Master Project

Point cloud compression for DNA based storage

Author:
Romain Graux

Supervisor:
Dr. Touradj Ebrahimi

Chapter 1

Introduction

Each year, we produce more data than the previous one. All those datas are stored in multiple datacenters spread all over the world. Those datacenters require a lot of energy to maintain bits of information in spinning disks, tapes, capacities, transistors, ...

The Independent reported in 2016 that data centers will consume three times as much energy as they are currently using over the course of the next decade. [1]

It becomes naturally important to find more eco-friendly ways to store data.

Chapter 2

Problem defintion

In this work, we will try to build an end-to-end point cloud compression model for quatarnary based entropy coding.

We will first discover the state of the art of compression models for point cloud that leads to the best bitrates for various point clouds and then adapt it to output **A**, **C**, **G** and **T** symbols instead of the classical 0 and 1 binary base.

Then we will have to study the state of the art of DNA based storage and how we can store our long sequence of **A**, **C**, **G** and **T** in the most efficient way to minimise the cost of storing those datas while trying to recover the compressed point cloud in the most XXXX way. For this last part, we will have to study the influence of each parameters to store the DNA strand as small chunks in a solution. At the end, we will have to propose the best parameters to store a DNA strand for a certain period of time.

And finally, we will have to provide the reconstruct model to go from the compressed version to the point cloud as closed as possible from the original one.

To continue

Chapter 3

State of the art

3.1 DNA based storage

As we produce more data every year, we need to find a way to store it efficiently. Currently, we store our data in big data center consuming a lot of energy to keep these informations in electronic devices, **Find the consumption of data centers**

It would be a good idea to find a way to store our data in a more efficient and ecological way. For storing information, hard drives don't hold a candle to DNA. Our genetic code packs billions of gigabytes into a single gram. A mere milligram of the molecule could encode the complete text of every book in the Library of Congress and have plenty of room to spare. [2]

But it can not be applied to all data types, for example, it is not possible yet to replace an USB stick by a DNA based USB stick and expecting the same experience. The information retrieval latency and high cost of the DNA sequencer and other instruments "currently makes this impractical for general use," says Daniel Gibson, a synthetic biologist at the J. Craig Venter Institute in Rockville, Maryland, "but the field is moving fast and the technology will soon be cheaper, faster, and smaller." Gibson led the team that created the first completely synthetic genome, which included a "watermark" of extra data encoded into the DNA. [2]

This does not mean that there are no applications for DNA based storage. DNA based storage can be used for long term media preservation archives (so called cold media storage) which are infrequently accessed and thus do not need low information retrieval latency.

3.1.1 Constraints

Unfortunately, nucleic acids have biological constraints and can not be assembled in any order like it is the case for binary digits. The DNA strands have to be created in a way that the double helix binds well together and is not immediately desctructed. We must therefore respect the biological constraints to build strong strands that can last for a certain period of time.

In this part, we are going to go through some of the constraints that we have to respect to build a DNA strand. Unfortunately, the list of constraints is not exhaustive and in the real life, each arangement of nucleic acids has an impact on the strength of a strand, therefore we can only simulate the longevity but not strictly respect the constraints.

To continue

3.1.2 Requirements

To continue

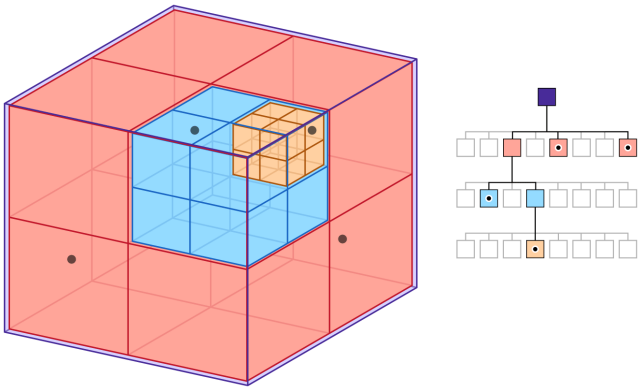

Figure 3.1: Octree representation of a point cloud

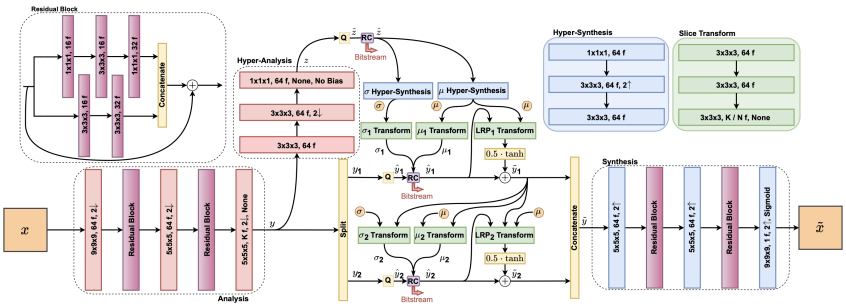

Figure 3.2: The "Latent Space Slicing For Enhanced Entropy Modeling in Learning-Based Point Cloud Geometry Compression" architecture with 2 slices.

