University of Beira Interior

Department of Informatics

2024: *Study and evaluation of EPFL_JPEG_DNA codec*

Produced by:

Alexandre Cunha Nunes Nº48267

Mentor:

Professora Doutora Maria Manuela Areias da Costa Pereira Sousa

June 2024

Acknowledgements

The conclusion of this work, as well as a significant part of my academic journey wouldn't have been possible without the guidance of my mentor, Professora Doutora Maria Manuela Areias da Costa Pereira Sousa, and as such I would like to express my eternal gratitude for her help and encouragement.

Special thanks to my colleagues and friends in *Universidade da Beira Interior* (UBI) for their camaraderie, support, and the interesting discussions and conversations that enriched my academic experience and my life. Of special note are: Artur Putyato, Tiago Silva, Miguel Marques, David Quadrado, Gabriel Lázaro,

Finally, I am profoundly grateful to my family for their unwavering support, patience, and encouragement throughout my academic journey. Their love and belief in me have been a constant source of strength and have allowed me to reach this step in my life.

To all those who have contributed, directly or indirectly, your support is deeply appreciated.

Contents

Contents ii

List of Figures iv

List of Tables v

List of Abbreviations vi

1 Introduction 1
- 1.1 Digital information storage scarcity . . . 1
- 1.2 General Objective . . . 2
- 1.3 Document Organization . . . 2

2 State of the art and concepts 4
- 2.1 Introduction . . . 4
- 2.2 Basic understanding of DNA . . . 4
- 2.3 DNA synthesis . . . 6
  - 2.3.1 Polymerase Chain Reaction . . . 7
  - 2.3.2 Artificial Gene Synthesis . . . 8
- 2.4 DNA sequencing . . . 10
  - 2.4.1 Sanger Sequencing . . . 10
  - 2.4.2 Illumina sequencing . . . 12
- 2.5 Biological DNA Constraints . . . 13
- 2.6 DNA coding . . . 14
  - 2.6.1 Encoding of data into DNA, Goldman's approach . . . 15
  - 2.6.2 PAIRCODE . . . 16
- 2.7 From Everything to Images . . . 17
- 2.8 Conclusion . . . 19

3 Study of the EPFL_JPEG_DNA codec 20
- 3.1 Introduction . . . 20
- 3.2 JPEG XL . . . 21
  - 3.2.1 Brief JPEG Explanation . . . 22
  - 3.2.2 JPEG XL lossy mode . . . 24

3.2.3 XYB Transform 27
3.2.4 Loop Filters 29
3.2.5 Variable-Size Discrete Cosine Transform 30
3.2.6 AC and DC Coefficients 32
3.2.7 AC Prediction 33
3.2.8 Adaptive Prediction 34
3.2.9 Adaptive Quantization 35
3.2.10 Entropy Coding 36
3.3 Error Correcting Coders 37
3.4 DNA 39
3.5 Conclusion 41

4 Evaluation of the EPFL_JPEG_DNA codec 43
4.1 Introduction 43
4.2 Evaluation Parameters 44
4.3 Graphical Analysis 48
4.4 Conclusion 55

5 General Conclusions and Future Work 56
5.1 Conclusions 56
5.2 Future Work 57

A JPEG DNA Dataset 58

B JPEG AI Quality Assessment Framework Example 60

Bibliography 62

List of Figures

2.1 Structure of DNA 6
2.2 The process of Polymerase Chain Reaction 8
2.3 The process of Artificial Gene Synthesis 9
2.4 The process of Sanger Sequencing 11
2.5 The process of Illumina Sequencing 13
2.6 Schematic of DNA-fragment-encoding 16

3.1 Architecture of the EPFL codec 21
3.2 Complete Architecture of JPEG XL 23
3.3 Visual representation of the JPEG compression 24
3.4 Architecture of JPEG XL lossy mode 25
3.5 Mathematical formula of DCT-II 31
3.6 Zigzag ordering pattern for an 8x8 block 34
3.7 Oligo generation 39

4.1 FSIM results 50
4.2 IW-SSIM results 50
4.3 MS-SSIM IQA results 50
4.4 MS-SSIM PyTorch results 51
4.5 NLPD results 51
4.6 PSNR HVS results 51
4.7 VIF results 52
4.8 VMAF results 52
4.9 PSNR individual results 53
4.10 Combined PSNR YUV results 53

A.1 JPEG DNA uncompressed dataset 58

List of Tables

2.1 Base-3 to DNA coding 16

4.1 Quality parameter rates 48
4.2 Distance parameter rates 49

A.1 Summary of the characteristics of the JPEG DNA uncompressed dataset 59

B.1 JPEG AI Metrics Example Table 60

List of Abbreviations

AC Alternating Current

AGS Artificial Gene Synthesis

ANS Asymmetric Numeral Systems

BAC Binary Arithmetic Coding

CD-ROM Compact Disc Read-Only Memory

DC Direct Current

DCT Discrete Cosine Transform

ddNTP Dideoxynucleotide Triphosphate

DNA Deoxyribonucleic Acid

dNTP Deoxyribonucleotide Triphosphate

DVD Digital Video Disc

EPFL *École Polytechnique Fédérale de Lausanne*

FEC Forward Error Correction

FSIM Feature Similarity

GIF Graphics Interchange Format

GM Gradient Magnitude

HDD Hard Disk Drive

HVS Human Visual System

IW-SSIM Information Content Weighted Structural Similarity Measure

IDC International Data Corporation

IDCT Inverse DCT

IQA Image Quality Assessment

LF Low Frequency

JPEG Joint Photographic Experts Group

MSE Mean Square Error

MS-SSIM Metric Multi-Scale Structural Similarity

NGS Next Generation Sequencing

NLPD Normalized Laplacian Pyramid Distance

nttp Nucleotide Per Pixel

PC Phase Congruency

PCR Polymerase Chain Reaction

PNG Portable Network Graphics

PSNR Peak Signal-to-Noise Ratio

PSNR-HVS Peak Signal-to-Noise Ratio - Human Visual System

PSNR-U Peak Signal-to-Noise Ratio - U Component

PSNR-V Peak Signal-to-Noise Ratio - V Component

PSNR-Y Peak Signal-to-Noise Ratio - Y Component

PSNR-YUV Peal Signal-to-Noise Ratio - YUV

SSD Solid State Drive

ssDNA Single-stranded Deoxyribonucleic Acid

UBI *Universidade da Beira Interior*

VIF Visual Information Fidelity

VMAF Video Multimethod Assessment Fusion

Chapter

1

Introduction

1.1 Digital information storage scarcity

The sudden surge in digital content has resulted in a significant increase in data generation globally. Every day, 2.5 exabytes (1 EB = 10^{18} bytes of data) enter our digital landscape. Data volume is projected to exceed 180 zettabytes (1 ZB = 10^{21} bytes of data) by 2025 (Taylor and Petroc [2023]). However, current storage infrastructure can only keep a fraction of this amount. Estimates suggest that only 20% of the data will be able to be stored.

The amount of digital data generated around the world is growing at an exponential rate; if we continue to use the same storage mediums, we will soon run out of physical space to store it all. To put it into context, storing the world's digital data on Digital Video Disc (DVD) would require a stack that could reach the moon 23 times, or circle the Earth 222 times. Over 8 million data centers have been built to meet rising storage demands, which is significantly more than the 500,000 data centers built in 2012. Building and maintaining these centers require significant investments in storage systems like Hard Disk Drive (HDD), Solid State Drive (SSD), or tapes, as well as infrastructure for security, power supply, and environmental controls to ensure normal conditions. Their carbon emissions are comparable to those of the aviation industry, and their energy consumption doubles every four years. By the year 2030, data centers are expected to consume up to 10% of the global electricity supply. The data centers also have a significant environmental impact, as a single data center uses as much water as three hospitals.

E-waste, used and thrown away electronic equipment such as servers, storage devices and networking hardware, also present a significant challenge, with 62 million metric tons of electronic equipment discarded in the entire

world in the year 2022 UNITAR [2024] . This large amount of consumption and disposal shows the urgent need for more sustainable data storage solutions. Surprisingly, as per the information provided by the International Data Corporation (IDC), roughly 60% to 80% of this data remains inactive or rarely used, commonly referred to as "cold data". However, it still needs to be preserved for security and regulatory compliance purposes, alongside its long lasting strategic importance. Currently, this data mainly resides on magnetic media (like tapes and hard disks), solid-state media (such as flash drives), or optical media (like Compact Disc Read-Only Memory (CD-ROM)). Despite proving to be effective solutions, these storage mediums have a significant energy cost and possess a very limited lifespan. For instance, HDD drives typically last 3 to 5 years, while tape drives can last in average between 20 to 30 years. But these types of storage are very vulnerable to mechanical breakdowns, temperature changes or magnetic interference and need data migration to new storage devices every few years to ensure they're reliability.

The scientific community recently presented an intriguing solution to this problem: using Deoxyribonucleic Acid (DNA) to store digital data. In light of this approach, this report will analyse and detail one of the current codes for transforming digital information into DNA. The EPFL_JPEG_DNA codec was developed by the *École Polytechnique Fédérale de Lausanne* (EPFL) team. Their effort is centered on the idea of combining the already existing JPEG_XL framework with a Raptor coder and a DNA module, with the overall goal of transforming photos into digital representations of the same information, but in DNA.

1.2 General Objective

The objective of this report is to present a extensive study and evaluation of the EPFL codec, showcasing its methods and functionality. The EPFL codec represents a promising advancement in the field of image compression, specially with its innovative approach of using DNA as a storage medium. This codec was submitted as a possible solution to the JPEG_DNA Call for Proposal.

1.3 Document Organization

To reflect the work that was make during this projects creation, this document will be structured in the following way:

1. The first chapter – **Introduction** – presents the project and the organization that it will follow.

2. The second chapter – **State of the art** – describes the most important concepts and technologies used during the projects creation.

3. The third chapter – **Study of the EPFL_JPEG_DNA codec** – will explain the methods that the EPFL team used to create their project.

4. The fourth chapter – **Evaluation of the EPFL_JPEG_DNA codec** – will evaluate the results gained from the execution of the codec and compare them to the values of the anchor's.

5. The fifth chapter – **General conclusions and Future Work** – is meant as a reflection of the work done and possible improvements in a future setting.

Chapter

2

State of the art and concepts

2.1 Introduction

The concept of using DNA to store information is not new; Richard Feynman first proposed this innovative idea in 1959. However, it wasn't until recently that this concept sparked widespread interest, owing to the fascinating biochemical properties of DNA , which make it an excellent candidate for future data storage. In this chapter, I will attempt to explain the various technologies used in DNA coding and the methods that each employs.

2.2 Basic understanding of DNA

DNA is a molecule that transports the genetic information of a living being. It is the main hereditary material that contains the information necessary for the development and reproduction of the organism. DNA is a polymer, a material consisting of very large molecules, that is composed by two strands, these are called polynucleotides, that are connected together in a spiral and form a double helix. Each polynucleotide is made up by simpler units, the nucleotides, and these simple units are composed of a sugar (called deoxyribose), a phosphate (which together with the sugar serves as the structural backbone of the polynucleotide strand) and nucleobase. There exist four nucleobases, these being Adenine [A], Cytosine [C], Guanine[G] and Thymine[T]. These nucleotides are connected together in a chain that is possible due to the bonding of a sugar of one nucleotide and the phosphate of the next one, and when together they are called base pair. The bonding of the nucleobases follows a strict pairing rule, cytosine always pairs with guanine and adenine

always pairs with thymine (CG - AT). See Figure 2.1 for a visual example of the structure of DNA.

DNA has bio-chemical properties that make it a innovative and very promising solution to our current storage problems, some of these attributes being:

- DNA has an extremely long lifespan, retaining data for several centuries even in harsh environmental conditions. The discovery of readable DNA records dating back 1.6 million years highlights its remarkable durability Callaway [2021].
- The process of in-vitro DNA replication is very efficient, and cost effective when compared to more traditional methods such as tape drive and HDD. These mediums often face bandwidth limitations, resulting in large processing times spanning hours or even days copying large-scale archives.
- The structure of DNA is incredible dense, with the capacity of 1 Exabyte per cubic millimeter. In comparison, the tape, which is the main method currently used to storage "cold data", offers approximately 4.6 Gigabytes per cubic millimeter Nguyen et al. [2020], a difference in size by a factor of 200 million in favor of DNA (meaning DNA is 200 million times better in relation to the storage of information per cubic millimeter).
- As DNA is the basis of life for most, if not all, living organisms, including humans, there will always be a need for the synthesization and sequencing of DNA molecules, which means it will always be a relevant area of study, regardless of the level of technology currently used in our society, be it present or future.
- DNA is a biological material that is always biodegradable, so discarding it poses no threat to the environment.
- The production of DNA is environmentally friendly because it does not use heavy metals that can harm the environment. Furthermore, DNA can be easily stored without the need for strict environmental controls, reducing the overall electronic requirements of the process.

Despite these promising characteristics, the main barrier to mainstream adoption of DNA-based data storage is the high cost and time-consuming biochemical processes involved in writing and reading DNA. However, rapid

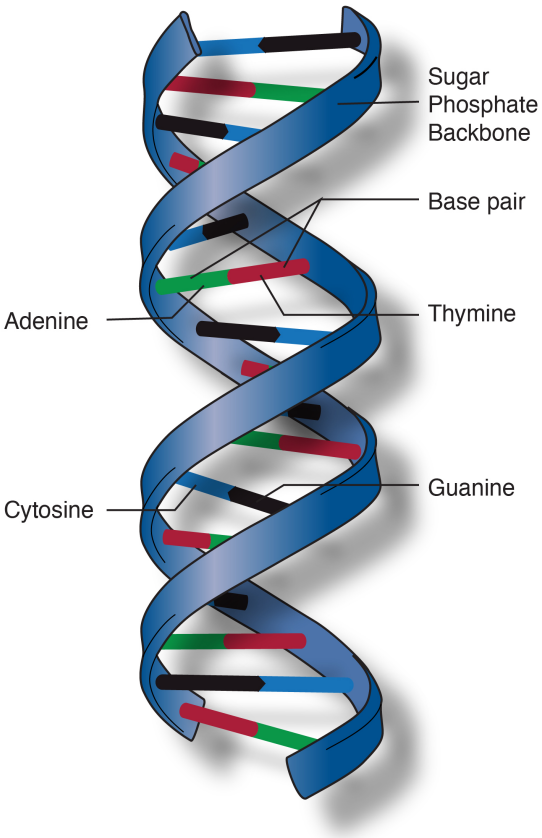


Figure 2.1: Structure of DNA NHGRI [2023]

technological advancements in this field point to a bright future for DNA based archival systems.

2.3 DNA synthesis

DNA synthesis is a important process that has a large amount of uses in a laboratory and so this process has been considered as incredibly important in the realm of research and development, in the interest of developing more productive and cost-effective methods of synthesising DNA. For this project, I will explain two of the most commonly used methods:

- **Polymerase Chain Reaction**;

- **Artificial Gene Synthesis**;

Both of these techniques are important in the field of DNA manipulation, but they diverge in how they work and where they should be used. Polymerase Chain Reaction (PCR) enables the amplification of specific DNA segments, which facilitate their analysis and detection, while Artificial Gene Synthesis (AGS) allows for the custom creation of DNA sequences, with the added benefit of more control over the genetic material.

2.3.1 Polymerase Chain Reaction

PCR is a technique that is used to amplify a specific part of the DNA molecule, making millions of copies of a target sequence. The process involves three main steps: **denaturation**, **annealing**, and **extension**. In the first step, the DNA strands are separated by the heating of the sample, resulting in single-strand templates. Then, short DNA sequences ,called primers, bind to complementary regions flanking the targeted sequence. In the last step, a DNA polymerase enzyme, which is a type of molecular builder that aids in the formation of new DNA strands by adding genetic building blocks that are known as nucleotides to a growing DNA chain, is in charge of synthesizing the new DNA strand using the primers as starting points, resulting in copies of the sequence chosen as the target. This cycle is repeated multiple times, exponentially increasing the amount of DNA fragments. See 2.2 for a visual guide of the PCR process.

