# GS-EECS6354M PROJECT REPORT
# A CODING SCHEME FOR LOSSLESS COMPRESSION USING PRIME FACTORIZATION AND AUXILIARY TREE STRUCTURE

*Yunge Hao*

York University
Department of Electrical Engineering and Computer Science

**ABSTRACT**

A new coding technique for lossless image compression based on prime factorization is proposed. This approach does not have transform step, but it takes advantages in an auxiliary tree structure. Tests for compression ratio are done on three gray scale images, barbara, lena and baboon. Results are compared with JPEG and JPEG2000.

## 1. INTRODUCTION

Image compression is critical for space and time efficiency of image storage and transmission. Both lossy and lossless compression techniques have been widely developed in many types of applications. Lossless compression meets the demand for high quality of reconstructed image, as in many medical diagnostic applications and astronomical surveys.

In this project I make use of some properties in number theory to devise a new coding technique. The idea is straightforward, by the unique prime factorization theorem, also known as the fundamental theorem of arithmetic, any integer greater than 1 can be uniquely represented by its prime factorization sequence. Therefore, a large number can be represented by the primes that consists of its factorization, plus the power for each prime. This way, the coding redundancy for a number represented in binary can be reduced. A grayscale image is a matrix of pixels with integer values between 0 and 255, where the prime factorization based compression scheme is naturally applicable.

Section 2 discusses some related work. section 3 presents the methodology, it consists of several preparatory compression models and then the final scheme they lead to. Section 4 includes the testing results of compression ratio. Section 5 draws some conclusions and future work.

## 2. RELATED WORK

Lossless JPEG2000 and JPEG-LS are comparably the state-of-the-art lossless compression algorithms [1]. While JPEG-LS came earlier as a standardization effort for lossless and near-lossless compression, it motivated the start of the JPEG2000 project thanks to the rich feature set provided by a voted-out algorithm CREW submitted to JPEG-LS by Ricoh [1]. The feature set of JPEG2000 includes superior performance at low bit-rate, continuous tone and bi-level compression, robustness to bit-errors, and also lossy and lossless compression, et cetera, and many further improvements have been made into JPEG2000 [1].

Lin proposed a new compression scheme to improve the compression efficiency in [2]. In traditional lossless compression, the original image is quantized in the first step. The scheme proposed in [2] differs from the tradition in that the quantization is performed in the transformed domain rather than the signal domain. The quantized coefficients are then losslessly coded. In the reconstruction part, again, the quantized coefficients are decoded. The sense of lossless applies to the quantized version, not the original image, and it is the same as traditional way, but since representing in transformed domain is sparser, the proposed compression efficiency is theoretically higher. Lin et al. also observed that at low bitrates, noise can be eliminated due to its low coefficients, which leads to desirable efficiency; while at high bitrates it is not the case. To overcome this deficiency , they also modified run-length based entropy coding to combine with the novel compression scheme. Their experimental result shows that bit savings is higher than the state-of-the-art scheme, maximum of $27.2\%$.

Focused on visual quality and efficiency, Rahman et al. proposed a lossless compression scheme using JPEG2000 with adaptive threshold [3]. They evaluated their compression results on three images, Lena image, Barbara image and Baboon image, a comparison between JPEG, JPEG2000 and their proposed method is made. Their method achieved a compression ratio of 13.90 for Lena image, 31.99 for Barbara image and 36.35 for Baboon image.

But the above [2] and [3] lossless compression schemes still

employ a lossy algorithm underneath, and the notion of lossless is relative. For applications that do not tolerate tiny loss, the scheme may be unsuitable.

## 3. METHODOLOGY

There are two components in an image compression system, an encoder that performs compression and a decoder that performs decompression. In an encoder, the input image first goes through a mapper function that is designed to reduce spatial and temporal redundancy in the transform domain, then the quantizer, and finally the symbol encoder; while the decoder is the reverse process [4].

The coding scheme proposed in this project only deals with the third step, symbol encoding of intensities directly. Since the encoding is a process by each intensity, i.e., number by number, and each number is combined with different primes, this coding is variable-length.

