HOMEWORK

알고리즘설계및해석 - CEG608

BRIEF INTRODUCTION

This is Bishal Ranjan Swain (비샬), a PhD candidate in the department of Computer Engineering under the guidance of Prof. JaePil Ko. I completed my Masters in India with my research focus on development of tools to visualize and predict tropical cyclones. I also worked in a network lab for a year where I used ML to allocate resources and recommend movement of data paths in a high mobility cluster.

RESEARCH FIELD

I would like to conduct my research in implementing Deep learning models in high mobility and low on-board resource devices such as drones to function intelligently. This requires not only models with high accuracy but also quick execution. For such time sensitive problems, the underlying algorithms used in the data pipelines play a pivotal role.

Another research area which interests me is 3D computer vision especially research works bound but not limited to Neural Radial Fields(NeRF). High complex problems such as these require proper management and handling of data which can be solved by using proper algorithms.

ALGORITHM USED

I faced a problem where I was working with large amount of scientific data and I needed a way to quickly calculate the inverse square-root of a number ($\frac{1}{\sqrt{x}}$). The Algorithm that I ended up using was Quake's Fast Inverse Square Root Algorithm (Quake is a 3D game that does a lot of such operations).

비샬 – 20225055

```
float InvSqrt(float x){
        float xhalf = 0.5f * x;
        int i = *(int*)&x;              // store floating-point bits in integer
        i = 0x5f3759df - (i >> 1);      // initial guess for Newton's method
        x = *(float*)&i;                // convert new bits into float
        x = x*(1.5f - xhalf*x*x);       // One round of Newton's method
        return x;
    }
```

The algorithm uses Newton's Method of approximation and finds the output using only multiplication and bit-shift operations.

The algorithm converts the floating point into integer and right shifts by one (which is same as diving by two) and to negate the exponent it subtracts from the number - 0x5f3759df. By doing so, it preserves the mantissa, handles odd-even exponents, etc. Thus, the result is that the initial guess is really close to the real inverse square root. Lastly the Newton's method is used to refine the guess. More rounds of Newton's method can be used for more accuracy at additional computational cost.

For more information - link to the paper.

COMPARISON

The traditional approach to finding the inverse of the square root of a number involves doing exponential and division-based calculations which are resource expensive. When working with a large amount of data that require numerous such calculations performing traditional based approaches often becomes a bottleneck. Even a slight improvement in performance optimizes the problem statement immensely.

Over the years there have been many refinements in the Fast Inverse Square Root algorithm, one such improvement includes, applying Newton-Raphson correction method which increases the accuracy of prediction by two-fold and almost seven-fold if done twice.

Additional Links-> link1, link2