This method has a large amount of benefits, such as:

- **Speed**. Rapid results due to the production of a large amount of copies within a small time frame, with the most common time frame being a few hours.

- **High Sensitivity**. PCR can amplify even tiny amounts of DNA, making it sensitive enough to detect small target sequences, which makes it invaluable to the diagnostics are of research.

- **Specificity**. With the use of specific primers, PCR can target and amplify only the desired DNA sequence, decreasing significantly the amount of noise and false positives.

- **Versatility**. PCR can be adapted for a large amount of different uses, such as genotyping, gene expression analysis, DNA sequencing and detection of pathogens, which makes it a versatile tool in molecular biology.

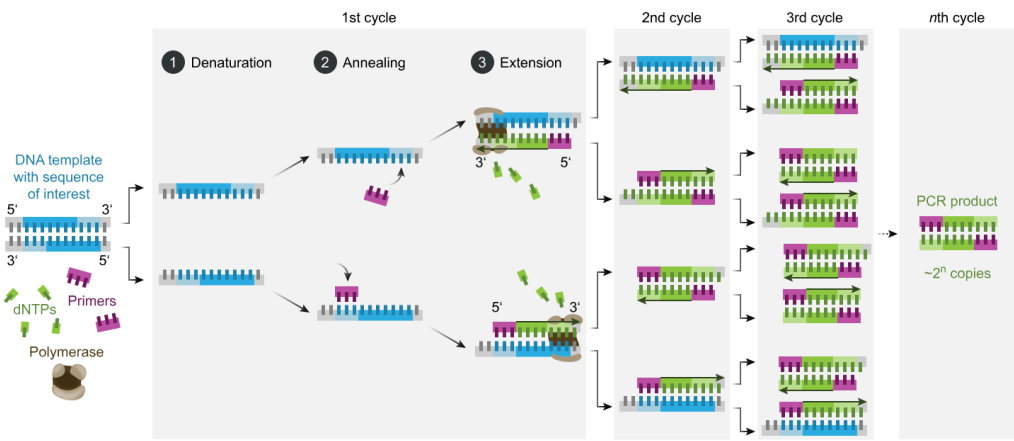


Figure 2.2: The process of Polymerase Chain Reaction Enzoklop [2020]

However, PCR also presents several challenges. Its high sensitivity makes it susceptible to contamination, which can compromise the accuracy of results. Moreover, PCR usage entails substantial costs and complexity primarily due to the expensive equipment required. Another significant issue is the limitation in the length of amplification. PCR is typically constrained to amplifying DNA fragments of up to a few thousand base pairs per reaction, imposing constraints on the size of sequences that can be effectively amplified.

2.3.2 Artificial Gene Synthesis

The technique known as AGS allows for the production of customised DNA sequences in a laboratory without the need for pre-existing templates. This technique enables scientists to create and construct DNA molecules with sequences customised for particular purposes or needs.

The procedure of AGS generally entails the use of solid-phase DNA synthesis methodologies. The following is an explanation of how these techniques operate: Researchers begin with a solid support, an object used to serve as the physical foundation, to which they affix the initial nucleotide of the desired DNA sequence. Subsequently, nucleotides are successively incorporated, with each one being joined to the elongating DNA strand, which remains firmly attached to the substrate. The addition of these nucleotides is carefully regulated, following the predetermined sequence of the DNA molecule. By means of a sequence of chemical reactions, every nucleotide is accurately integrated into the expanding chain, enabling the desired DNA sequence to gradually form, one base at a time. Through meticulous coordination of the assembly process, scientists are able to create DNA sequences with

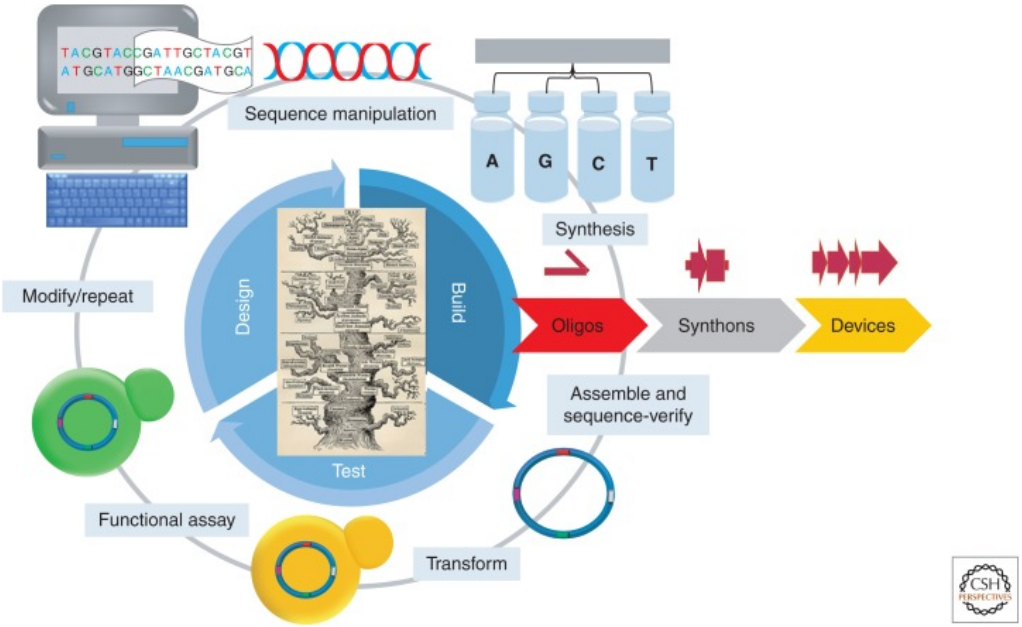


Figure 2.3: The process of Artificial Gene Synthesis Hughes and Ellington [2017]

incredible accuracy, guaranteeing that each nucleotide is correctly placed according to the intended sequence. This degree of manipulation allows for the production of artificial DNA molecules with precise characteristics and capabilities designed for a specific purpose. Refer to Figure 2.3 for enhanced comprehension.

The use of this method brings benefits such as:

- **Customization**. This method allows the design and synthesis of DNA sequences with precise specifications tailored to specific needs.
- **Rapid Development**. AGS accelerates the process of DNA generation in comparison to traditional cloning methods by bypassing the need for DNA templates, reducing the overall time required for the process.
- **Error Correction**. Synthetic DNA sequences can be designed with optimized sequences to minimize errors and enhance efficiency. This ability to fine-tune DNA improves the reliability of experimental results in various biological applications.

The main negative aspect is its complexity, as designing synthetic DNA sequences is a difficult task that requires a high degree of knowledge in the areas of molecular biology and bio informatics.

2.4 DNA sequencing

The determination of the nucleotide sequence of a DNA molecule is a process commonly known as DNA sequencing. This process has become vital to modern biological research as the knowledge of DNA sequences have allowed the scientific community to expand and develop numerous applied fields such as medical diagnosis, forensic biologic and others. Taking medical diagnosis as a example, DNA sequencing allows doctors to compare healthy and mutated sequences of DNA to diagnose different diseases such as cancer, and can allow for more individualized medical care thought the analysis of the patients DNA sequence. There exist a large amount of sequencing techniques, but for the purposes of this project, I will only explain two of the most important ones, these being Sanger and Illumina.

2.4.1 Sanger Sequencing

Sanger sequencing, pioneered by Sanger and colleagues in 1977 Sanger and Coulson [1975], revolutionized DNA sequencing and remained the gold standard for nearly four decades. Despite its declining use, it still holds value in smaller-scale projects, validation of deep sequencing results, and public health efforts such as viral outbreak surveillance, including SARS-CoV-2.

This method involves several key steps, which can also be seen in the image 2.4 with the correspondent numeration:

1. In this step we denature the double-stranded DNA molecule that contains the sequence that is being targeted. To denature the molecule, we heat it to separate the two complementary strands that result in two single-stranded DNA molecules.

2. Next, a short DNA primer that specifically binds to one end of the sequence is added, this primer provides a starting point for the next procedures.

3. Four separate reaction mixtures are prepared, each containing a specific DNA polymerase solution and all four types of Deoxyribonucleotide Triphosphate (dNTP). Additionally, each reaction mixture contains only one type of modified nucleotide called a Dideoxynucleotide Triphosphate (ddNTP).

4. The process of DNA synthesis begins as DNA polymerase catalyzes the addition of nucleotides to the growing DNA strand, gradually extending it. This elongation continues until, at random intervals, a termination

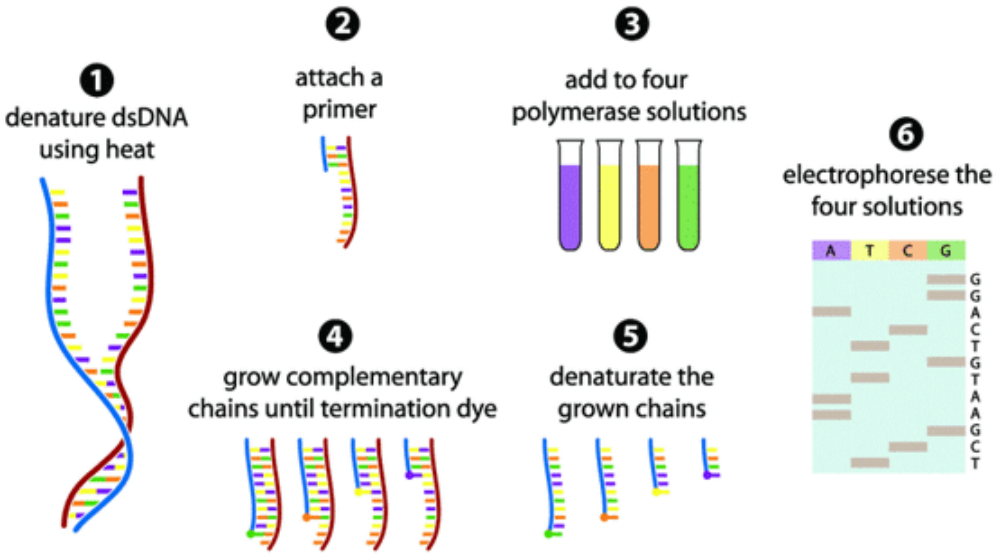


Figure 2.4: The process of Sanger Sequencing Gauthier [2008]

nucleotide, also known as a chain-terminating nucleotide, is incorporated into the growing chain.

5. The resulting DNA fragments undergo denaturation, where the double-stranded DNA molecules are separated into Single-stranded Deoxyribonucleic Acid (ssDNA) molecules.

6. Following denaturation, the individual DNA fragments are separated using gel electrophoresis, a technique that separates molecules based on size. Subsequently, the sequence of the DNA fragments is determined.

However, Sanger sequencing has its limitations. Its restricted read length inhibits the sequencing of long DNA strands, while challenges in resolving repetitive sequences introduce ambiguity. These drawbacks, coupled with the advent of more advanced sequencing technologies, have led to its decline in favour of next-generation sequencing methods. Despite these limitations, Sanger sequencing remains invaluable for targeted sequencing and validating variants detected by next-generation sequencing.

2.4.2 Illumina sequencing

Illumina Dye Sequencing, or sequencing by synthesis, is one of the most used method in Next Generation Sequencing (NGS) platforms that is being devolved by the company Illumina. This method enables the parallel sequencing of millions of DNA fragments, which allows for the rapid and cost-effective sequencing of a genome. It was developed upon the principles of Sanger sequencing but through the use of some key modifications allows for the previously mentioned parallel sequencing.

This method works in three basic steps: Amplify, Sequence and Analyse. Which can be seen in 2.5.

- **Amplification**: The process starts with purified DNA, which is fragmented and then modified with adapters containing reference segments. These adapters provide reference points for amplification, sequencing, and analysis. The modified DNA is loaded onto a flow cell with nanowells spaced apart to prevent overcrowding.

- **Sequence**: Cluster generation occurs within the flow cell, where each fragment of DNA undergoes bridge amplification PCR, producing about a thousand copies. Primers and modified nucleotides, with reversible fluorescent blockers, are washed onto the chip. These nucleotides allow the DNA polymerase to add one nucleotide at a time onto the DNA fragment.

- **Analyze**: After each round of synthesis, a camera captures an image of the chip. Using the wavelength of the fluorescent tag, a computer determines which base was added and records it for every spot on the chip. Non-incorporated molecules are washed away, and a chemical deblocking step removes the 3' fluorescent terminal blocking group. This process repeats until the full DNA molecule is sequenced.

Illumina sequencing enables massive parallel sequencing of thousands of locations across the genome. This high-throughput capability is a key feature of Illumina technology, allowing researchers to efficiently analyze large portions of the genome in a single experiment. As a result, Illumina sequencing has transformed genome research and clinical applications, allowing scientists and clinicians to perform comprehensive genome analyses at unprecedented speeds and accuracy.

Substitutions are among the most common errors in Illumina-sequenced data. These errors occur when one base is incorrectly identified as another

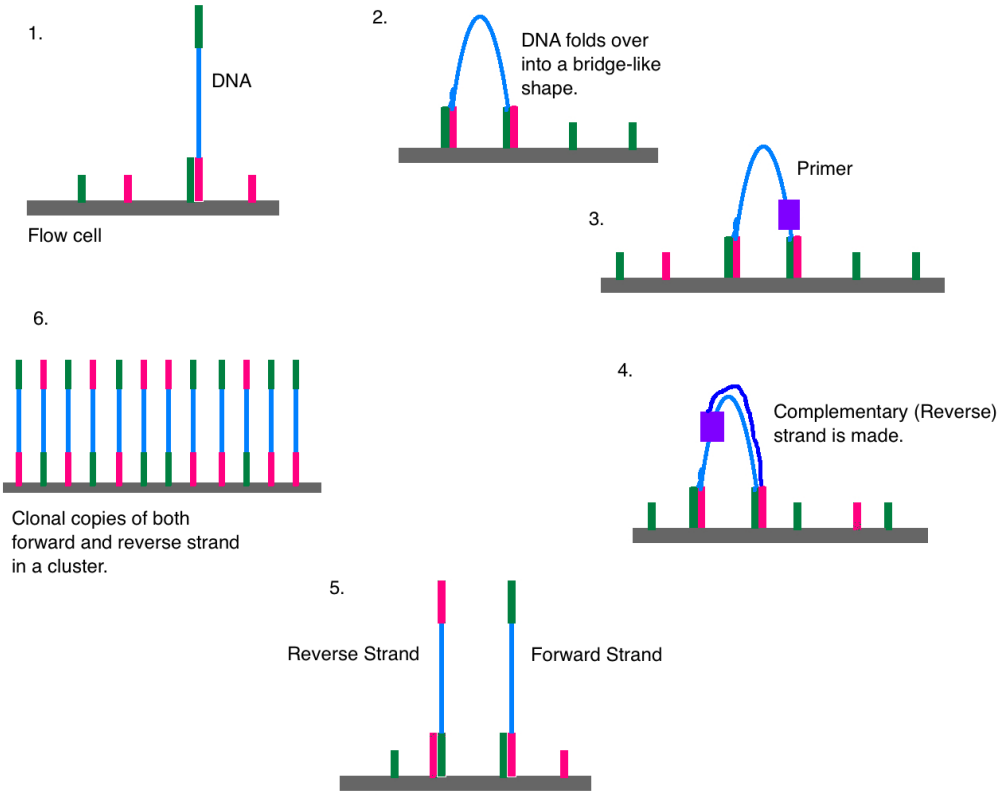


Figure 2.5: The process of Illumina Sequencing DMLapato [2015]

during the sequencing process. This is frequently caused by fluorescence similarities between certain base pairs, resulting in an incorrect interpretation of the sequencing signal. Despite the overall high accuracy of Illumina sequencing, substitution errors do occur and must be considered during data analysis and interpretation.

2.5 Biological DNA Constraints

Regardless of the approach devised by the scientific community to encode DNA, inherent limitations persist. This originates from the necessity to adhere to the fundamental principles governing DNA molecules when converting digital data into a DNA sample. These constraints, known as biological constraints, are :

- Strand length limitations: Very long strands of DNA can cause challenges during synthesis, so a good criteria is to decrease the size of the each strand to a fixed value. The value most commonly used is 200 nucleotides per strand.

