EasyChair Preprint

№ 4816

Quantum Generators: Foundations of the Compute Units in Pattern Reconstruction.

Poondru Prithvinath Reddy

December 28, 2020

Quantum Generators: Foundations of the Compute Units in Pattern Reconstruction.

Poondru Prithvinath Reddy

ABSTRACT

Quantum Generators is a means of achieving mass food production with short production cycles, and when and where required by means of machines rather than land based farming which has serious limitations. The process for agricultural practices for plant growth in different stages is simulated in a machine with a capacity to produce multiple seeds from one seed input using computational models of multiplication (generating multiple copies of kernel in repetition). In this paper, we present the structure of Compute Units resulted by the computational models of multiplication and also present methods related to manipulation of Compute Units so that they can be linked to tissues of the kernel which mimic the real cell structure that grows into full-fledged natural tissue. We use simulation to show that we achieve manipulative structure with respect to the size of the input space and realize good pattern. The results suggest that it is possible to achieve viable cell structure for quantum generation.

INTRODUCTION

A **Quantum** (plural quanta) is the minimum amount of any physical entity (physical property) involved in an interaction. On the other hand, **Generators** don't actually create anything instead, they generate quantity prescribed by physical property through multiplication to produce high quality products on a mass scale. The aim of Quantum Generators is to produce multiple seeds from one seed at high seed rate to produce a particular class of food grains from specific class of **seed** on mass scale by means of machine rather than land farming.

The process for agricultural practices include preparation of soil, seed sowing, watering, adding manure and fertilizers, irrigation and harvesting. However, if we create same conditions as soil germination, special watering, fertilizers addition and plant growth in different stages in a machine with a capacity to produce multiple seeds from one seed input using computational models of multiplication(generating multiple copies of kernel in repetition) then we will be closure to achieving mass food production by means of quantum generators(machine generated) rather than traditional land based farming which has very serious limitations such as large space requirements, uncontrolled contaminants, etc. The development of Quantum Generators requires specialized knowledge in many fields including Cell Biology,

Nanotechnology, 3D Cellprinting, Computing, Soil germination and initially they may be big occupying significantly large space and subsequently small enough to be placed on roof-tops.

The Quantum Generators help world meet the food needs of a growing population while simultaneously providing opportunities and revenue streams for farmers. This is crucial in order to grow enough food for growing populations without needing to expand farmland into wetlands, forests, or other important natural ecosystems. The Quantum Generators use significantly less space compared to farmland and also results in increased yield per square foot with short production cycles, reduced cost of cultivation besides easing storage and transportation requirements.

In addition, Quantum Generators Could Eliminate Agricultural Losses arising out of Cyclones, Floods, Insects, Pests, Droughts, Poor Harvest, Soil Contamination, Land Degradation, Wild Animals, Hailstorms, etc.

Quantum generators could be used to produce most important *food* crop *like* rice, wheat and maize on a mass scale and on-demand when and where required.

Computers and Smartphones have become part of our lives and Quantum Generators could also become very much part of our routine due to its potential benefits in enhancing food production and generating food on-demand wherever required by bringing critical advanced technologies into the farmland practices.

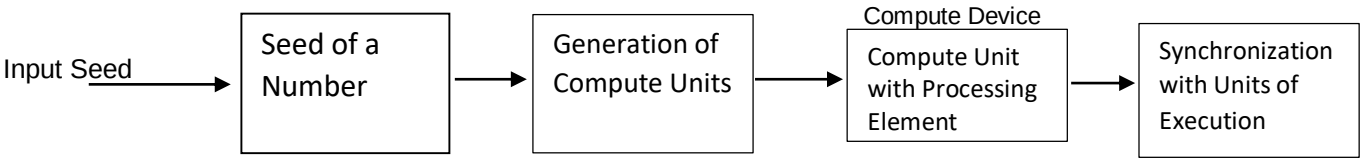
3D Bioprinting

3D Bioprinting is a form of additive manufacturing that uses cells and other biocompatible materials known as bioinks, to print living structures layer-by-layer which mimic the behavior of natural living systems. Three dimensional bioprinting is the utilization of 3D printing–like techniques to combine cells, growth factors, and biomaterials to fabricate biomedical parts that maximally imitate natural tissue characteristics.

Bioprinting (also known as **3D bioprinting**) is combination of **3D printing** with biomaterials to replicate parts that imitate natural tissues, bones, and blood vessels in the body. It is mainly used in connection with drug research and most recently as cell scaffolds to help repair damaged ligaments and joints. In this paper, we are looking at natural tissues related to food crops like rice, wheat or maize.

