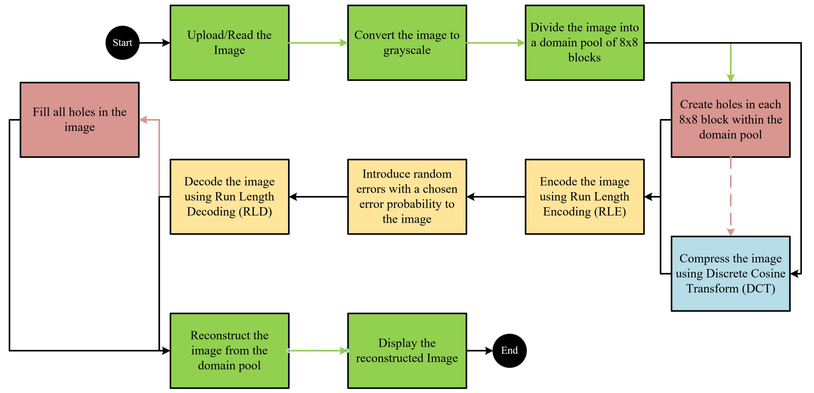


Figure 1: Image compression implementation overview

Algorithms

Algorithm 1 High-level algorithm for creating holes in `8x8` blocks

for every block in the domain pool **do**
  Go to `2x2` square in `8x8` block
  Calculate average of pixels in `2x2`

  for every pixel in the `2x2` **do**
    Check Chebyshev distance between pixels and average
  end for
  if distance between the average and each pixel < 6 **then**
    Go to `4x4` square in `8x8` block
    Calculate average of pixels of `4x4`

    for every pixel in the `4x4` **do**
      Check Chebyshev distance between each pixel
    end for
    if distance between the average and each pixel < 6 **then**
      Go to `6x6` square in `8x8` block
      Calculate average of pixels of `6x6`

      for every pixel in the `6x6` **do**
        Check Chebyshev distance between each pixel and average value
      end for
      if distance between average and each pixel < 6 **then**
        Create hole in `6x6`
      else
        Create hole in `4x4`
      end if

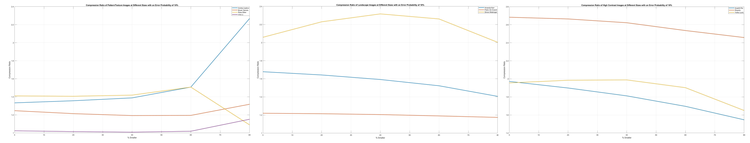
    else
      Create hole in `2x2`
    end if
  else
    Move to next block in domain pool
  end if
end for

Results: Processing

In the figures below, the various steps and processes that the image goes through can be seen.

Results: Simulation

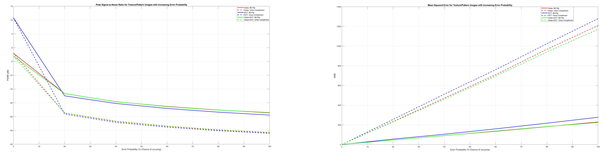
Figure 2 and the subsequent sub-figures test the created *Holes* algorithm on different images. Pattern images are chosen for the repetitive colour and smooth features; landscape images are chosen for the texture features and intense detail; high contrast images are chosen for its combination of both repetitive colour, smooth and texture features.



(a) Pattern Images **(b)** Landscape Images **(c)** High Contrast Images

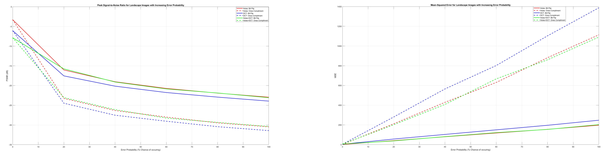
Figure 2: Compression Ratio vs Same Image at Different Sizes

An error analysis is carried out on the different image types calculating the peak signal-to-noise ratio (PSNR) and mean squared error (MSE) of the reconstructed images. The PSNR is a dimensionless number expressed on a logarithmic decibel scale, to identify the perceived errors noticeable by the human vision. The MSE is the cumulative squared error of the compressed image against the original image [1].



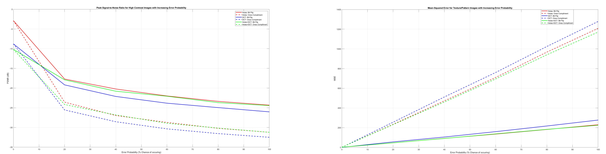
(a) PSNR for all Pattern Images **(b)** MSE for all Pattern Images

Figure 3: Pattern Images Error Analysis



(a) PSNR for all Landscape Images **(b)** MSE for all Landscape Images

Figure 4: Landscape Images Error Analysis



(a) PSNR for all High Contrast Images **(b)** MSE for all High Contrast Images

Figure 5: High Contrast Images Error Analysis

Results: Processing

Formulas

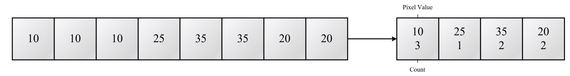
The Discrete Cosine Transform Formula

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (1)$$

The Chebyshev Distance Formula

$$D(p,q) = \max_i(|p_i - 1_i|) \quad (2)$$

Run Length Encoding



Compression Ratio

$$CR = \frac{No.\ of\ bits\ in\ uncompressed\ image}{No.\ of\ bits\ transmitted\ after\ encoding} \quad (3)$$

Future Work

The designed algorithm for image compression can be improved in numerous ways, including:

- Utilizing parallel computing and programming to speed up the processing time of the algorithm
- A neural network can be trained on multiple images so that holes can be created in the larger picture as opposed to smaller 8x8 blocks within the image
- A neural network trained on multiple images at different compression depths can determine the correct check value
- A trained neural network can ultimately reconstruct an image and improve detail and quality of low-quality images.

Conclusion

This project is more research based, and is proof of concept that multiple compression types can be utilized together for image compression. The

Acknowledgements

The pattern, landscape and high contrast images are taken from `Unsplash.com`, and the images used in this poster are by Andrej Lisakov and Pietro De Grandi.
The engineers would like to thank Professor Fambirai Takawira for his assistance as supervisor for this project.

References

[1] Uthayakumar, J. et al; *A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications*; Journal of King Saud University - Computer and Information Sciences (2018)