- Homopolymer runs: A homopolymer run is the repetition of the same nucleotide several time in a strain, the existence of this occurrence is accepted only if in a small number. The parameters generally considered are that homopolymers of length grater than 3 should be avoided.

- CG content balance: The CG content describes the use of the based C and G in the strand, more specifically the percentage of all bases in the code that are either a G or C base. The most accepted are in between 40% and 50%.

- Repetition of patterns: A short pattern is a sequence of between 3 to 5 nucleotides that are repeated many times consecutively in a strand. The common acceptable values are that they should not be repeated more than 4 times.

2.6 DNA coding

DNA coding, the process of encoding digital data into DNA molecules, has undergone significant advancements since its inception. From pioneering research to current challenges, this field has evolved rapidly, driven by contributions from diverse scientific disciplines.

In 1994, computer scientist Leonard Adleman first proposed the idea of using DNA as a sort of computational system to solve a complex mathematical algorithm, laying the foundation for DNA computing, such that he is referenced by some as the "Father of DNA Computing" Adleman [1994].

In 2012, a study by researchers George Church, Yuan Gao, and Sriram Kosuri demonstrated the encoding of digital data into DNA sequences. It involved encoding with a simple pattern:the 0 values are mapped to A and C nucleotides, while 1 values are mapped to T and G. Their pioneering work involved the successful encoding of a 659 Kb book into synthetic DNA, showing the storage capacity of DNA as a possible medium for data storage, and it allowed for the study of biological errors in DNA storage Church et al. [2012].

In 2013, Nick Goldman, Ewan Birney and other researchers at the European Bioinformatics Institute, further advanced DNA coding by demonstrating a method for encoding digital data into DNA using a binary code Goldman et al. [2013]. The proposed method involved a encoding scheme which

converted a sequence of bytes into base-3 using a ternary Huffman tree and then encoding these trits into nucleotides using a scheme that avoids the nucleotide previously written. This was the first method of DNA encoding that respected the biological constraint of avoiding homopolymers, see 2.5 for more information. They encoded a diverse range of data, including Shakespeare's sonnets, an audio file, and a PDF document, into DNA molecules. See 2.6.1 for a more detailed explanation of how Goldman's encoding methods works.

In 2015, Robert Grass and other researchers at the Swiss Federal Institute of Technology in Zurich, proposed the first DNA encoding scheme with error correction using what are called Reed Solomon codes, and while this method will not be explained in this project, it is still a fundamental part of the DNA coding history because it allowed the scientific community to understand the need for the existence of error correction in the DNA coding process Grass et al. [2015].

In 2020, Melpomeni Dimopoulou at the *Université Côte d'Azur*, proposed a low complexity solution to construct biologically constrained DNA encoding called PAIRCODE. This work was incredibly important because it was proposed solution that met all of the biological constraints of DNA coding. See 2.6.2 for a brief explanation of how PAIRCODE works Dimopoulou [2020].

The contributions of Goldman, Goldman et al. [2013], and Dimopoulou, Dimopoulou [2020] are still important to the DNA coding study, and the following sections, 2.6.1 and 2.6.2, will present them with more detail.

2.6.1 Encoding of data into DNA, Goldman's approach

Nick Goldman's approach to encoding data into DNA involves the use of unique properties of DNA molecules with the goal of storing digital information. See Figure 2.6 for visual aid. One significant aspect of his approach is the utilization of ternary Huffman Encoding to circumvent the challenges posed by homopolymers in DNA sequences Goldman et al. [2013].

Homopolymers are repetitive sequences of the same nucleotide, such as 'AAAAA' or 'TTTTT'. These sequences can pose difficulties in accurate DNA sequencing and storage due to their repetitive nature. To address this challenge, Goldman proposed ternary Huffman Encoding, an extension of the traditional Huffman Encoding technique.

Huffman Encoding is a method of lossless data compression in which input characters are assigned variable-length codes based on their frequency. Goldman introduced a third symbol, commonly known as 'X', to represent homopolymers in DNA sequences. By incorporating 'X' into the encoding scheme, ternary Huffman Encoding efficiently represents repetitive sequences with minimal redundancy.

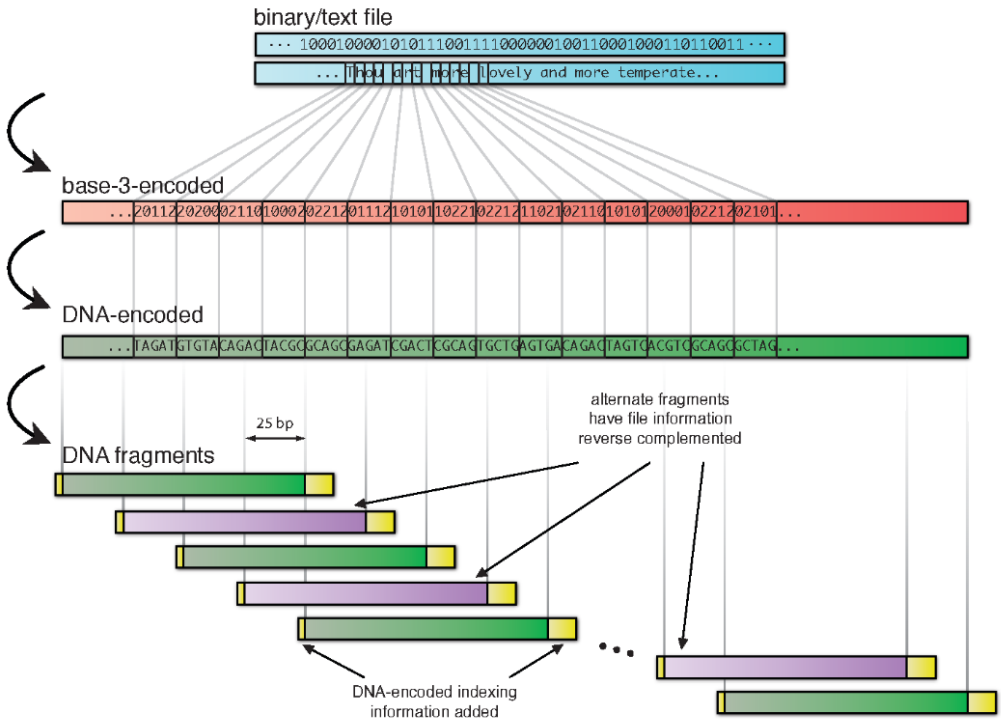


Figure 2.6: Schematic of DNA-fragment-encoding Goldman [2013]

Previous nt written	∅	1	2
A	C	G	T
C	G	T	A
G	T	A	C
T	A	C	G

Table 2.1: Base-3 to DNA encoding Goldman [2013]

This approach enables the encoding of arbitrary computer files into DNA fragments while mitigating the effects of homopolymers, thereby enhancing the reliability and efficiency of DNA-based data storage and retrieval systems. See Table 2.1 for more information on the transformation of one nucleotide to another.

2.6.2 PAIRCODE

The PAIRCODE algorithm works by using two distinct dictionaries C_1 and C_2:

$$C_1 = \{AT, AC, AG, TA, TC, TG, CA, CT, GA, GT\}$$

$$C_2 = \{A, T, C, G\}$$

These dictionaries are then used to create codewords, these are unique sequences of nucleotides that represent information. Codewords of even length are constructed by selecting $\frac{l}{2}$ elements of C_1, while codewords of odd size are made by selecting $\frac{l-1}{2}$ elements of C_1 and one element of C_2.

Melpomeni Dimopoulou's thesis Dimopoulou [2020] introduces the concept of mapping the same value to two distinct codewords, each belonging to a separate "class" - A and B. For instance, given K values, class A would include codewords from c_0 to c_K, while class B would encompass codewords from c_{K+1} to c_{2K}. This method involves assigning one codeword from each class to the same value, enhancing robustness and error correction capabilities. Through the use of this technique it is possible to obey all of the biological constraints of DNA coding, making it a very promising solution for all future developments in this area of study.

2.7 From Everything to Images

While the codification of digital information into DNA is the same no matter the end result, by this I mean that all digital information is in truth just 1's and 0's, there are reasons as to why images were chosen as the method by which DNA coding is tested. Some of these being:

High Data Density and Compression: Images, especially in formats like Joint Photographic Experts Group (JPEG), are highly compressed and contain a lot of information in a relatively small file size. This makes them ideal candidates for showcasing the high data density capabilities of DNA storage. JPEG images showcase how large amounts of data can be encoded, stored, and retrieved efficiently using DNA.

Wide Usage and Familiarity: Most people are familiar with images, which helps to make the concept of DNA data storage more relatable and understandable. Transforming a recognizable image to and from DNA demonstrates the technology's practical application and effectiveness.

Error Tolerance and Correction: Digital images can tolerate a certain level of error without significantly degrading in quality, thanks to techniques like error correction and the redundancy inherent in JPEG compression.

This makes images suitable for testing the robustness and accuracy of DNA storage systems.

Visualization of Results: The success of the storage and retrieval process can be easily visualized with images. By comparing the original image with the retrieved image, it is straightforward to assess the fidelity and accuracy of the DNA storage process. This visual proof of concept is compelling and easy to demonstrate.

Bench marking and Comparison: The use of standard images (like common test images used in image processing research) allows for bench marking and comparison with other data storage methods. Researchers can evaluate the performance, efficiency, and reliability of DNA storage by using well-known image datasets.

Recent Developments

It's important to mention that most of the ideas proposed during the earlier days of digital storage in DNA, presented solutions for the encoding of any kinda of digital data, translating from binary to quaternary representation. For the specific case of coding images, most of the works proposed compressing images with JPEG before encoding and then translating into a quaternary format.

Other papers, like Pan et al. [2019], have also proposed the concept of storing quantized images in DNA by integrating Huffman coding into the encoding and applying image processing methods to correct possible alterations in the color of the reconstructed image and improve the overall quality of the decoder process.

In 2021, authors proposed a solution for image compression based in the JPEG standard with variable length Dimopoulou et al. [2021]. The changes in the algorithm are the replacement of the two binary encoding methods with quaternary ones that respect the constraints of DNA coding, with the replacement methods being Goldman encoding and Paircode.

Later, with inspiration from the previously written solution, Secilmis et al. [2022] developed a algorithm that was capable of retrieving the coefficients from a JPEG file already compressed and encode them with a similar method.

After this Ramos et al. [2023] explored the impact that errors added during synthesis, storage and sequencing of such already compressed files, and proposing a method to protect the nucleotides created.

With a concept of a similar nature to the JPEG_DNA_BC, Pic et al. [2023] replaced the MQ-coder of the JPEG_2000 coding with an arithmetic encoder

that was designed to produce nucleotides that respected the biological constraint of homopolymer repetition.

2.8 Conclusion

In conclusion, there is a clear need for a new and superior storage method that can combat the rising need for storage. The concept of using DNA is not only novel but also very promising as a long-term solution to the problem at hand. Due to the evolution of DNA coders specific to images, the JPEG committee created a group to study the possibility of a new norm specifically for these DNA coders. The work of this group resulted in the opening of a proposal for new coders, in which one of the respondents was EPFL_JPEG_DNA codec. As such, this report aims to explain, analyze, compare, and showcase the EPFL codec, one of the emerging ideas for incorporating DNA into our digital storage.

Chapter

3

Study of the EPFL_JPEG_DNA codec

3.1 Introduction

The EPFL team is currently working on codifying images into DNA representations to showcase the promising future of digital data storage in DNA form. They have created a codec that encodes or decodes a data stream or signals, which can be called V-DNA or EPFL_JPEG_DNA, one being its technical name and the other being the name associated with their entry in the Call for Proposal.

A Call for Proposals is a formal request, normally issued by organizations and institutions with the intent of inviting individuals or groups to submit proposals for specific projects, initiatives or programs, with the intent of receiving possible solutions to the problems indicated and the possibility of funding or support to the submission that better fits the criteria initially indicated.

The JPEG_DNA Call for Proposal, was made with the scope of creating a standard for efficient coding of images that consider constraints and offers robustness to noise introduced by different stages of the storage process that is based on DNA synthetic polymers Antonini and Ebrahimi [2023].

The submission made by the EPFL team is composed by three main modules, the image compression module, handled by the state of the art JPEG_XL codec, the error correction module, implemented using a modified Raptor code and the DNA encoding module. The image 3.1 shows the operational flow of the codec, with the encoding and decoding side working practically in a similar fashion, with only some small differences.

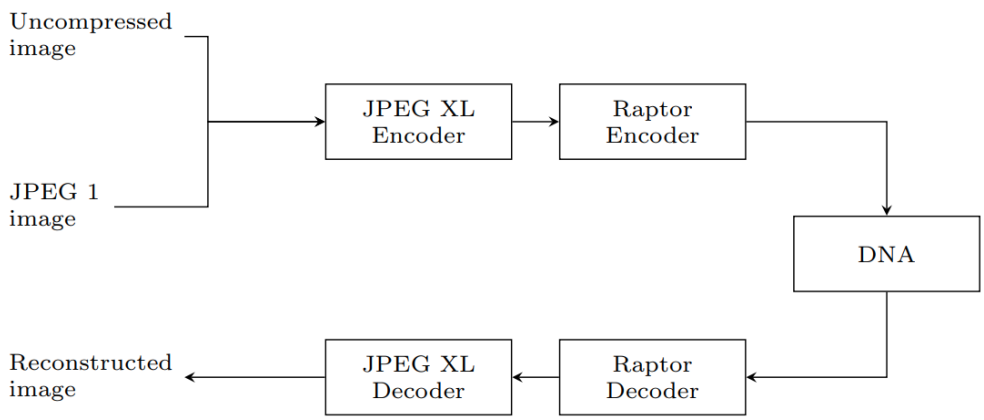


Figure 3.1: Architecture of the EPFL codec Lazzarotto et al. [2023]

This chapter is organized by three main sections, each explaining how each module works:

- **JPEG XL**: This section, 3.2, aims to explain in a detail manner how the JPEG_XL module works in the compression and handling of images.
- **Error Correcting Coders**: This section, 3.3, provides a short description and workflow of the Raptor coder in the EPFL codec.
- **DNA**: This sections, 3.4, explains the methods by which the DNA module works.

3.2 JPEG XL

JPEG_XL (the name comes a combination of JPEG, X which is the name of the standard version and L for long-term) is a modern image compression standard that builds upon the concepts of its predecessor, JPEG. It introduces significant improvements and innovations, making it a versatile and efficient solution for various image compression needs. See Figure 3.2 for a overview of the processes involved in the JPEG_XL architecture.

JPEG_XL has various compression modes, but the most distinct ones are lossy and lossless. Lossy mode attempts to reduce file size by approximating image data, which results in some loss of quality. This mode achieves higher compression ratios through techniques such as chroma sub sampling, quantization, and frequency domain transformations. It is especially useful for applications that require minimal storage space and bandwidth, such as web

images and social media, where a slight reduction in image quality is acceptable in exchange for faster loading times and lower storage costs as well as in image codification into DNA.

In contrast, lossless mode retains the original image quality, allowing the image to be perfectly reconstructed from compressed data. This mode compresses the image without losing any data by using techniques such as redundancy elimination, predictive coding, and entropy coding. While the file sizes are larger than those obtained through lossy compression, they are still smaller than the uncompressed original. Lossless mode is ideal for applications that require the highest quality, such as professional photography, medical imaging, and archival storage of high quality images.

This means that in situations where there exists a need for error tolerance and where storage efficiency is crucial, such as the EPFL codec, the lossy mode is better suited.

This report will only cover JPEG_XL *lossy mode*, as it is the mode that was chosen for the EPFL codec.

3.2.1 Brief JPEG Explanation

As this report is only focused on the codec created by the EPFL team, and as the codec uses JPEG_XL instead of JPEG, I will only briefly explain the basic steps of JPEG since JPEG_XL is based on them.

As this report only covers the codec developed by the EPFL team, and given that this codec uses JPEG_XL rather than JPEG, I will explain only the fundamental steps of JPEG, as JPEG_XL builds upon these foundations. See Figure 3.3 for a basic visual example of how JPEG works.

Color Space Conversion: The images color representation is converted from the RGB color space to the Y'CbCr color space, separating brightness (Y') from color information (Cb and Cr). Sometimes this step is skipped.