BASIC CONCEPTS and THE ARCHITECTURE

We give below a description of multiplication model, its execution design, its reconstruction model and synchronization support.



A Quantum Generator device has one or more Compute Units. A work-group executes on a single Compute unit. A Compute Unit is composed of one Processing Element and Seed Object. A Compute Unit may also include filter Units that can be accessed by its processing elements.

A Device is a collection of Compute Units. Quantum Generator device typically corresponds to a collection of multiple Compute Units generated by the seed of a number.

A Seed is a function declared in a program and executed on a quantum generating device. A seed is identified by the Seed Qualifier applied to any function defined in any program.

A Seed Object encapsulates a specific seed function declared in a program and the argument values to be used when executing this Seed Function.

A Synchronization refers to mechanisms that define the order of execution and the visibility of operations between two or more units of execution. The Operations are that define order controls in a program. They play a special role in controlling how operations of in one unit of execution (such as work-items) are made visible to another. Synchronization essentially involves establishing a relation between operations in two different units of execution that define an order control in a device.

Seed Objects

A seed is a function declared in a program. A seed is identified by the seed qualifier applied to any function in a program. A Seed Object encapsulates the specific seed function declared in a program and the argument values to be used when executing this seed function.

Seed Objects are created for any seed functions in program that have the same function definition across all Compute Units for which a program has been built successfully in a device.

Support Vector Machine for Pattern Reconstruction

Kernel machines are a class of algorithms for pattern analysis, whose best known member is the Support Vector Machine. Kernel methods use kernel functions, which enable them to operate in a high-dimensional, implicit feature space and this operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the "**kernel trick**" and Kernel functions have been introduced for sequence data.

Algorithms capable of operating with kernels include Kernel Perceptron, Support Vector Machine (SVM), Gaussion Processes, etc. Any linear model can be turned into a non-linear model by introducing kernel function to the model replacing the features by a kernel function.

Support Vector Machines (SVM) are widely used for classification problems in machine learning. The SVM is essentially used for simple class separation and it tries to find a line/hyperplane (in multidimensional space) that separates the classes. Subsequently it classifies the new point depending on whether it lies on the positive or negative side of the hyperplane depending on the class to predict.

In the SVM classifier, it is easy to have a linear hyper-plane between any two classes. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs we've defined.

Hyperparameters of the SVM Algorithm

The following are the few important parameters of SVM:-

- Kernel – A kernel helps us find a hyperplane in the higher dimensional space without increasing the computational cost. Usually, the computational cost will increase if the dimension of the data increases. This increase in dimension is required when we are unable to find a separating hyperplane in a given dimension and are required to move in a higher dimension.
- Hyperplane – This is basically a separating line between two data classes in SVM.
- Decision Boundary – A decision boundary can be thought of as a demarcation line on one side of which lie positive examples and the other side lie the negative examples on this very line, the examples may be classified as either positive or negative.

Kernel methods with some fixed set of parameters corresponding to the features of their inputs and we used SVM algorithm for pattern recognition however, data points are not mapped to a 3-dimensional space wherein a separating hyperplane can easily be found.

As there are multiple Compute Units represented by the seed of a number, We used iterative reconstruction i.e. iterative algorithm to reconstruct pattern of a sequence of data generated by each of the Seed Object in a Compute Unit.

Since, Compute Units are similar across a device, we use recursion to generate a Seed Structure in multiple units across a quantum device.

In this paper, we have dealt with simulation of Compute Units to show that we achieve manipulative structure with respect to the input space and not about synchronizing Compute Units to tissues of the kernel which mimic the real cell structure that grows into full-fledged natural tissue.

Unlike the simulation results which are based on few parameters, In natural or real tissues which are 3D Bioprinted there are number of parameters to be considered for pattern analysis.

The QG System

Our objective is to build a target system, we need to generate the cell for the device by running synthesis and implementation on the design. The cell includes custom logic for every Compute unit in the cell container. The generation of custom compute units uses the High-Level synthesis tool, which is the computer unit generator in the application compilation flow. Therefore, it is normal for this step to run for longer period of time than the other steps in the system build flow.

After all compute units have been generated, these units are connected to the infrastructure elements provided by the target device in the solution. The infrastructure elements in a device are all of the memory, control and output data planes which the device is formulated to support an application. The environment combines the custom compute units and the base device infrastructure to generate a cell binary which is used to program the QG device during application execution.

