Subject : ___ / / ___ موضوع الدرس :

#problem 7:-

* In computer graphics, there are many operations that is performed on objects such:

- translation
- Scaling
- Rotation

* Translation:- is the process an object from position (X,Y) to a new position (X',Y')

⇒ We achieve translation by adding vector to the old position translation value (Tx,Ty)

Subject: / / موضوع الدرس:

* Translation liner equation

$X_{new} = X_{old} + T_x$
$Y_{new} = Y_{old} + T_y$

⇛ the best way to store these linear equation is to store them as 2D matrix in a computer and applay matrix operation on them

* so translation can be described as

① $[X_{new} \quad Y_{new}] = [X_{old} \quad Y_{old}] + [T_x \quad T_y]$
(~~X~~) (~~X~~)

OR

② $[X_N \quad Y_N \quad 1] = [X_o \quad y_o \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$

Subject : ____ / / ____ موضوع الدرس :

as in the problem vertex is represented as (X, Y, 1)

⇒ So we will use second method: Matrix multiplication

* Scaling

⇒ Scaling is used to resize the object size by increasing or decreasing it or stay the same by scaling factors Sx, Sy

* Scaling linear equation:

$$X_N = X_O \cdot S_x$$

$$Y_N = Y_O \cdot S_y$$

⇒ We will use 2D matrix to represent these linear equation on a computer

Subject : / / موضوع الدرس :

⇒ The representation can be done using 2 forms

$$1- \begin{bmatrix} x_N & y_N \end{bmatrix} = \begin{bmatrix} x_O & y_O \end{bmatrix} \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix}$$

$$2- \begin{bmatrix} x_N & y_N & 1 \end{bmatrix} = \begin{bmatrix} x_O & y_O & 1 \end{bmatrix} \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

⇒ we will use second representation as vertex is represent in this proble as (x y 1)

Subject: / / موضوع الدرس :

⇛ The algorithm to calculate scaling and translation is Matrix multiplication

* a Naive algorithm to multiply matrices

* Pesudo Code

Matrix Mutiplication (A, B, C)

{

For i ← 1 to Size[A] {

For j ← 1 to Size[B] {

$$C_{ij} = \sum_{k=1}^{N} A_{ik} * B_{kj}$$

}

}

}

⇒ The time complexity of this algorithm is $O(n^3)$

* we can use divide and conquer technique to reduce time complexity.

⇒ we divide matrices A and B into 4 sub-matrices of size $N/2 \times N/2$

⇒ Calculate values recursively

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \qquad B = \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

$$C_{11} = A_{11} * B_{11} + A_{12} * B_{21}$$
$$C_{12} = A_{11} * B_{12} + A_{12} * B_{22}$$
$$C_{21} = A_{21} * B_{11} + A_{22} * B_{21}$$
$$C_{22} = A_{21} * B_{12} + A_{22} * B_{22}$$

- Addition takes $O(n^2)$ and there are 8 multiplication

so Using Master theorm

$$T(n) = 8T(n/2) + n^2$$

⇒ which is $O(n^3)$

⇒ we can improve that and make only 7 multiplication

$P = (A_{11} + A_{22})(B_{11} + B_{22})$
$Q = (A_{21} + A_{22})B_{11}$
$R = A_{11}(B_{12} - B_{22})$
$S = A_{22}(B_{21} - B_{11})$
$T = (A_{11} + A_{12})B_{22}$
$U = (A_{21} - A_{11})(B_{11} + B_{12})$
$V = (A_{12} - A_{22})(B_{21} + B_{22})$

and calculate $C_{11}, C_{12}, C_{21}, C_{22}$ as follows

Subject : / / موضوع الدرس :

- $C_{11} = P + S - T + V$
- $C_{12} = R + T$
- $C_{21} = Q + S$
- $C_{22} = P + R - Q + U$

⇒ Now reduce the number of multiplicat to 7

* as before addition and subtraction take $O(n^2)$

* Using Master theorm

$T(n) = 7T(n/2) + O(n^2)$

$\Rightarrow O(n^{\log_2 7}) \Rightarrow O(n^{2.81})$

- Using Strassens Matrix is not the optimal way in all-cases, as there are some cases where naive algorithm is better