Chroma Sub sampling: The resolution of the chroma (color) data is typically reduced, as the human eye is less sensitive to fine color details than to brightness details.

Block-based Transformation: The image is divided into blocks, usually 8x8 pixels each. Each block undergoes a Discrete Cosine Transform (DCT), similar to a Fourier transform, which converts spatial data into frequency components.

Quantization: The amplitudes of the frequency components are quantized, reducing the precision of high-frequency components more than low-

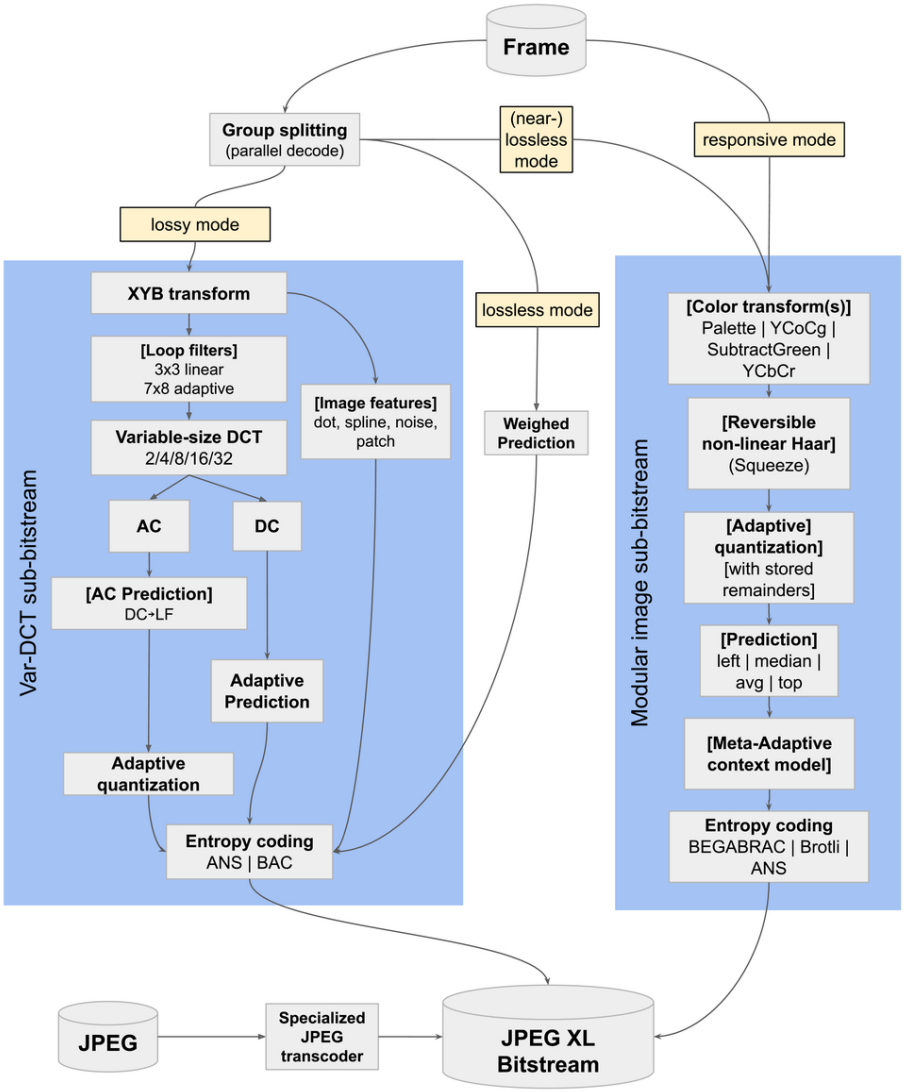


Figure 3.2: Complete Architecture of JPEG XL Alakuijala et al. [2019]

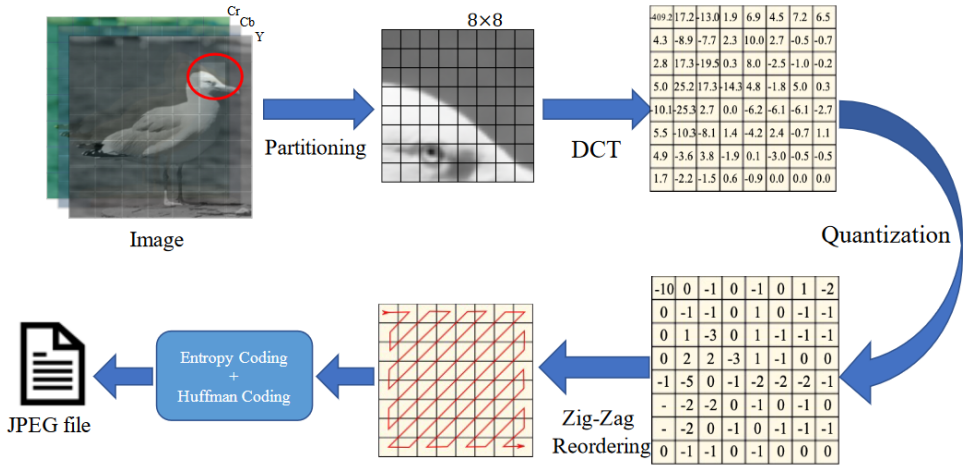


Figure 3.3: Visual representation of the JPEG compression Li et al. [2020]

frequency components. This step is where the quality setting of the encoder comes into play, as it determines the extent of quantization.

Lossless Compression: The quantized data from all blocks is further compressed using a lossless algorithm, typically a variant of Huffman encoding. This step ensures efficient storage of the compressed data.

JPEG_XL builds upon this foundation by introducing several improvements and new techniques, such as advanced quantization methods, more efficient block splitting, and better entropy encoding schemes. Additionally, JPEG_XL incorporates modern features like support for alpha channels, animation, and variable compression quality within a single image. Overall, JPEG_XL aims to provide better compression efficiency and image quality compared to traditional JPEG compression.

This may appear complicated, but in the following sections, when such terms are mentioned again, I will explain them in a more simple and accessible manner.

3.2.2 JPEG XL lossy mode

The JPEG_XL lossy mode, shown in Figure 3.4, is made up of several parts. A detailed description is provided in Jyrki Alakuijala and Wassenberg [2023]. I will list these parts below, which correspond to those shown in the Figure 3.4.

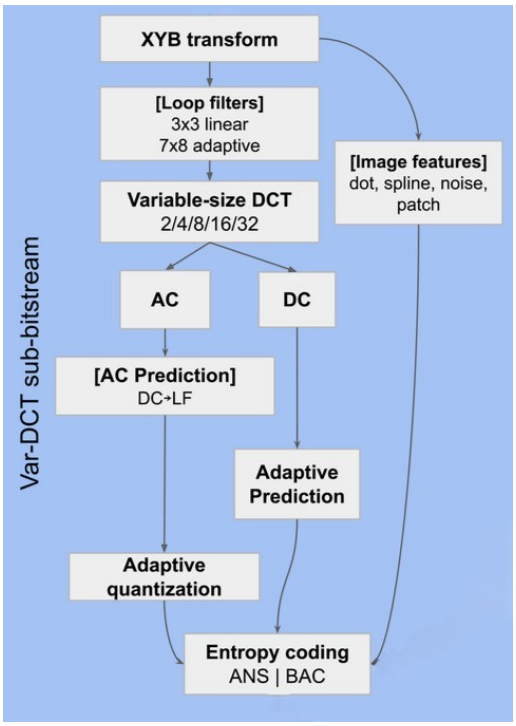


Figure 3.4: Architecture of JPEG XL lossy mode Alakuijala et al. [2019]

Following the list, each sub-section provides a more detailed explanation of each one.

1. **XYB Transform**
   - Converts color spaces to the XYB color space for better compression efficiency aligned with human vision.
   - Transforms linear RGB values to separate luminance (Y) from chrominance (X, B).
2. **Loop Filters**
   - Reduces noise and artifacts to make the image smoother.
   - Uses linear filters and adaptive filters to balance smoothing and edge preservation.
3. **Image Features**
   - Identifies and encodes specific features like dots, lines, and textures.

- Efficiently detects and encodes dots, spline features, noise, and patches.

4. **Variable-Size Discrete Cosine Transform**

   - Transforms image data into frequency components.
   - Applies DCT to image blocks of different sizes for optimal detail representation.

5. **Alternating Current (AC) and Direct Current (DC) Coefficients**

   - Separates and encodes average values (DC) and variations (AC) in image blocks.
   - Uses differential encoding for DC coefficients and zigzag scanning for AC coefficients.

6. **AC Prediction**

   - Predicts AC coefficients to reduce redundancy.
   - Uses surrounding blocks to predict current block coefficients, encoding only the differences.

7. **Adaptive Prediction**

   - Refines predictions based on local image characteristics.
   - Utilizes local statistics and machine learning approaches for improved accuracy.

8. **Adaptive Quantization**

   - Controls the precision of coefficients to balance quality and compression.
   - Adjusts quantization step sizes based on image content and perceptual importance.

9. **Entropy Coding**

   - Compresses quantized data without losing information.
   - Utilizes advanced algorithms like Asymmetric Numeral Systems (ANS) and Binary Arithmetic Coding (BAC) for efficient encoding.

3.2.3 XYB Transform

The XYB transform used in JPEG_XL is a critical step in the image compression process, designed to optimize the representation of color and brightness in a way that closely matches human visual perception.

The XYB Transformation process has various uses, such as:

- **Efficient Compression:** RGB color data frequently contains redundant information, particularly chromatic details, which are less relevant to human visual perception than luminance. The XYB transform distinguishes between luminance (important image information) and chrominance (which has less impact on perceived image quality). This split allows for more efficient compression algorithms that prioritize vital visual features while minimizing storage or transmission requirements for less critical details.

- **Alignment with Human Perception:** The XYB conversion coefficients are calculated using the human eye's sensitivity to various colors. These coefficients reflect the human eye's greater sensitivity to variations in green light, followed by red and finally blue. The XYB transform, which emphasizes luminance and appropriately scales chrominance, ensures that the compressed image remains true to what humans perceive even after significant data compression.

The XYB color space separates intensity (luminance) from color information, similar to other popular color spaces like YUV or YCbCr. In XYB color space, X and Y represent chrominance components while B represents intensity or brightness. This separation is key for efficient compression, as it allows JPEG_XL to prioritize important luminance details while reducing less critical chrominance information. The XYB color space is derived from the LMS color space, which is a perceptually uniform color space that closely reflects human vision. LMS stands for:

- **L Component:** The L component is sensitive to long wavelengths, which are primarily associated with the red-green axis. This component plays a significant role in detecting variations in colors that are most apparent to human vision in the red and green spectrum.

- **M Component:** The M component is sensitive to medium wavelengths, corresponding to the yellow-blue axis. This component captures color variations that are critical for perceiving differences in hues within the middle spectrum of visible light.

- **S Component:** The S component is sensitive to short wavelengths, predominantly associated with luminance. This component is crucial for detecting brightness and contrast, which are vital for recognizing fine details and sharpness in images.

By leveraging the LMS color space, the XYB transform effectively aligns with the human visual system, ensuring that the resulting image compression maintains high visual fidelity while optimizing storage and transmission efficiency. This alignment with human perception allows JPEG_XL to deliver superior image quality even at lower data rates, making it a powerful tool for modern image compression needs.

EXAMPLE: Conversion from RGB to XYB Color Space

See Jyrki Alakuijala and Wassenberg [2023] for original source.

Starting with RGB: You have RGB color values in the sRGB color space, with a linear transfer curve, ranging from 0 to 1.

Bias Addition: Add a small constant value, called bias, to each color component combination with the intent of making the module more uniform:

$$\text{bias} = 0.00379307325527544933$$

Mixing the Colors: Calculate three intermediate mixed color values using linear combinations of the R, G, and B values, plus the bias:

$$L_{\text{mix}} = 0.3R + 0.622G + 0.078B + \text{bias}$$
$$M_{\text{mix}} = 0.23R + 0.692G + 0.078B + \text{bias}$$
$$S_{\text{mix}} = 0.24342268924547819R + 0.20476744424496821G + 0.55180986650955360B + \text{bias}$$

Gamma Correction: Apply a square root transformation to each mixed color value and subtract the square root of the bias:

$$L_\gamma = \sqrt{L_{\text{mix}}} - \sqrt{\text{bias}}$$
$$M_\gamma = \sqrt{M_{\text{mix}}} - \sqrt{\text{bias}}$$
$$S_\gamma = \sqrt{S_{\text{mix}}} - \sqrt{\text{bias}}$$

Calculate X, Y, B: Compute the XYB color space values using the gamma-corrected values:

$$X = \frac{L_\gamma - M_\gamma}{2}$$
$$Y = \frac{L_\gamma + M_\gamma}{2}$$
$$B = S_\gamma$$

Summary:

- Start with linear RGB values.
- Add a small bias to combinations of RGB values to get mixed values $(L_{\text{mix}}, M_{\text{mix}}, S_{\text{mix}})$.
- Apply a gamma correction using square roots to these mixed values.
- Combine the gamma-corrected values to get the final XYB values.

This transformation is used to better compress and handle colors in JPEG_XL images.

3.2.4 Loop Filters

JPEG_XL employs two primary loop filters to enhance image quality post decompression, Alakuijala et al. [2019], specifically targeting block boundaries and ringing artifacts. These filters are informally termed as:

- Smoothing Convolution Filter
- Ringing Artifact Reduction Filter

Smoothing Convolution Filter: One of the essential techniques utilized in JPEG_XL is the smoothing convolution filter. Its purpose is to diminish the noticeable block boundaries that can appear in the decompressed image. This filter operates using a linear approach with a fixed kernel size, typically 3x3 or similar, to average pixel values and smooth out transitions between blocks. However, the process of smoothing inherently reduces sharpness. To counteract this effect, the encoder implements a pre-processing sharpening filter before applying the DCT (see section 3.2.5 for an explanation of DCT) step during encoding. This sharpening stage is crucial as it helps preserve the fine details present in the original image while simultaneously reducing block artifacts.

Ringing Artifact Reduction Filter: Another significant challenge in compressed images is the presence of ringing artifacts, which manifest as spurious signals near sharp edges. JPEG_XL addresses this issue by employing an adaptive smoothing algorithm inspired by Non-Local Means Buades et al. [2005]. Unlike the fixed-size linear approach of the smoothing convolution filter, the adaptive nature of this algorithm adjusts its parameters based on the local image content. It dynamically enhances processing efficiency while retaining the texture and details of the image.

To further refine the preservation of fine details, the JPEG_XL format incorporates a quantization constraint in its decoding process. After the adaptive smoothing filter is applied, the decoder performs a re-application of the DCT step to ensure that resulting coefficients align within the expected range. This corrective measure compensates for any deviations introduced during the filtering process. Subsequently, a Inverse DCT (IDCT) step is executed to generate the final filtered output.

These loop filters in JPEG_XL represent a significant advancement in image compression technology, effectively reducing artifacts while maintaining the sharp details and texture fidelity from the original image. Their implementation showcases a balance between enhancing visual quality and optimizing compression efficiency.

3.2.5 Variable-Size Discrete Cosine Transform

The DCT serves as a foundational technique in image compression, notably employed in formats like JPEG. It operates by breaking down image data into frequency components, enabling efficient representation and compression. In practical terms, DCT processes small blocks of pixels, typically 8x8 in JPEG, transforming them into coefficients that denote the presence and intensity of different frequencies within the block.

During transformation, each pixel value undergoes normalization, where it is adjusted to be centered around zero and scaled to a standardized range. The core of the DCT involves multiplying these normalized values by cosine functions of varying frequencies. The resulting coefficients reflect how much each frequency contributes to the block's overall composition. Lower frequencies, concentrated in the top-left corner of the block, carry more significant coefficients, representing smoother and broader changes in pixel values. In contrast, higher frequencies, situated towards the bottom-right, possess smaller coefficients, indicating finer details or rapid changes.

