

數位影像生成

翁振虔

Height Field 直接 Rendering 方法

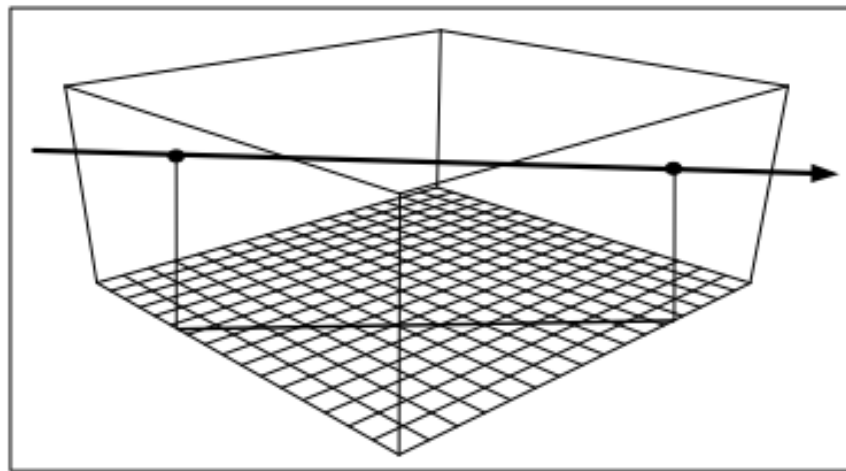
加速：三階段分離

一.BBOX 分離：

先找出 height field 最高與最低的高度，利用 BBOX 將其封裝起來，因此 ray 會先與 BBOX 做交集測試，成功再往下做。

二.二維分離：

利用 height Field 是由一張 2D 平面來描述何形狀的特性，首先將 ray 轉成 2 維(只看 x,y 方向)，由下圖所示，這樣可以有效的利用較少的資訊先分離不可能會交集的格子。

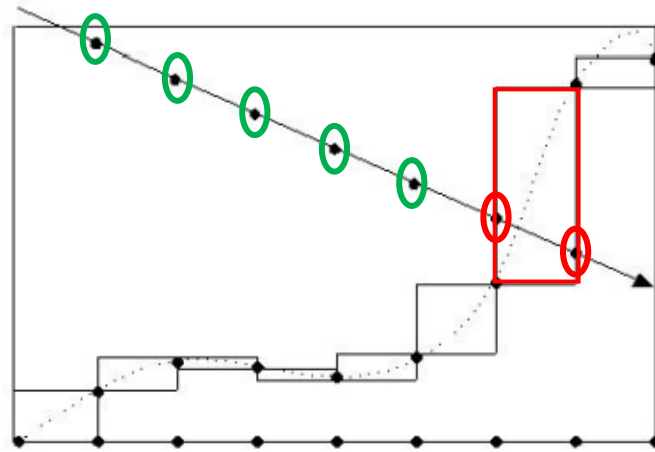


三.Voxel 分離：

因為每個間隔之間的平滑平面一定會被四周四點最高高度與最低高度所圍成的 voxel 包住(如下圖所示)

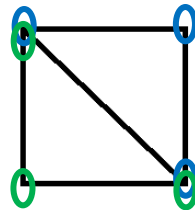


因此我先算出 ray 經過每隔邊線的高度(z 值)，如下圖綠色圈起點，如果 ray 與 height field 有交集，勢必通過 height field 各格所圍成的 voxel，若 ray 在邊線的高度介於 voxel 高度差之間才以可能有交點(如下圖“紅框”與“紅圓”所示)，這樣即可有效的分離出真正需要做“交點計算的 voxel”



交集：

- A. 將 height field 中每個方框拆成兩個三角形(如下圖藍色點 與 綠色點 分別代表不同的三角形)，對有交集的 Voxel 所包含的兩個三角形做交點測試。