### A. Compression Model

#### *A.1. A Naïve Representation of Factorization*

Although the primes factored from an integer are by themselves more bit-saving, simply creating a sequence of these primes does no good in terms of compression. Quite often it needs equal or even more storage if summing up the number of bits.

| | | | | | |
|---|---|---|---|---|---|
| 2 | 2 | | | | |
| 4 | 2 | 2 | | | |
| 16 | 2 | 2 | 2 | 2 | |
| 20 | 2 | 2 | 5 | | |
| 32 | 2 | 2 | 2 | 2 | 2 |
| 40 | 2 | 2 | 2 | 5 | |

**Table 1:** A list of integers and their naïve representation sequence as factorization. The bit length of each intensity is 2, 4, 8, 7, 10, 9. In a $256 \times 256 \times 8$-bit image, the size of the representation often exists its original digit even for small values.

#### *A.2. Primes and Powers*

A major storage cost in the previous representation comes from the duplication of primes within a factorization. Such redundancy can be reduced by using the power of each prime factor. But as it turns out, the improvement in this way is negligible. It works fine for some selectively small integers, that is, coding redundancy is reduced, but for large integers, bit-length still easily exceeds 8.

| intensity | factorization | prime$_2$ | power$_2$ | bit-length |
|---|---|---|---|---|
| 2 | $2^1$ | 10 | 1 | 3 |
| 3 | $3^1$ | 11 | 1 | 3 |
| 4 | $2^2$ | 10 | 10 | 4 |
| 20 | $2^2 5^1$ | 10,101 | 10,1 | 8 |
| 120 | $2^3 3^1 5^1$ | 10,11,101 | 11,1,1 | 11 |
| 200 | $2^3 5^2$ | 10,101 | 11,10 | 9 |

**Table 2:** Intensities represented as combination of primes and powers of its factorization in binary.

#### *A.3. Indexed Primes and Powers*

To avoid such an awkward situation, notice that when factorizing integers, a multiple of prime factors rarely happen. The number of unique prime factors for an integer is small. A few unique prime factors can cover a large range of integers. Therefore, instead of coding intensities as their prime factors, coding intensities as indices of their prime factors is more compact.

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| Prime | 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 |

**Table 3:** Some examples of primes and their indices.

Compared with table 3, the bit-lengths in table 4 are reduced. Although for 120 the bit-length is 8, the same as the original source, my goal for the compression is for the overall result. As long as the size for the entire image is shrunken after encoding, some occasional violation of the shrinkage can be tolerated.

| intensities | prime index$_2$ | power$_2$ | bit-length |
|---|---|---|---|
| 2 | 0 | 1 | 2 |
| 3 | 1 | 1 | 2 |
| 4 | 0 | 10 | 3 |
| 20 | 0,10 | 10,1 | 6 |
| 120 | 0,1,10 | 11,1,1 | 8 |
| 200 | 0,10 | 11,10 | 7 |

**Table 4:** Intensities represented as combination of prime indices and powers of its factorization in binary. For the special cases of intensity 0 and 1, let their index be 00 and 11, respectively.

#### *A.4. Auxiliary Structure for Encoder and Decoder*

The table for looking up the index of primes does not need to be stored together with the image because all primes required to factor the intensities of an image can be generated during the encoding and decoding process. Auxiliary structure is needed for the decompression part. The encoded

bits for an intensity value could consists of several primes and several powers. Since the bit-length of each prime and power are indeterminate, they cannot be distinguished without the help of an indicator, for which in this case I choose an auxiliary tree structure.

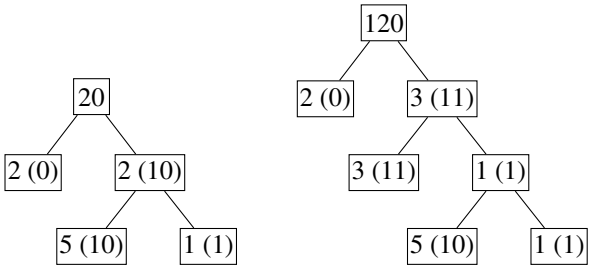


**Figure 1:** Two examples of coded intensity in a binary tree. Only binary digits in the parentheses are actually stored.