To reduce data size, DCT coefficients undergo quantization, a process where they are divided by values from a quantization matrix. This step effectively reduces precision, especially in higher frequency components where human

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad \text{for } k = 0, \ \ldots \ N-1$$

Figure 3.5: Mathematical formula of DCT-II

perception is less sensitive to minor variations. Quantization leads to many coefficients becoming zero, which significantly aids in compression. Entropy encoding techniques, such as Huffman coding, further compress the quantized coefficients by assigning shorter codes to more frequent symbols.

During decompression, the process reverses. Entropy decoding retrieves quantized coefficients, which are then multiplied by the values from the quantization matrix to reconstruct approximate DCT coefficients. These coefficients undergo an IDCT to retrieve pixel values for each block. After re-normalization and adding the DC offset, the original pixel block is reconstructed.

JPEG_XL enhances upon the DCT approach by introducing variable-sized DCT transforms Alakuijala et al. [2019]. This innovation allows for block sizes of 8x8, 16x16, and 32x32 pixels, adapting to different characteristics within an image. Larger block sizes are advantageous in homogeneous areas, reducing entropy and preserving visual quality with fewer artifacts. Specialized transforms like 8x4, 4x8, 4x4, and 2x2 DCT further refine compression by addressing specific image attributes. For instance, the Identity transform handles pixels with distinct values from nearby ones, optimizing compression efficiency.

In managing DC coefficients across larger transform sizes, JPEG_XL ensures reversibility by computing pseudo DC coefficients for each block. This maintains accuracy during decoding, crucial for preserving image fidelity across various transformations and compression levels.

For readers interested in the mathematical details, Figure 3.5, presents the formula for DCT Type II, the specific DCT algorithm employed in this codec. This formula is also widely used in various computing and programming applications.

Overall, JPEG_XL builds upon the robust foundation of DCT by expanding its capabilities to handle diverse image content more efficiently. By integrating variable-sized DCT transforms and specialized processing techniques, JPEG_XL aims to deliver superior compression performance and image quality across a broad spectrum of applications and viewing conditions.

3.2.6 AC and DC Coefficients

DC Coefficient

The DC coefficient represents the average color or intensity of an image block. Each block in JPEG_XL has a single DC coefficient, which is the first value in the block after the DCT. It captures the low-frequency content of the image block. The DC coefficient is crucial for the reconstruction of the image as it conveys the general brightness level of the block.

In JPEG_XL, and similarly to JPEG, the DC coefficients undergo several processing steps to enhance compression. These steps, which will be detailed later in the report, include:

- **Prediction**: The DC coefficient of a block is often predicted from neighboring blocks to reduce redundancy and improve compression.
- **Quantization**: The DC coefficient is quantized, approximating its value to a smaller set of possible values, which helps in reducing the amount of data required to represent it.
- **Entropy Coding**: The quantized DC coefficients are then entropy coded (often using Huffman or arithmetic coding) to compress them further.

AC Coefficients

The AC coefficients represent the higher frequency content of the image block, capturing the details and variations within the block. For instance, for a 8x8 block there are 63 AC coefficients following the DC coefficient, thought the number of AC coefficients obviously change according to size of the block. These coefficients describe the amplitude of different cosine waves needed to reconstruct the finer details of the image block.

In JPEG_XL, AC coefficients undergo processing steps similar to those in JPEG. These steps, which will be detailed later in the report, include:

- **Quantization**: AC coefficients are quantized to reduce precision and the amount of data needed to store them. The level of quantization can be adjusted to balance between image quality and compression ratio.
- **Zigzag Ordering**: After quantization, the AC coefficients are reordered in a zigzag pattern, clustering low-frequency coefficients (which are generally larger and more significant) before high-frequency coefficients (often zero or very small after quantization).

- **Run-Length Encoding**: Given that many high-frequency AC coefficients are zero after quantization, run-length encoding efficiently represents sequences of zeroes.
- **Entropy Coding**: Finally, the AC coefficients, after quantization and run-length encoding, are entropy coded.

Zigzag Ordering has not been addressed in any of the preceding sections of this report, and as such, a brief explanation is warranted. Zigzag ordering is a method used to rearrange the AC coefficients of an image block in a specific sequence that enhances compression efficiency. In the context of JPEG_XL, this ordering starts at the top-left corner of an 8x8 block, for example, and proceeds diagonally to the bottom-right corner. By following this pattern, low-frequency coefficients, which tend to have higher magnitudes and are more likely to be non-zero, are grouped together and processed before the high-frequency coefficients, which often become zero after quantization.

This method leverages the typical distribution of DCT coefficients, where lower frequencies are more significant for image reconstruction. The reordering facilitates run-length encoding by clustering the non-zero coefficients at the beginning and the zero coefficients towards the end. Consequently, it improves the efficiency of entropy coding, resulting in better compression performance.

Figure 3.6 illustrates the zigzag ordering pattern for an 8x8 block. This pattern helps in systematically reducing the data required for image representation while maintaining the integrity of the visual information."

3.2.7 AC Prediction

In JPEG_XL, AC Prediction (DC + Low Frequency (LF)) refers to a technique used to enhance the compression efficiency of the image by predicting AC coefficients using information from DC coefficients and components of neighboring blocks.

LF components refer to the AC coefficients that correspond to the lower frequencies in the DCT transformed block. These coefficients capture gradual changes in intensity and color over larger areas of the image block. The LF components are crucial as they often contain significant visual information that influences the appearance of the image.

AC Prediction in JPEG_XL leverages both the DC coefficient and the low-frequency components from neighboring blocks to predict the AC coefficients of the current block. This method reduces the redundancy between blocks and enhances compression efficiency. The prediction process can be broken down into the following steps:

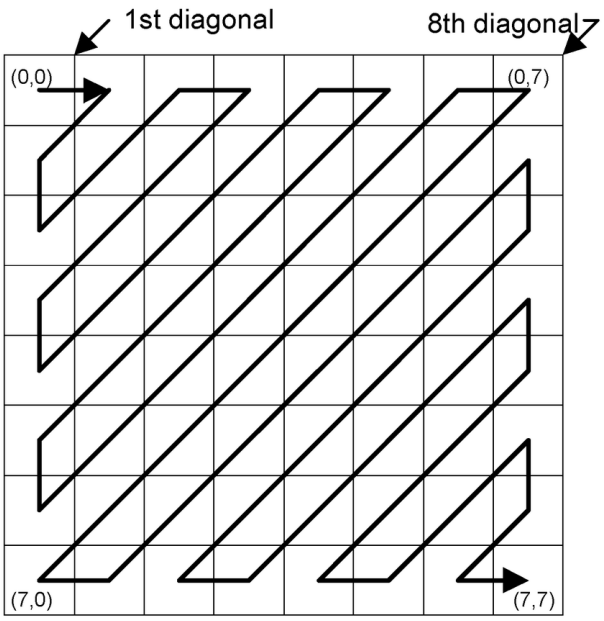


Figure 3.6: Zigzag ordering pattern for an 8x8 block Adjeroh and Lee [2004]

- **Initial Prediction:** The AC coefficients of a block are initially predicted using the DC coefficient and low-frequency components from adjacent blocks. This prediction assumes that neighboring blocks share similar visual characteristics, allowing for a more accurate estimate of the current block's AC coefficients.
- **Correction:** After the initial prediction, the predicted AC coefficients are compared with the actual AC coefficients. The difference, or residual, between the predicted and actual values is computed. This residual is typically smaller and more compressible than the original AC coefficients.
- **Entropy Coding:** The residuals are then entropy coded using advanced techniques such as ANS Entropy coding efficiently represents the residuals, further reducing the amount of data needed to store the image.

3.2.8 Adaptive Prediction

JPEG_XL incorporates an adaptive predictor to enhance image compression compared to traditional JPEG methods. This predictor takes into account the two dimensional nature of images, which improves compression efficiency and image quality.

Traditional Prediction in JPEG: Legacy JPEG compression predicts the DC coefficient (brightness information) of a block by subtracting the DC coefficient of the previous block. This method benefits from local similarities in images but is limited to one-dimensional prediction.

Bi dimensional Adaptive Prediction: JPEG_XL introduces a more advanced bi dimensional adaptive predictor. It considers nearby pixels when choosing from eight prediction modes. These modes are evaluated based on their expected performance on pixels to the left $(x-1, y)$, above $(x, y-1)$, and diagonally left-above $(x-1, y-1)$ of the current pixel. The predictor selects the mode that is anticipated to produce the least error, improving accuracy.

Error Estimation and Mode Selection: To determine the best prediction mode, the adaptive predictor calculates the maximum error for each mode based on nearby pixels. It then chooses the mode with the smallest expected error. If multiple modes are expected to produce no error, it tracks these modes to efficiently encode flat areas in images using specific compression methods tailored to predicted results.

Lossless Prediction Variation: In lossless compression mode, the predictor simplifies by using four prediction modes instead of eight. Instead of choosing a single mode, it computes a weighted average of predictions from these modes. This approach maintains image fidelity while ensuring efficient compression.

Advantages: The adaptive predictor in JPEG_XL offers several advantages:

- **Improved Compression Efficiency:** By dynamically selecting the best prediction mode based on nearby pixel evaluations, JPEG_XL achieves higher compression ratios.
- **Enhanced Image Quality:** Reduced prediction errors result in clearer and more accurate reconstructed images.
- **Adaptability:** It adjusts to different image types, handling both smooth gradients and detailed textures effectively.
- **Optimized Lossless Compression:** Even in lossless mode, the predictor maintains efficiency by averaging predictions, ensuring no loss of image data.

3.2.9 Adaptive Quantization

In traditional JPEG compression, a uniform quantization matrix is applied uniformly across the entire image for each color channel. This approach, while simple, can lead to noticeable artifacts because it doesn't adapt to the

varying levels of detail present in different parts of the image. Areas with intricate details may suffer from visible compression artifacts, while simpler areas may not utilize the available bit budget efficiently.

JPEG_XL addresses these limitations with its adaptive quantization technique. While it still uses a global quantization matrix (for each specific transform applied during compression), JPEG_XL introduces the capability to locally adjust or scale this matrix. This means that different regions of the image can undergo different levels of quantization.

For example, in regions with high detail or complex textures, JPEG_XL can apply finer quantization to preserve image details and reduce compression artifacts. Alternately, in smoother or less detailed areas, it may apply rougher quantization to save bits without significantly affecting visual quality. This adaptive approach allows JPEG_XL to allocate bits more efficiently across the image, resulting in better overall image quality and reduced perceptual errors.

Furthermore, JPEG_XL incorporates a perceptual quality metric into its encoding process. This metric helps ensure that the amount of perceived loss in image quality remains consistent across the entire image. By targeting a uniform perceptual quality level, JPEG_XL avoids drastic variations in visual fidelity between different parts of the image, thus providing a more cohesive viewing experience.

3.2.10 Entropy Coding

Entropy coding is a crucial step in image compression as it determines how data is represented using variable length codes that are based on symbol probabilities. Unlike the legacy JPEG format, which uses Huffman coding and Arithmetic Coding as entropy coding options, JPEG_XL utilizes ANS as the primary entropy coding method.

ANS is a modern technique for encoding data efficiently. It combines the best features of arithmetic coding, which is great for compressing data with complex probabilities, and Huffman coding, which is fast but less optimal for diverse symbol probabilities. ANS achieves this by using a structured approach that operates efficiently on large sets of symbols without the computational overhead of arithmetic methods Duda [2014].

BAC is another efficient entropy coding method widely used in JPEG_XL. It operates by partitioning the symbol space into intervals based on their probabilities and encoding symbols using binary representations that span these intervals. BAC is particularly effective for encoding large alphabets and achieves compression ratios similar to ANS while offering different trade-offs in terms of computational complexity and speed.

JPEG_XL integrates ANS extensively for encoding image data, employing several key strategies:

- **Single ANS Stream**: In JPEG_XL, all symbols from different probability distributions are encoded into a single stream using ANS. This approach allows for dynamic adaptation to the statistical characteristics of the image data, improving overall compression efficiency.
- **BAC**: BAC complements ANS by providing additional flexibility in encoding symbol sequences efficiently. It partitions the symbol space and encodes symbols using binary representations, tailored to specific data patterns within the image.
- **Distribution Clustering**: To manage the encoding of symbols from multiple distributions effectively, JPEG_XL groups similar distributions into clusters. Instead of encoding each distribution separately, a single histogram representing probabilities within each cluster is encoded. This reduces the amount of data needed to describe the probabilities, making the compression more efficient.
- **Variable-Precision Encoding**: JPEG_XL supports flexible encoding of distribution histograms. For simpler data patterns, a straightforward representation is used, assuming uniform probabilities. For more complex distributions, a variable precision mode allows precise control over the encoding of each probability value, optimizing compression performance.
- **Hybrid Unsigned Integer Encoding**: To further optimize ANS, JPEG_XL uses a hybrid encoding scheme for integers. Smaller values below 16 are encoded directly as symbols, while larger values use a specialized method to encode the most significant bits and leave the rest uncompressed. This reduces the number of different symbols encoded in the stream, speeding up both compression and decompression processes.

3.3 Error Correcting Coders

The EPFL codec uses, for the DNA encoding module, a implementation of a modified Raptor Coder that follows the RU10 (Raptor Unsystematic), Schwarz and Freisleben [2021] , description implemented by the NOREC4DNA software framework Shokrollahi et al. [2007].

The description of Raptor codes in this report is very simple, but that is by design, as the report's objectives do not include the study of error correction

codes. For more information on the Raptor codes see the original documents present in the references.

Raptor codes are a class of fountain codes, Shokrollahi [2011], known for their efficiency in data transmission over unreliable communication channels. Unlike traditional error correction codes, Raptor codes can generate a potentially limitless number of encoded symbols from a finite set of original data symbols. This characteristic makes them highly adaptable and robust in varying transmission conditions. Fountain codes, of which Raptor codes are a subset, are a type of Forward Error Correction (FEC) codes. They are named for their ability to generate a "fountain" of symbols.

- **Symbol Generation:** Fountain codes can produce an infinite number of encoded symbols from a finite number of original data symbols. Each encoded symbol combines the original symbols.
- **Decoding Process:** The receiver can begin decoding as soon as it receives a sufficient number of encoded symbols, which are used to recreate the original data. The exact number required typically exceeds the number of original symbols by a small margin, referred to as the overhead.
- **Adaptability:** Fountain codes are dynamic in nature, which means they can adapt to changing transmission channel conditions, such as varying levels of noise or packet loss, without having to retransmit data.

FEC is a technique used in data communication to detect and correct errors without requiring retransmission. Its use has the following benefits:

- **Redundancy Addition:** Redundant data, known as parity bits, is added to the original data before transmission. This redundancy allows the receiver to detect and correct errors.
- **Error Detection and Correction:** The receiver uses the redundant information to identify errors in the received data and correct them. This process significantly enhances the reliability of data transmission, especially in noisy or error-prone channels.

The EPFL_JPEG_DNA codec uses Raptor Codes to encode compressed image data into DNA sequences stored in FASTA files as well as adding redundancies to the generated chunks with the goal of ensuring that the data can be safely reconstructed in case of errors during transmission. See 3.7 for a visual example of Raptor code use in EPFL codec. Here's a breakdown of the process relative to the Raptor coder:

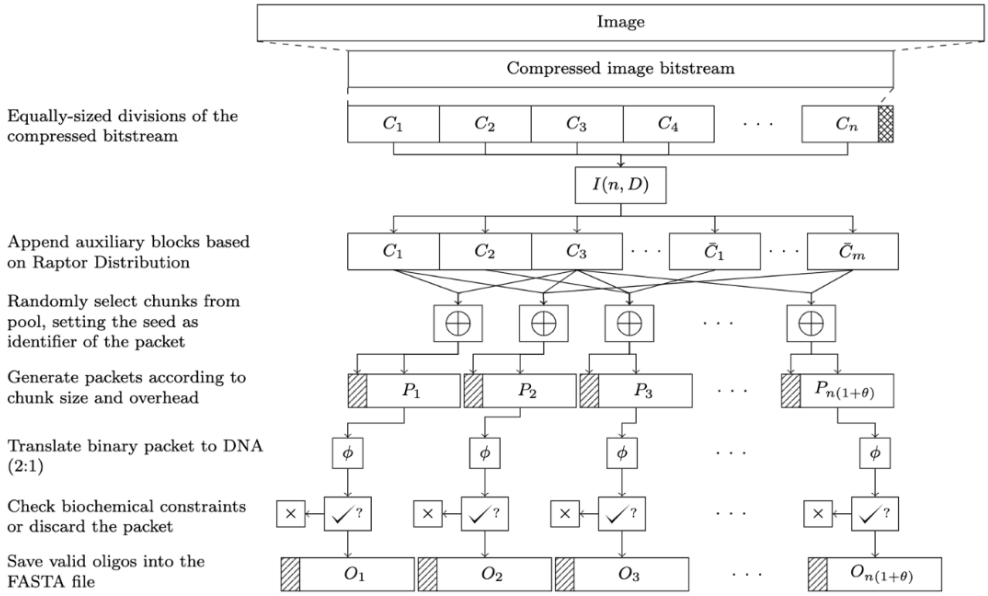


Figure 3.7: Oligo generation Lazzarotto et al. [2023]

Data Chunking: The compressed image data is meticulously divided into equally sized blocks known as chunks. This simplifies the conversion process and ensures consistent data handling.