The processing flow of application execution is given as below:-

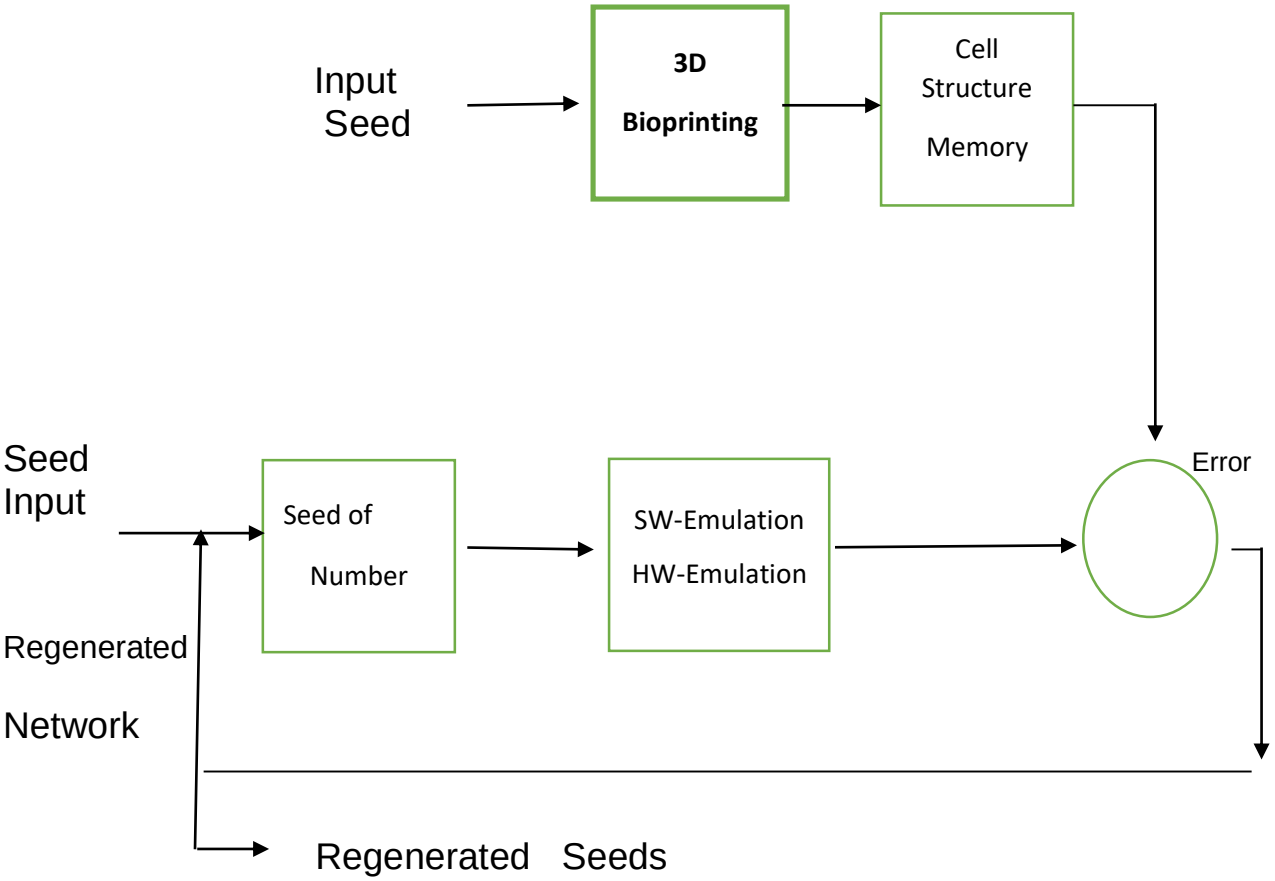


Fig. 1 Process Flow in a Quantum Generator.

The different steps in application are as below:-

1. 3D print a seed and copy its cell structure to memory.
2. Input seed with a seed of a number required.
3. Generate a seed kernel once.
4. Compare the kernel with 3d printed cell
5. If error in seed structure, generate the kernel again.
6. Repeat many times till the seed number is met.

TEST RESULTS

We have generated sequence of data based on function parameters and applied SVM to find a hyperplane and pattern recognition which is of significantly popular algorithm. Although, we have generated seed structure with good pattern but this is not mapped to original tissues of seed kernel which are in 3D plane to test the deviation.

CONCLUSION

Quantum Generators (QG) creates new seeds iteratively using the single input seed and the process leads to a phenomenon of generating multiple copies of kernels in repetition. We presented the structure of Compute Units which can be manipulated to mimic the tissues of real kernel. The results suggest that it is possible to achieve well-organized cell structure for quantum generation.

REFERENCE

1. Poondru Prithvinath Reddy: “Quantum Generators: A Formulation of Computational Models of Multiplication”, Google Scholar.