- B. 利用以下公式：

$$O + tD = (1 - u - v)V_0 + uV_1 + vV_2$$

$$u, v \geq 0 \text{ and } u + v \leq 1$$

$$\begin{bmatrix} -D & V_1 - V_0 & V_2 - V_0 \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = O - V_0$$

$$\begin{bmatrix} -D & E_1 & E_2 \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = T$$

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{|-D, E_1, E_2|} \begin{bmatrix} |T, E_1, E_2| \\ |-D, T, E_2| \\ |-D, E_1, T| \end{bmatrix}$$

$$|A, B, C| = -(A \times C) \cdot B = -(C \times B) \cdot A$$

$$Q = T \times E_1 \quad P = D \times E_2$$

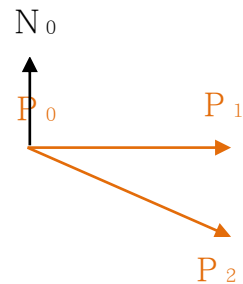
$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{P \cdot E_1} \begin{bmatrix} Q \cdot E_2 \\ P \cdot T \\ Q \cdot D \end{bmatrix}$$

可以算出 ray 與三角形的交點。

平滑修正：

三個頂點 **Normal** 的計算：

$$N_0 = (P_1 - P_0) \times (P_2 - P_0)$$



平面上任一頂點計算：

因為 $O + tD = (1 - u - v)V_0 + uV_1 + vV_2$ 所以光所交會的 **Normal** 會等於

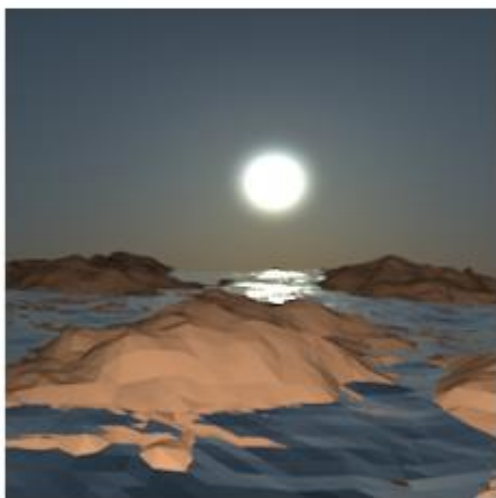
$$N_{ray} = (1 - u - v)N_0 + uN_1 + vN_2 \text{。}$$

$$dN = \frac{\partial N}{\partial u} du + \frac{\partial N}{\partial v} dv$$

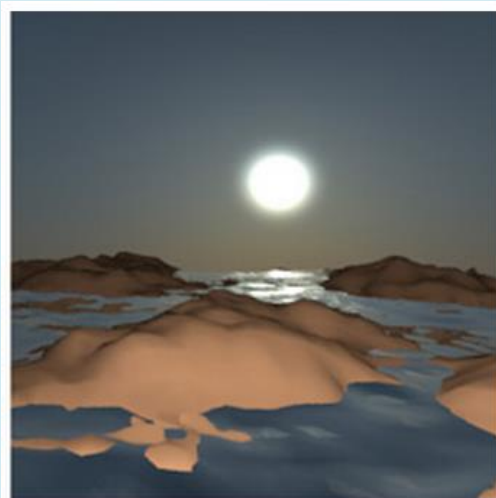
$$\begin{bmatrix} du_1 & dv_1 \\ du_2 & dv_2 \end{bmatrix} \begin{bmatrix} \frac{\partial N}{\partial u} \\ \frac{\partial N}{\partial v} \end{bmatrix} = \begin{bmatrix} dN_1 \\ dN_2 \end{bmatrix}$$

結果如下

原圖



使用後



Performance 比較

| | |
|------------|------------|
| 原程式 : 1.4s | 修改後 : 1.5s |
|------------|------------|

可能原因，因為做得是比較少應該會有比較好的 **performance**，但我對於 **memory allocate** 與一些節省運算時間的概念不是很懂，所以效果沒有比較好。