Raptor Encoding: Raptor Codes are used to enhance fault tolerance. These codes generate additional data blocks (auxiliary blocks) based on Raptor Distribution. These auxiliary blocks add redundancy to the data, allowing for reconstruction even when some original chunks are missing or corrupted.

In essence, the EPFL_JPEG_DNA codec uses Raptor Codes to introduce redundancy into the compressed image data. This redundancy is then encoded into DNA for storage, allowing for potential recovery even if some of the DNA is damaged or lost.

3.4 DNA

In the EPFL codec, the DNA module serves the purpose of encoding and decoding digital information from and into DNA. In practice, when testing the codec, it wouldn't be realistic to always encode and decode the images into

DNA molecules, and as such there is a need for a system that simulates the biological system, something that will be the placeholder for the synthesis and sequencing processes. The information that represent the biological DNA nucleotides is stored in FASTA files. In the Figure 3.7 the steps of Packet Formation, Binary to DNA conversion, Biochemical Validation and FASTA file creation belong to this module, as seen below:

Packet Formation: A random selection of data chunks, both original and auxiliary, is made. The selection process is controlled by a unique packet identifier, which provides some predictability for reconstruction. The chosen chunks are then combined to form packets.

Binary to DNA conversion: Each packet consists of binary data (0s and 1s). This binary data is efficiently converted to a DNA sequence. To represent data efficiently, two bits are typically encoded into a single DNA base pair (A, C, G, or T). This and the following parts are handled by the DNA module.

Biochemical Validation: After conversion, the generated DNA sequence goes through an important validation step. The codec meticulously checks the sequence against established biochemical principles to ensure that it can function in a living cell. Any sequence that fails this check is removed to avoid storing invalid DNA.

FASTA file creation: Finally, the validated DNA sequences, also known as oligos, are meticulously saved as a FASTA file. FASTA is a standard file format developed specifically for storing DNA sequences. This produces a permanent record of the encoded image data in biological format.

FASTA

FASTA files are a standard text-based format for storing nucleotide sequences Pearson and Lipman [1988]. The format was first introduced in the early 1980s as part of the FASTA software package for sequence alignment. It is now commonly used in bio informatics to store sequences and their annotations. Each sequence entry in a FASTA file is preceded by a one-line description that starts with the greater-than (">") symbol. This header line is followed by the actual sequence data. The header may contain various types of information, such as the sequence name or a short description.

The sequence data is made up of a series of characters that represent the nucleotide bases (A, T, G, and C) in DNA sequences or the amino acids (via single-letter codes) in protein sequences. The sequence is usually written in

uppercase letters, but lowercase letters can also be used and may have different meanings depending on the context. Sequences can span multiple lines, but each line is typically consistent in length, such as 200 in this case.

FASTA files are adaptable and can hold multiple sequences, each with their own header and sequence data. This makes them useful for storing large datasets, such as different organisms' genomes or a database of protein sequences. The format is straightforward and easy to parse, which has contributed to its widespread use in the bio informatics community.

In the EPFL codec, FASTA files are used as the files where the biological information is kept, serving as the input to the decoding process and the output from the encoding process.

3.5 Conclusion

This chapter has explored the EPFL JPEG_DNA submission, focusing on the roles of DNA, JPEG_XL, and Raptor codes within the codec. Each element contributes in unique ways to the codecs performance, particularly in terms of image compression and data reliability.

The JPEG_XL framework has been discussed as a cutting-edge image compression standard used in the EPFL codec. By transforming images from the spatial domain to the frequency domain and using a large number of important and innovative techniques, JPEG_XL achieves excellent compression ratios while maintaining high image quality.

Raptor codes were discussed as they relate to error correction and data recovery. Raptor codes, a type of fountain code, are critical for ensuring data integrity and reliability, particularly in situations involving data transmission over noisy channels or storage in error-prone environments. Their ability to generate an infinite number of encoded symbols from the original data makes them extremely resistant to data loss and corruption.

The DNA module is crucial for representing the biological processes as well as ensuring that the results follow the biological constraints. By using the FASTA format is can represent nucleotide sequences efficiently. Its simple yet versatile structure ensures compatibility with a large number of bio informatics tools, making it essential for handling DNA based data.

The EPFL codecs use of JPEG_XL, Raptor codes and FASTA demonstrates a sophisticated approach to DNA data storage and image compression. Each component addresses a specific issue related to data encoding, compression, and transmission, thereby improving the codecs robustness and efficiency. The EPFL JPEG_DNA submission represents a significant step forward in the

field, showcasing novel solutions at the intersection of bio informatics and digital image processing.

This chapter has provided a detailed overview of these key components, emphasizing their individual and collective significance in the EPFL JPEG_DNA submission. The findings of this study highlight the codecs potential to revolutionize data storage and compression methodologies, paving the way for future research and development in this emerging field.

Chapter

4

Evaluation of the EPFL_JPEG_DNA codec

4.1 Introduction

The evaluation of the EPFL_JPEG_DNA codec presented in this chapter is based on a series of methodical transformations between file formats, employing a array of specialized software tools. This evaluation seeks to determinate the efficacy and reliability of the codec in preserving the integrity and quality of image data through multiple conversion stages. To initiate the process, the transformation from a Portable Network Graphics (PNG) image file to a JXL file was executed using **CJXL version 0.10.0**. The encoded JXL file was then converted into a FASTA file. This conversion was performed using the **"encode.py"** program provided within the codec submission. In the reverse sequence, the FASTA file was decoded back into a JXL file using the **"decode.py"** program also included in the codec submission. This crucial step tests the codecs ability to accurately reconstruct the JPEG_XL file from the DNA sequence representation, thereby verifying the codecs fidelity and robustness. Finally, the JXL file was transformed back into a PNG image file using **DJXL version 0.10.1.** This last conversion completes the full cycle of transformations, enabling a direct comparison of the original and final images to assess any potential loss in quality or data integrity. The complete sequence of transformations—PNG to JXL, JXL to FASTA, FASTA to JXL, and JXL back to PNG—provides a comprehensive framework for evaluating the performance and practicality of the EPFL_JPEG_DNA codec. Through this meticulous process, the chapter aims to provide insightful analysis on the codecs effectiveness in digital image preservation and in biological storage applications.

4.2 Evaluation Parameters

To grasp a comprehensive understanding of the EPFL codec analysis, it is crucial to delve into the fundamental parameters employed during its evaluation phase. These parameters play a important role in assessing both the quality of the images and the efficiency of the compression techniques. They form the cornerstone for comparing different codecs and assessing their performance within the context of the JPEG_DNA call for proposal. These metrics, whose explanation derives from the Antonini and Ebrahimi [2023] paper, are the following:

- The **Nucleotide Per Pixel (nttp)** metric, also written as (Nt/px), quantifies the number of nucleotides required to encode each pixel in an image. Lower nttp values indicate more efficient encoding. When comparing image quality to nttp, the x-axis represents nttp values, emphasizing the trade-off between storage efficiency and image fidelity. This metric is important in the JPEG_DNA call for proposals because it allows for the comparison of different encoding schemes, with lower nttp rates preferred for cost-effectiveness and practicality in DNA data storage.

- The **Feature Similarity (FSIM)** metric, Zhang et al. [2011], is an image quality assessment tool that evaluates images using low-level features important to human visual perception. It primarily relies on Phase Congruency (PC), which measures local structure significance, and Gradient Magnitude (GM), which captures contrast. These features complement each other in determining image quality. FSIM creates local quality maps from PC and GM, integrates them, and uses PC as a weighting function to produce a single quality score. A color version of FSIM also exists for color images. A high FSIM value indicates better image quality.

- The **Normalized Laplacian Pyramid Distance (NLPD)** metric, Laparra et al. [2016], is an image quality metric that assesses images based on local luminance subtraction and local contrast gain control, both of which are critical components of the human visual system. To assess image quality, NLPD uses a Laplacian pyramid decomposition combined with a local normalization factor. The metric is calculated in the normalized Laplacian domain, and the root mean squared error in a weight-normalized Laplacian domain is used to assess the quality of a distorted image in comparison to its reference. A lower NLPD score suggests higher image quality.

- The **Peak Signal-to-Noise Ratio (PSNR)** metric, Ponomarenko et al. [2007], is widely used for assessing the quality of reconstructed images com-

pared to their original versions. It quantifies the similarity between two images by measuring the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is expressed in decibels and is computed using the Mean Square Error (MSE) between the original and reconstructed images. A higher PSNR value indicates better image quality, as it signifies lower error and noise in the reconstructed image. The PSNR metric also has separate versions to assess different parameters, such as:

1. **Peak Signal-to-Noise Ratio - Human Visual System (PSNR-HVS)** is an enhanced version of the standard PSNR metric that incorporates models of the Human Visual System (HVS). While standard PSNR measures the fidelity of a reconstructed image by comparing it to the original using pixel-by-pixel differences, PSNR-HVS takes into account the sensitivity of the human eye to different types of distortions. This makes PSNR-HVS a more perceptually accurate measure of image quality, aligning better with human subjective assessments.

2. **Peak Signal-to-Noise Ratio - V Component (PSNR-V)** specifically measures the quality of the V (Value or Luminance) component in the YUV color space. The V component represents the brightness information in the image. PSNR-V calculates the ratio of the maximum possible signal power to the power of the noise affecting the luminance component. Higher PSNR-V values indicate that the brightness information in the image has been preserved more accurately.

3. **Peak Signal-to-Noise Ratio - U Component (PSNR-U)** measures the quality of the U (Chrominance) component in the YUV color space, which represents the blue color differences in the image. It evaluates how well the color information corresponding to the blue channel is preserved in the reconstructed image compared to the original. Like other PSNR metrics, a higher PSNR-U value indicates better preservation of the blue chrominance information.

4. **Peak Signal-to-Noise Ratio - Y Component (PSNR-Y)** is the PSNR metric applied to the Y (Luminance) component of the YUV color space, representing the gray scale information of the image. This component is critical for human visual perception as it carries the structural and intensity information of the image. PSNR-Y evaluates how accurately the luminance values are retained in the re-

constructed image. Higher PSNR-Y values correspond to better preservation of the image's luminance.

5. **Peal Signal-to-Noise Ratio - YUV (PSNR-YUV)** is the PSNR metric applied to the full YUV color space, representing the full spectrum. This metric is calculated using the individual PSNR values for the Y, U, and V components. The formula is given by:

$$\mathrm{PSNR}_{YUV} = \frac{6 \cdot \mathrm{PSNR}_Y + \mathrm{PSNR}_U + \mathrm{PSNR}_V}{8} \tag{4.1}$$

   Where:

   - PSNR_Y is the PSNR value for the Y (luminance) component.
   - PSNR_U is the PSNR value for the U (chrominance) component.
   - PSNR_V is the PSNR value for the V (chrominance) component.

- The **Metric Multi-Scale Structural Similarity (MS-SSIM)** metric, Wang et al. [2003], is a widely used image quality evaluation algorithm that compares the relative quality of reference and distorted images at various resolutions. It performs better for learning-based image codecs than single-scale methods, accommodating variations in image resolution and viewing conditions. MS-SSIM introduces an image synthesis-based approach for adjusting parameters that weight the importance of various scales. A higher MS-SSIM score indicates improved image quality. Different versions of MS-SSIM include:

  1. **MS-SSIM PyTorch** uses a PyTorch function to evaluate structural similarity between images at multiple scales. This implementation leverages PyTorch's efficient tensor operations and GPU acceleration, making it suitable for real-time applications and integration into deep learning pipelines.
  2. The **MS-SSIM Image Quality Assessment (IQA)** variant compares reference and distorted images by evaluating luminance, contrast, and structural similarities across different resolutions. This approach effectively captures perceptual degradation, providing a robust measure that aligns well with human subjective evaluations.

- **Information Content Weighted Structural Similarity Measure (IW-SSIM)** is a metric, Wang and Li [2011], that extends the structural similarity index by using information content weighted pooling. It assumes that

perceptual weights proportional to local information content should be used when viewing natural images. Advanced statistical models of natural images are employed to derive optimal weights, combined with multi scale structural similarity measures, to achieve high correlation with subjective scores from well-known databases. A higher value shows better results.

- The **Video Multimethod Assessment Fusion (VMAF)** metric, Li et al. [2023], developed by Netflix, focuses on artifacts from compression and re-scaling. It estimates quality by computing and fusing scores from multiple quality assessment algorithms using a support vector machine, a supervised learning model machine. Although designed for videos, VMAF also performs well for single images, especially in learning-based image codecs. A higher VMAF score indicates better image quality.

- The **Visual Information Fidelity (VIF)**, Sheikh and Bovik [2004], metric measures the loss of human perceived information during degrading processes like image compression. It evaluates information fidelity using natural scene statistics and is related to the Shannon mutual information between the degraded and original images. Operating in the wavelet domain, VIF has been found to correlate well with human perception, including for learning based image codecs. A higher VIF score indicates better image quality.

These parameters collectively provide a comprehensive evaluation framework for assessing the performance of the EPFL code, leveraging modern advancements in image quality assessment and compression techniques.

JPEG_XL uses various parameters for control of the encoding process. But for this report I will only mention those that are used by the codec in study, this being "**quality**" and "**distance**." The "**distance**" parameter defines quality in terms of just-noticeable differences, with lower values ensuring higher fidelity, similar to mathematically lossless compression (e.g., -d 0), and higher values producing denser files with lower fidelity. For lossy sources such as JPEG and Graphics Interchange Format (GIF), compression is set to lossless, allowing bit-for-bit reconstruction from the original JPEG. Alternatively, the "**quality**" parameter, which ranges from 100 for lossless to lower values for smaller file sizes, offers another way to specify encoding preferences. This dual-parameter system ensures that JPEG_XL images can be encoded with the desired quality or file size.

(Note: The quality and the distance parameters are mutually exclusive.)

The JPEG_DNA datasets, see appendix A for more information, are a group of uncompressed images that were compressed using an encoding workflow that aimed to achieve the target nucleotide rates specified in the call for proposals. To optimize JPEG_XL compression, only the **quality** parameter was adjusted, as the encoder's other components translate binary data to DNA at a constant rate. The Table 4.1, which was sourced from the original paper for the EPFL submission to JPEG_DNA, Lazzarotto et al. [2023], shows the corresponding **quality** values for each target rate and image.

To further evaluate the encoding workflow of the EPFL codec, I conducted additional tests using a variation of the distance parameter. These values, recommended by my mentor, excluded those greater than 1.0 as they were deemed unnecessary for testing purposes. Details of the specific values used are provided in Table 4.2. Another reason to use the distance parameter is that in the newer JPEG_XL versions, the quality parameter is being depreciated therefore is is assumed that in future versions it could be completely removed. As such, the quality parameter was included in the testing, mainly due to the fact that it was the parameter used in the response to the JPEG_DNA proposal, but the distance parameter was also included due to its use in newer codec versions.

