Design and Analysis of Algorithms (24CS2203)

Title: Strassen's Algorithm for Speeding Up Matrix Calculations in 3D Rendering Pipelines

S. Aswath Nag: 2420080009

D.Sri Charan Teja:2420080051

Course Instructor

Dr. J Sirisha Devi

Professor

Department of Computer Science and Engineering

Case study - statement

Case Study Statement:

3D graphics rendering depends heavily on matrix transformations for scaling, rotation, translation, and projection. Since standard matrix multiplication uses $O(n^3)$ operations, it becomes computationally expensive when processing thousands of vertices per frame.

This study applies Strassen's Algorithm, which reduces multiplication time to $O(n^{2.81})$,to enhance the efficiency of matrix operations in rendering pipelines.

Objective:

To optimize matrix multiplications in rendering engines by applying Strassen's divide-and-conquer approach to improve performance in real-time 3D scenes.

Algorithm /Pseudo code

- Divide matrices into 4 submatrices:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

- Compute 7 intermediate products:

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22})B_{11}$$
$$M_3 = A_{11}(B_{12} - B_{22})$$

Algorithm /Pseudo code

- Remaining intermediate products:

  $M4=A22(B21-B11)$

  $M5=(A11+A12)B22$

  $M6=(A21-A11)(B11+B12)$

  $M7=(A12-A22)(B21+B22)$

- Combine results to form matrix C:

  $C11=M1+M4-M5+M7$

  $C12=M3+M5$

  $C21=M2+M4$

  $C22=M1-M2+M3+M6$

Time Complexity

- Strassen's Algorithm reduces classical $O(n^3)$ time to approximately $O(n^{2.81})$.
- This is due to reducing scalar multiplications from 8 to 7 through recursive divide-and-conquer steps.
- Provides significant speed-up in large matrix multiplications common in 3D graphics transformations.

Space Complexity

- Requires additional space $O(n^2)$ for intermediate submatrices in recursion.
- Extra memory needed for storing seven products at each level.
- Memory management can be optimized by in-place computations for small matrix sizes.