As shown in figure 1, intensity 20 and 120 are stored in a binary tree after the encoding. The tree grows to the right as the encoding proceeds, each index of the prime factored out is stored in the left node, while its corresponding power is stored in the right node, where the next prime factor is rooted. When decoding, the intensity can be restored top-down recursively.

### B. Distribution of Prime in Cumulative Factorization

Figure 2 is a plot of the histogram of primes that occur in the factorization of integers from 2 to 255. From the histogram, we can see that the cumulative number of each prime follows an approximate of negative exponential distribution. In the case of natural ordering of integers, for a given upper bound, the larger the prime, the exponentially smaller its frequency of occurrence.

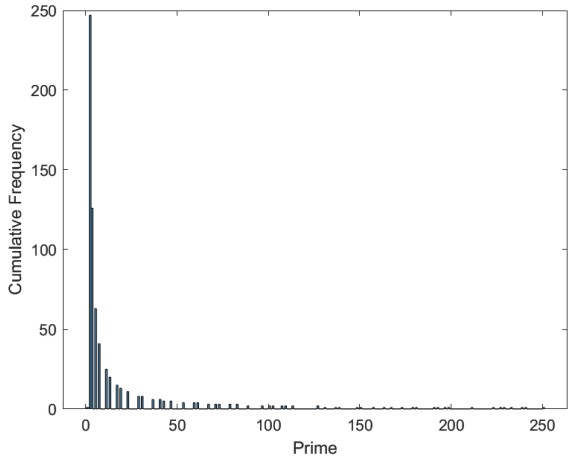


**Figure 2:** Distribution of primes in factorization of integers from 2 to 255.

This approximate negative exponential distribution of prime factor also applies to real images. Figure 3 shows a 512 $\times$ 512 8-bit image and its histogram of intensity. Comparing it with the histogram of its prime in cumulative factorization in figure 4, the latter exhibits a similar distribution as in figure 2. This can be attributed to the intensities of a image and their prime factors, after breaking down the unit that carries information, not being correlated.

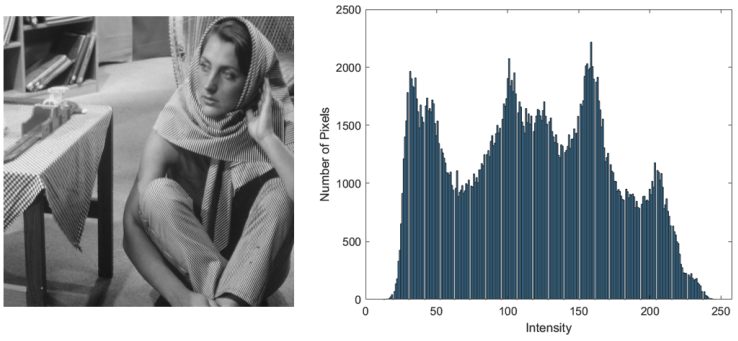


**Figure 3:** barbara_bw.png and its histogram of intensity.

Such a distribution tells that the majority of the prime factors that have a high frequency of large power is condensed to y-axis, with relatively small indices. Although large primes are inevitably contained in any factorization, and their code length may not be less than the original bit length, they do not appear often, which is the key to the proposed coding scheme. The next section has a further analysis of this point.

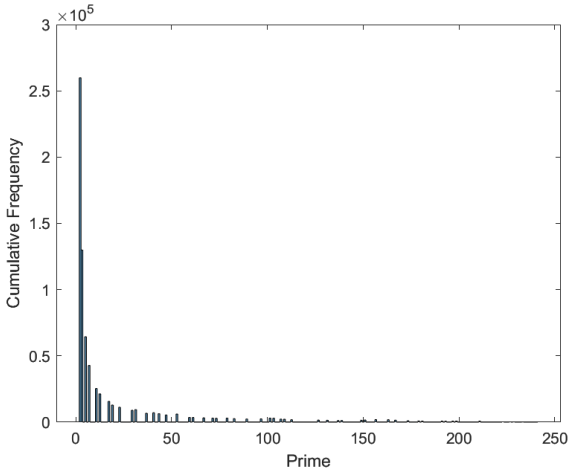


**Figure 4:** Distribution of primes in factorization of intensities of the image barbara_bw.png.

### C. Entropy Coding

According to Shannon's noiseless coding theorem [4], the number of bits per pixel that is necessarily needed to be

able to represent an image is lower bounded by the entropy of the image,

$$H = -\sum_i P_i log(P_i)$$

where $P_i$ is the probability of intensity i in the image. Hereby the entropy for image Barbara, Lena and Baboon are obtained as shown in table 5.