Table 4.1: Quality parameter used for each image at each target rate Lazzarotto et al. [2023]

Image	**r1**	**r2**	**r3**	**r4**	**r5**	**r6**	**r7**
00001_1192x832	96	94	92	90	86	81	76
00002_853x945	95	94	92	90	88	84	81
00003_945x840	96	95	92	89	85	78	70
00004_2000x2496	91	88	84	79	74	65	62
00005_560x888	94	92	89	86	82	76	69
00006_2048x1536	97	96	94	92	89	85	80
00007_1600x1200	96	94	92	88	84	78	70
00008_1430x1834	98	96	94	92	88	84	78
00009_2048x1536	99	98	96	94	91	87	80
00010_2592x1946	95	94	92	90	87	83	78

4.3 Graphical Analysis

The graphics presented here originate from two distinct sources. The graphics on the right are from the paper titled "Report on Crosschecking of the

Table 4.2: Distance parameter used for each image at each target rate (source: Experimentally chosen)

Image	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
00001_1192x832	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00002_853x945	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00003_945x840	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00004_2000x2496	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00005_560x888	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00006_2048x1536	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00007_1600x1200	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00008_1430x1834	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00009_2048x1536	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
00010_2592x1946	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0

BioCoder and HiDNA Submissions to the JPEG_DNA_CfP" (Testolina et al. [2023]). This paper aimed to evaluate several submissions to the JPEG_DNA Call for Proposal, including a submission from EPFL. Conversely, the graphics on the left result from independent testing conducted by myself, with the objective of evaluating the EPFL codec and its reported results. For the purpose of this report, only the "Encoder EPFL CfP" data matters.

In the images created by my experimentation, the term "EPFL by author" refers to the images provided by the author in the codec folder. "EPFL by reproduction" denotes the images created and tested by myself using the tools provided by the codec, following the **quality** parameters specified in Table 4.1, this was the method used by the original authors of the EPFL codec when submitting their results to the Call For Proposal. Finally, "EPFL by distance" refers to the images created and tested by myself using the tools provided by the codec, adhering to the **distance** parameters outlined in Table 4.2.

The evaluation of the results, both the original images from the codec folder as well as the images created during my experimentation, were made with the tool provided by the original Call for Proposal, see Karabutov [2021] for original source and see appendix B for a example of how the metric tool returns results.

Figure 4.1: FSIM results: left - obtained from my experiment, right - reported in the referenced paper

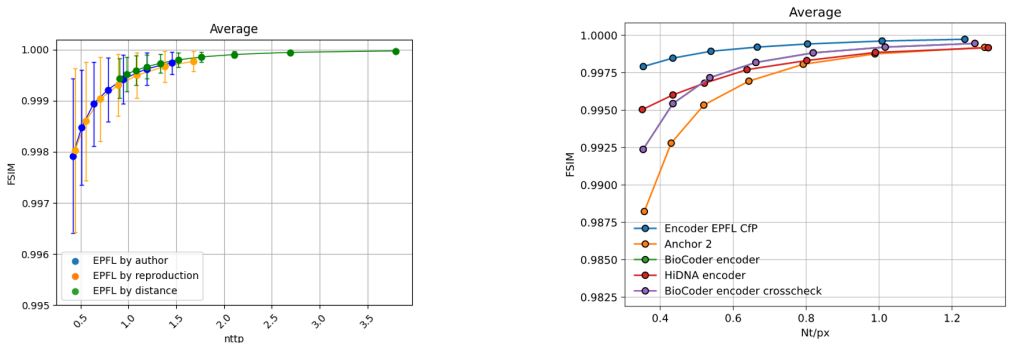


Figure 4.2: IW-SSIM results: left - obtained from my experiment, right - reported in the referenced paper

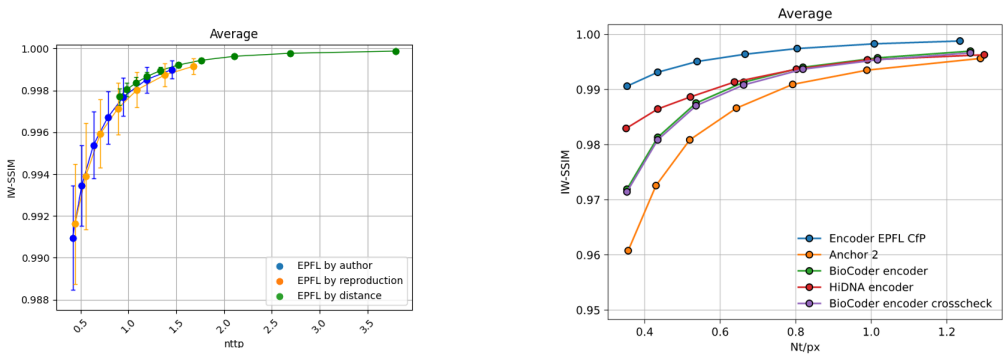


Figure 4.3: MS-SSIM IQA results: left - obtained from my experiment, right - reported in the referenced paper

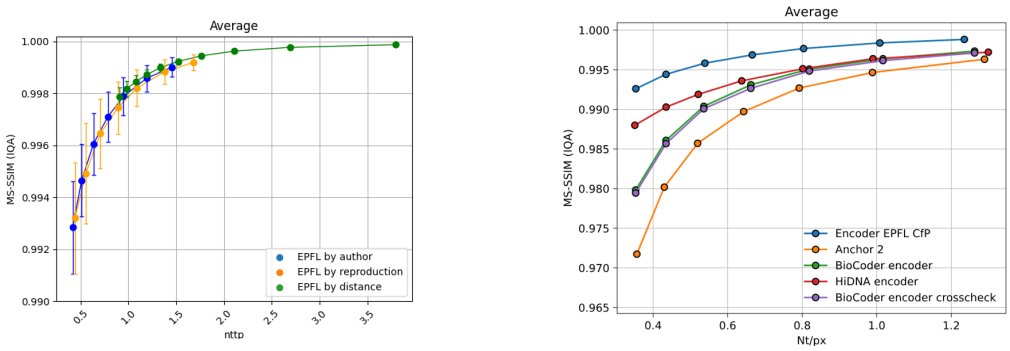


Figure 4.4: MS-SSIM PyTorch results: left - obtained from my experiment, right - reported in the referenced paper

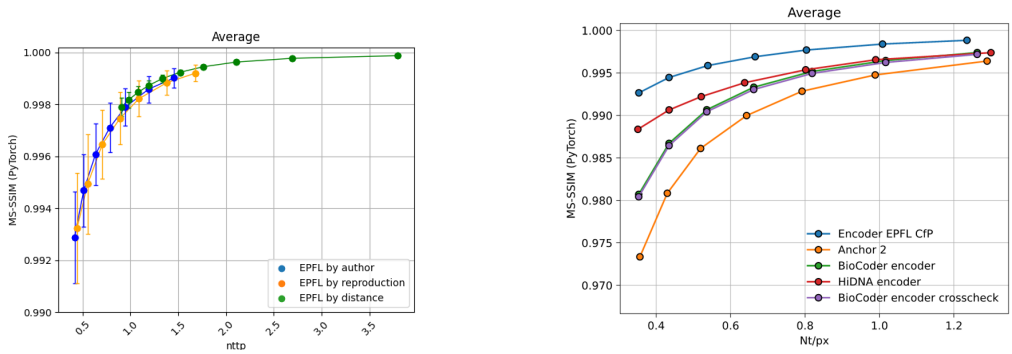


Figure 4.5: NLPD results: left - obtained from my experiment, right - reported in the referenced paper

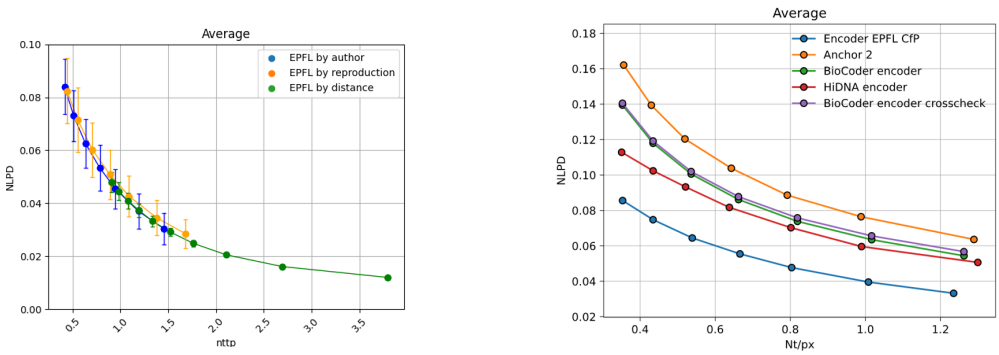


Figure 4.6: PSNR HVS results: left - obtained from my experiment, right - reported in the referenced paper

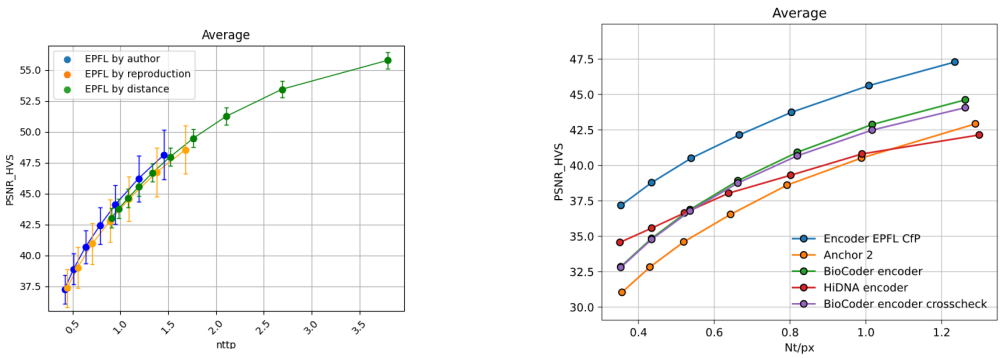


Figure 4.7: VIF results: left - obtained from my experiment, right - reported in the referenced paper

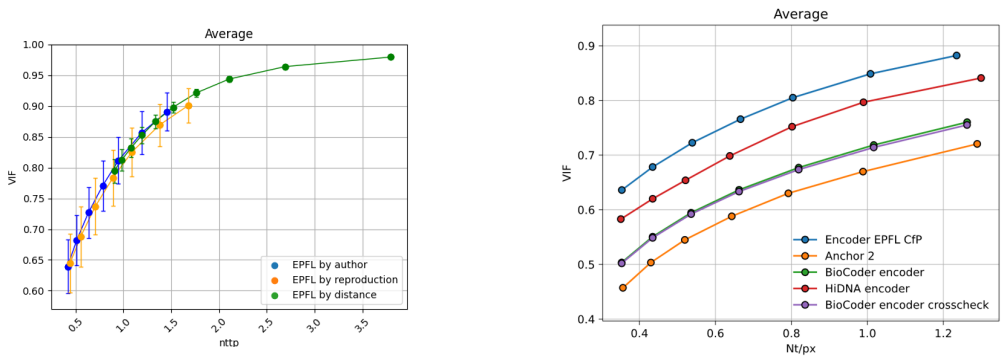


Figure 4.8: VMAF results: left - obtained from my experiment, right - reported in the referenced paper

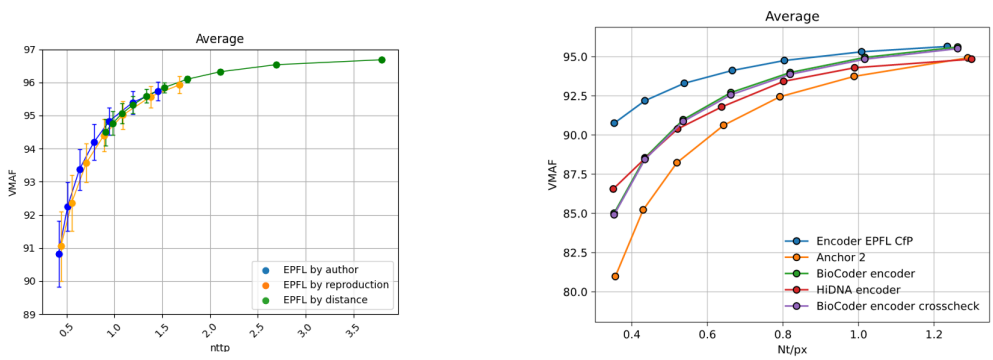


Figure 4.9: Comparison of PSNR results: Luminance (Y), Chrominance (U and V) from my experiment

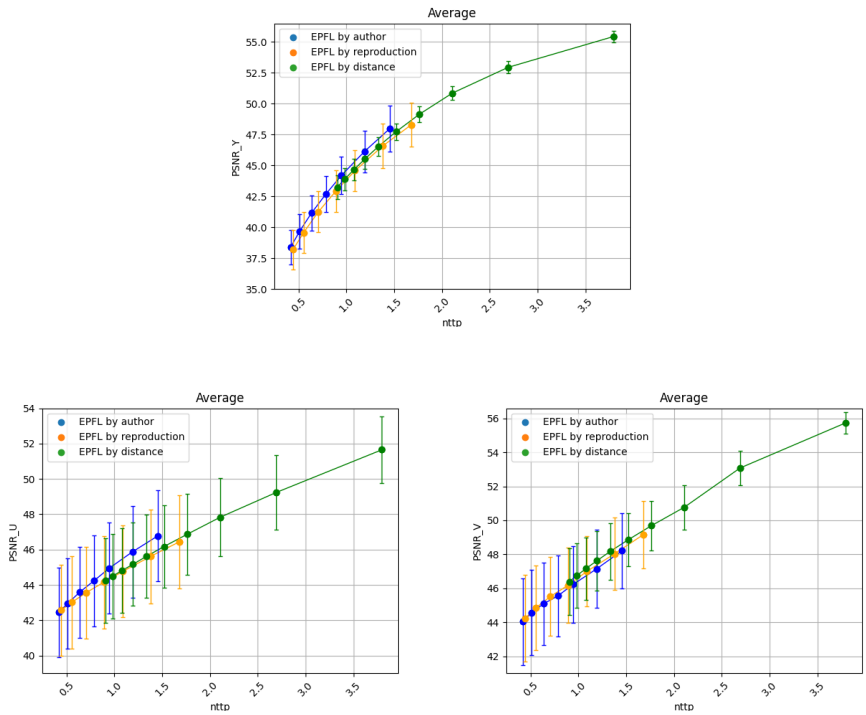
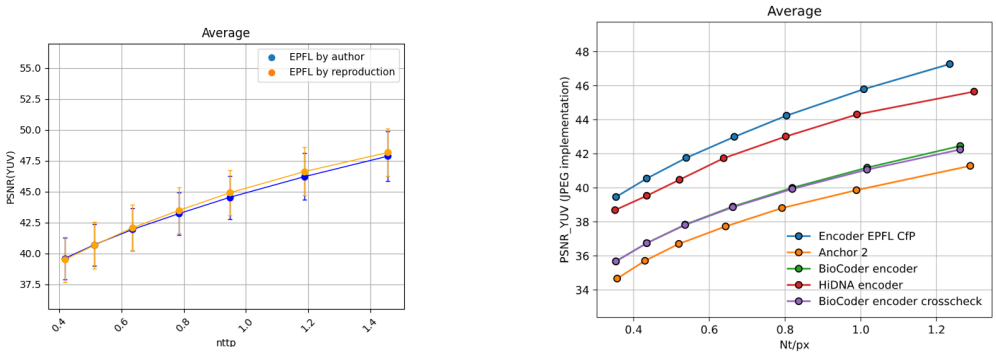


Figure 4.10: PSNR YUV results: left - obtained from my experiment, right - reported in the referenced paper



The results presented in the Figures illustrate that the outcomes generated by the EPFL codec closely align with reported findings across a range of metrics, demonstrating minimal differences with the reported values, and this difference more likely than not comes from the fact that the versions of the JPEG_XL software were different from the ones that were used in the results presented in the JPEG_DNA proposal. This evaluation includes metrics such as FSIM and PSNR, affirming the codecs robust performance. Moreover, the EPFL codec consistently demonstrates superior performance relative to all other codecs examined in the original proposal. These findings underscore its efficacy and reliability across varied evaluation criteria, positioning it as a leading contender in the JPEG_DNA Call for Proposal.

The graphical analysis provides compelling evidence of the EPFL codecs superiority across all evaluated criteria. It consistently outperforms reported benchmarks with minimal margin of error, showcasing robustness and reliability. These findings highlight the codecs potential as a standout choice in the JPEG_DNA Call for Proposal, reaffirming its position as a leading candidate based on comprehensive evaluation metrics.