3.2 Point cloud compression

Numerous methods have been proposed to compress point clouds in the literature. They are usually based on different structures than a classical list of coordinates. Octrees, for example, have been widely used for this purpose [3]. The octree representation consists of diving recursively the three dimensional space as nodes of a tree as shown on Figure 3.1.

Compression algorithms using learning based autoencoders architectures have also demonstrated good performance. While some take as input point coordinates [4], others take as input voxelized versions of the point clouds. Voxelized point clouds consist of occupancy grid of regular spaced points so that several points are merged together in a single voxel. A voxel is similar to a three dimensional pixel. These three dimensional grids can be then used as input for a 3D convolutional layer.

The current state of the art for point cloud compression that we will use in this project is a model called "Latent Space Slicing For Enhanced Entropy Modeling in Learning-Based Point Cloud Geometry Compression" and that has been developed by Nicolas Frank, Davi Lazzarotto and Touradj Ebrahimi at the MMSPG laboratory (EPFL).

3.2.1 Model architecture

This model is shown on Figure 3.2 and consists in a 3D autoencoder architecture with latent entropy coding. The input of the model is an occupancy cubic grid with $k \times k \times k$ voxels represented by 1 when occupied and 0 otherwise.

The first block (Analysis transform) of the model is composed of 3 3D convolutional layers and 2 convolutional residual blocks arranged staggered which produces a latent representation y of shape $l \times l \times l \times d$ with d being the latent dimension.

This latent representation is then fed into an Hyper-Analysis transform block yielding z. This hyperprior is passed to the bitstream as side information after quantization, and is used to model the entropy of the quantized latent features $\hat{y}$ after going through the hyper-synthesis.

While in other solutions for learning-based point cloud compression the hyperprior would be the only variable used to estimate the scale and mean of $\hat{y}$, they use previously decoded

channels for entropy modeling.

The latent representation y is sliced along the channel dimension into N non-overlapping and equally sized tensors y_i with $i \in \{1, \ldots, N\}$.

Once the latent representation y is sliced, they compute the entropy parameters (μ_i, σ_i) for each slice y_i from the global entropy parameters (μ, σ) and the previously decoded latent representation slices $\tilde{y}_j \, \forall j \in \{0, \ldots, i-1\}$.

The final latent representation reconstruction $\tilde{y}_i$ is produced from $\hat{y}_i$ after going through a latent residual prediction (LRP) transform that predicts the quantization error $y_i - \hat{y}_i$ in order to take into account this error from the global entropy parameters (μ, σ), the current $\hat{y}_i$ and previously decoded latent representation slices $\tilde{y}_j \, \forall j \in \{0, \ldots, i\}$. A *tanh* non-linearity scaled by a factor 0.5 is applied to the output of the transform to keep the output of the LRP within the range of quantization error. The predicted residuals are then added to $\hat{y}_i$, generating $\tilde{y}_i$. These slices are then concatenated along the channel dimension before going through the synthesis transform. This last learned block finally generates the output block $\tilde{x}$ containing a probability estimation for the occupancy of each voxel.

Chapter 4

Implementation

4.1 DNA latent representation

In this section, we will plug all parts together in order to implement a latent representation of DNA. With these pieces of DNA strands, we will be able to synthesis and encapsulate them in a real medium to preserve them.

The first step in order to achieve this is to extract the features of the point clouds with the analysis transform of the model presented previsously.

Bibliography

[1] A. Dellinger, "The environmental impact of data storage is more than you think — and it's only getting worse," jun 2019. https://www.mic.com/impact/the-environmental-impact-of-data-storage-is-more-than-you-think-its-only-getting-worse-18017662.

[2] J. Bohannon, "Dna: The ultimate hard drive," aug 2012. https://www.wired.com/2012/08/dna-data-storage.

[3] J. Kammerl, N. Blodow, R. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in *Proceedings - IEEE International Conference on Robotics and Automation*, 05 2012.

[4] X. Wen, X. Wang, J. Hou, L. Ma, Y. Zhou, and J. Jiang, "Lossy geometry compression of 3d point cloud data via an adaptive octree-guided network," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2020.