The length of a code word is determined by the total length of individual index and power of the prime factors. To provide an estimation of the expected code length, I assume the primes and powers are independent random variables. Then I define $E[L]$ to be the expected number of primes times the sum of the expected length of prime and power. But this is a very loose upper bound. Actual average code length, also in table 5, is much less than the estimation.

## 4. RESULTS

| image | entropy | average code length |
|---|---|---|
| Barbara | 7.6321 | 6.9492 |
| Lena | 7.4988 | 7.0051 |
| Baboon | 7.3804 | 7.1810 |

**Table 5:** Entropy and average code length of barbara_bw.png, lena_bw.png and baboon_bw.png.

To explain why the average code length is less than the entropy, I attribute it to counting the number of bits while ignoring the storage cost of the auxiliary tree structure that functions as disambiguating in the decompression.

The compression ratios for the three images using JPEG, JPEG2000 are from [3], and their adaptive threshold JPEG2000 is also included to compare with my proposed method of prime factorization. However, the results are much to my disappointment. The compression ratio of the proposed method is very low.

| Compression Technique | Size | Compression Ratio |
|---|---|---|
| Original image (Barbara) | 256KB | — |
| JPEG | 19.89KB | 12.92 |
| JPEG2000 | 9.18KB | 27.97 |
| JPEG2000 with Adaptive Threshold | 8.07KB | 31.81 |
| Proposed Method | 222.38KB | 1.15 |

**Table 6:** Comparison between proposed method, JPEG, JPEG2000 and JPEG2000 with Adaptive Threshold for image barbara_bw.png.

| Compression Technique | Size | Compression Ratio |
|---|---|---|
| Original image (Lena) | 256KB | — |
| JPEG | 18.48KB | 13.90 |
| JPEG2000 | 8.03KB | 31.99 |
| JPEG2000 with Adaptive Threshold | 7.07KB | 36.35 |
| Proposed Method | 224.16KB | 1.14 |

**Table 7:** Comparison between proposed method, JPEG, JPEG2000 and JPEG2000 with Adaptive Threshold for image lena_bw.png.

| Compression Technique | Size | Compression Ratio |
|---|---|---|
| Original image (Baboon) | 256KB | — |
| JPEG | 23.92KB | 10.74 |
| JPEG2000 | 10.71KB | 23.99 |
| JPEG2000 with Adaptive Threshold | 9.42KB | 27.26 |
| Proposed Method | 229.79KB | 1.11 |

**Table 8:** Comparison between proposed method, JPEG, JPEG2000 and JPEG2000 with Adaptive Threshold for image baboon_bw.png.

## 5. CONCLUSIONS

The novel lossless compression scheme I proposed evolves from a naïve way of representation to index-based prime factorization coding, with the help of additional structure. It achieves a low compression ratio, although for the image Lena it exceeds Huffman coding if counting number of bits only [4].

This coding scheme could be made a lossy compression scheme, where the prime factorization will not longer be done on original intensities, but on quantized values from a transform domain. This could be a future direction.

## 6. REFERENCES


[1] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer Science & Business Media, 2012, vol. 642.

[2] J. Lin, "A new perspective on improving the lossless compression efficiency for initially acquired images," *IEEE Access*, vol. 7, pp. 144 895–144 906, 2019.

[3] M. M. Rahman and M. M. Rahman, “Efficient image compression technique using jpeg2000 with adaptive threshold,” *International Journal of Image Processing (IJIP)*, vol. 9, no. 3, p. 166–174, 2015.

[4] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 4th ed. Pearson, 2018.