All of the information, results and graphics are being stored in the following GitHub repository: Nunes [2024].

For more information on the HiDNA codec, another of the codecs created for the JPEG_DNA call for proposal, see Putyato [2024], also present is a project by a colleague of mine Artur Putyato.

4.4 Conclusion

The evaluation of the EPFL_JPEG_DNA codec has provided valuable insight into its performance across a variety of metrics and testing scenarios. Through systematic transformations from PNG to JPEG_XL to FASTA and back to PNG thorough the opposing order, the codec demonstrated its capability to maintain image fidelity and data integrity throughout complex encoding and decoding processes. The comprehensive analysis across various image quality metrics such as FSIM, IW-SSIM and many other metrics consistently showcased the codecs robustness and efficiency in preserving image quality.

Graphical comparisons between my experimental results and those reported in the original response to the JPEG_DNA proposal highlighted the codecs competitive edge, often surpassing other submissions to the proposal. Notably, the EPFL codec excelled in achieving lower nttp values, indicating superior compression efficiency without significant compromises on visual quality.

Moreover, the flexibility of the codec in parameter settings, including quality and distance adjustments, further enhances its versatility for diverse image storage and transmission needs. The findings underscore its potential applications not only in traditional digital imaging but also in cutting-edge biological storage systems leveraging DNA as a medium.

In conclusion, the EPFL_JPEG_DNA codec stands out as a robust and reliable solution for digital image preservation, offering advanced compression capabilities and maintaining high fidelity across multiple evaluation criteria. Its performance in this evaluation positions it strongly within the landscape of candidates vying for adoption in the JPEG_DNA initiative, promising significant advancements in image encoding technologies.

This comprehensive evaluation not only validates the codecs technical prowess but also lays the groundwork for future refinements and applications, solidifying its role in shaping the future of image compression and DNA based data storage.

Chapter

5

General Conclusions and Future Work

5.1 Conclusions

In conclusion, the EPFL codec represents a significant advancement in the field of data storage, using state of the art technologies from image compression to develop a novel and highly efficient solution. By employing the JPEG_XL format, the codec achieves efficient image compression, ensuring a minimal loss of quality, while maximizing the compression ratios. Combined with the robust and advanced error correction capabilities provided by the RAPTOR coder, the EPFL codec ensures a high-density storage solution for images encoded into DNA sequences.

The successful two-way conversion of digital images to DNA sequences, more specifically their FASTA representation, demonstrates a significant potential for using DNA as a long-term, high-capacity solution to address the emerging problem of storage capacity limitations. This novel approach not only creates a new paradigm for data storage, but it also pushes the boundaries of how we imagine and how we use biological molecules in the world of digital information.

The EPFL project exemplifies the kind of innovation required to meet the exponentially growing demand for data storage. By integrating sophisticated image compression techniques with advanced error correction algorithms, the project sets a new standard in the field. Furthermore, it paves the way for future research and development in data encoding solutions, opening up possibilities for even more efficient and reliable storage techniques and technologies.

In summary, the EPFL codec not only addresses the current challenges of data storage but also lays a solid foundation for ongoing and future advancements in this field. This field, the idea of DNA coding, demonstrates a successful combination between biological science and digital technology, marking the start of a new era of innovative data storage solutions that could revolutionize the way we think and the way we manage digital information.

5.2 Future Work

This report aimed to provide an analysis and explanation of the EPFL codec. However it is important to note that by the time this chapter is being written, there have been impressive results in the development of the JPEG_AI framework. The scope of JPEG_AI is the creation of a standard based in learning based image coding. When this report began being written, the JPEG_AI framework was still in a early development stage and as such I deemed it to premature to include a examination of this framework. This has recently changed due to the initiation of a study by the EPFL team with the goal of replacing the JPEG_XL with JPEG_AI in the DNA codec. With this information in mind, a possible future research should prioritize the analysis and comparison of the JPEG_AI codec with emerging competitors.

This report focused primarily on the encoding and decoding capabilities of the EPFL codec, without considering the presence of errors. As such, a potential future development could involve the evaluation of the codecs actual performance when confronted with errors. This could entail subjecting the codec with files coded with errors on purpose and testing its ability to decode them correctly.

Appendix

A

JPEG DNA Dataset

This appendix provides specific information about the image dataset used in the analyses presented in this report. The dataset was obtained from the original Call for Proposal and is cited in the paper Antonini and Ebrahimi [2023]. The initial dataset comprises ten images, each sourced from different origins and displaying a variety of sizes and levels of quality. The inclusion of diverse image characteristics in the dataset ensures a comprehensive analysis, thereby facilitating a thorough evaluation of the methods and results presented in the original codec paper. By incorporating images with diverse attributes, a more thorough comprehension of the effects and results of the implemented techniques can be achieved. The images are sequentially arranged as follows: images 1 to 5 are horizontally positioned in the first row from left to right, while images 6 to 10 are similarly arranged in the second row.

Figure A.1: JPEG DNA uncompressed dataset Antonini and Ebrahimi [2023]

The images displayed in the Figure A.1 showcase the variety and range of the dataset used in our analysis. Each image, selected for its unique characteristics, contributes to a comprehensive evaluation of the methods under review. To provide a more detailed understanding of these images, the Table A.1 summarizes key attributes of each image, including its content type and resolution.

IMAGE NUMBER	**CONTENT**	**RESOLUTION (pixels)**
00001	Object	1192x832
00002	Human face	853x945
00003	Food	945x840
00004	Computer-generated	2000x2496
00005	Animal	560x888
00006	Water	2048x1536
00007	Night scene	1600x1200
00008	Fabric/fine texture	1430x1834
00009	Landscape	2048x1536
00010	Buildings	2592x1946

Table A.1: Summary of the characteristics of the JPEG DNA uncompressed dataset Antonini and Ebrahimi [2023]

Appendix

B

JPEG AI Quality Assessment Framework Example

Table B.1: JPEG AI Metrics Example Table Karabutov [2021]

Metric	**Value**
File	ori.png
Codec	CODEC_NAME
BPP	0.001220703125
MS-SSIM (PyTorch)	0.8754700422286987
MS-SSIM (IQA)	0.737159013748169
PSNR_Y	10.80887709861642
PSNR_U	58.99101031445841
PSNR_V	58.99101031445841
VIF	0.45557889342308044
FSIM	0.99155193567276
NLPD	0.17580391466617584
IW-SSIM	-100
VMAF	55.168029
PSNR_HVS	6.679726338484886
MAC/pxl	196608.0
Decoding Time (GPU)	3.0
Decoding Time (CPU)	0.002263
Encoding Time (GPU)	N/A
Encoding Time (CPU)	N/A

The Table B summarizes the performance metrics presented by the JPEG_AI

metric evaluator as a example of how the results are presented. It should be taken in consideration that this is a supposed result of only one file. A real result would be a CSV file of dozens of lines of results.

Note: Values such as encoding or decoding times on GPU and CPU are not available (denoted as N/A), these always happens independently of the codec used.

Bibliography

Donald Adjeroh and Moon-Chuen Lee. Scene-adaptive transform domain video partitioning. *Multimedia, IEEE Transactions on*, 6:58 – 69, 03 2004. doi: 10.1109/TMM.2003.819578.

Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994. doi: 10.1126/science.7973651. URL https://www.science.org/doi/abs/10.1126/science.7973651.

Jyrki Alakuijala, Ruud Van Asseldonk, Sami Boukortt, Martin Bruse, Iulia-Maria Comșa, Moritz Firsching, Thomas Fischbacher, Evgenii Kliuchnikov, Sebastian Gomez, Robert Obryk, et al. Jpeg xl next-generation image compression architecture and coding tools, 2019. URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11137/111370K/JPEG-XL-next-generation-image-compression-architecture-and-coding-tools/10.1117/12.2529237.full#_=_.

Marc Antonini and Touradj Ebrahimi. Jpeg dna common test conditions version 2.0, 2023. URL https://ds.jpeg.org/documents/jpegdna/wg1n100517-099-ICQ-JPEG_DNA_Common_Test_Conditions_v2.pdf. Accessed 18 April 2024.

A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

Ewen Callaway. Million-year-old mammoth genomes shatter record for oldest ancient dna. In *Nature*, volume 590,7847, 2021. doi: 10.1038/d41586-021-00436-x. URL https://pubmed.ncbi.nlm.nih.gov/33597786/.

George M. Church, Yuan Gao, and Sriram Kosuri. Next-generation digital information storage in dna. *Science*, 337(6102):1628–1628, 2012. doi: 10.1126/science.1226355. URL https://www.science.org/doi/abs/10.1126/science.1226355.

Melpomeni Dimopoulou. *Encoding techniques for long-term storage of digital images into synthetic DNA*. Theses, Université Côte d'Azur, December 2020. URL https://theses.hal.science/tel-03185468.

Melpomeni Dimopoulou, Eva Gil San Antonio, and Marc Antonini. A jpeg-based image coding solution for data storage on dna. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 786–790, 2021. doi: 10.23919/EUSIPCO54536.2021.9616020.

DMLapato. Cluster generation, 2015. URL https://commons.wikimedia.org/wiki/File:Cluster_Generation.png. This file is licensed under the Creative Commons Attribution-Share Alike 4.0 International license, https://creativecommons.org/licenses/by-sa/4.0/deed.en.

Jarek Duda. Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding, 2014.

Enzoklop. Schematic mechanism of pcr, 2020. URL https://commons.wikimedia.org/wiki/File:Polymerase_chain_reaction-en.svg. This file is licensed under the Creative Commons Attribution-Share Alike 4.0 International license, https://creativecommons.org/licenses/by-sa/4.0/deed.en.

Michel Gauthier. *Simulation of polymer translocation through small channels: A molecular dynamics study and a new Monte Carlo approach*. PhD thesis, University of Ottawa, 2008. URL http://dx.doi.org/10.20381/ruor-19878.

Nick Goldman. Method for encoding and decoding arbitrary computer files in dna fragments, August 2013. URL https://www.ebi.ac.uk/sites/ebi.ac.uk/files/groups/goldman/file2features_2.0.pdf. Version 2.0: 21 August 2013.

Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M. Leproust, Botond Sipos, and Ewan Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized dna. *Nature*, 494:77–80, 2013. URL https://www.nature.com/articles/nature11875.

Robert N Grass, Reinhard Heckel, Marcello Puddu, Damian Paunescu, and Wendelin J Stark. Robust chemical preservation of digital information on dna in silica with error-correcting codes. *Angewandte Chemie International Edition*, 54(8):2552–2555, 2015. doi: 10.1002/anie.201411378.

Randall A Hughes and Andrew D Ellington. Synthetic dna synthesis and assembly: Putting the synthetic in synthetic biology. *Cold Spring Harb Perspect Biol*, 9(1):a023812, 2017. doi: 10.1101/cshperspect.a023812. URL https://doi.org/10.1101/cshperspect.a023812.

Luca Versari Jyrki Alakuijala, Jon Sneyers and Jan Wassenberg. Jpeg white paper: Jpeg xl image coding system. https://ds.jpeg.org/whitepapers/jpeg-xl-whitepaper.pdf, 2023.

Alexander Karabutov. Jpeg ai quality assessment framework. https://gitlab.com/wg1/jpeg-ai/jpeg-ai-qaf, 2021.

V. Laparra, J. Ballé, A. Berardino, and E. P. Simoncelli. Perceptual image quality assessment using a normalized laplacian pyramid. In *S&T Symposium on Electronic Imaging: Conf. on Human Vision and Electronic Imaging*, San Francisco, CA, USA, February 2016.

Davi Lazzarotto, Jorge Encinas Ramos, Michela Testolina, and Touradj Ebrahimi. Technical description of the epfl submission to the jpeg dna cfp, 2023. URL https://arxiv.org/pdf/2312.00560.

Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara. Toward a practical perceptual video quality metric. https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652, 2023.

Zhijing Li, Christopher De Sa, and Adrian Sampson. Optimizing jpeg quantization for classification networks, 2020.

Bichlien H. Nguyen, Julie C. Sinistore, Jake A. Smith, Praneet Singh Arshi, Lauren M. Johnson, Tim Kidman, T. J. diCaprio, Douglas M. Carmean, and Karin Strauss. Architecting datacenters for sustainability: Greener data storage using synthetic dna, 2020. URL https://api.semanticscholar.org/CorpusID:230672051.

NHGRI, 2023. URL https://www.genome.gov/sites/default/files/tg/en/illustration/double_helix.jpg. Accessed 26 April 2024.

Alexandre Nunes. Study and evalutaion of epfl jpeg dna codec, 2024. URL https://github.com/AlexandreNunes0/Study-and-evaluation-of-EPFL_JPEG_DNA-codec.

C. Pan, S. Yazdi, S. K. Tabatabaei, A. G. Hernandez, C. Schroeder, and O. Milenkovic. Image processing in dna. *arXiv preprint*, arXiv:1910.10095, 2019.

W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, 85(8):2444–2448, April 1988. doi: 10.1073/pnas.85.8.2444.

X. Pic, M. Dimopoulou, E. G. S. Antonio, and M. Antonini. Mq-coder inspired arithmetic coder for synthetic dna data storage. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3304–3308, Kuala Lumpur, Malaysia, 2023. doi: 10.1109/ICIP49359.2023.10223113.

N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin. On between-coefficient contrast masking of DCT basis functions. In *Proceedings of the third international workshop on video processing and quality metrics*, volume 4, 2007.

Artur Putyato. Study and evalutaion of hidna codec, 2024. URL `https://github.com/Pobedinsky/Study-and-Evaluation-of-the-HiDNA-codec`.

JE Ramos, D Lazzarotto, M Testolina, and T Ebrahimi. Analysis of the influence of errors in dna-based image coding. In *CORESA*, Lille, France, June 7-9 2023.

Frederick Sanger and Alan R Coulson. A rapid method for determining sequences in dna by primed synthesis with dna polymerase. *Journal of Molecular Biology*, 94(3):441–448, 1975. doi: 10.1016/0022-2836(75)90213-2.

P. M. Schwarz and B. Freisleben. Norec4dna: using near-optimal rateless erasure codes for dna storage. *BMC Bioinformatics*, 22(1):406, 2021. doi: 10.1186/s12859-021-04318-x. URL `https://doi.org/10.1186/s12859-021-04318-x`.

Luka Secilmis, Michela Testolina, Davi Lazzarotto, and Touradj Ebrahimi. Towards effective visual information storage on dna support. In Andrew G. Tescher and Touradj Ebrahimi, editors, *Applications of Digital Image Processing XLV*, volume 12226, page 1222605. International Society for Optics and Photonics, SPIE, 2022. doi: 10.1117/12.2636678. URL `https://doi.org/10.1117/12.2636678`.

H. R. Sheikh and A. C. Bovik. Image information and visual quality. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, August 2004.

Amin Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52: 2551–2567, 2011. URL `https://api.semanticscholar.org/CorpusID:1731099`.

Amin Shokrollahi, Thomas Stockhammer, Michael Luby, and Mark Watson. Raptor forward error correction scheme for object delivery. RFC 5053, oct 2007. URL https://www.rfc-editor.org/info/rfc5053.

Taylor and Petroc. Amount of data created, consumed, and stored 2010-2020, with forecasts to 2025. https://www.statista.com/statistics/871513/worldwide-data-created/, 2023. Accessed 4 April 2024.

Michela Testolina, Davi Lazzarotto, and Touradj Ebrahimi. Report on cross-checking of the biocoder and hidna submissions to the jpeg dna cfp, 2023. wg1m101156.

UNITAR. Global e-waste monitor 2024: Electronic waste rising five times faster than documented e-waste recycling. https://www.unitar.org/about/news-stories/press/global-e-waste-monitor-2024-electronic-waste-rising-five-times-faster-documented-e-waste-recycling, 2024. Accessed: 2024-06-23.

Z. Wang and Q. Li. Information content weighting for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 20(5), May 2011.

Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *ThirtySeventh Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, CA, USA, November 2003.

L. Zhang, L. Zhang, X. Mou, and D. Zhang. FSIM: a feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20 (8):2378–2386, August 2011.