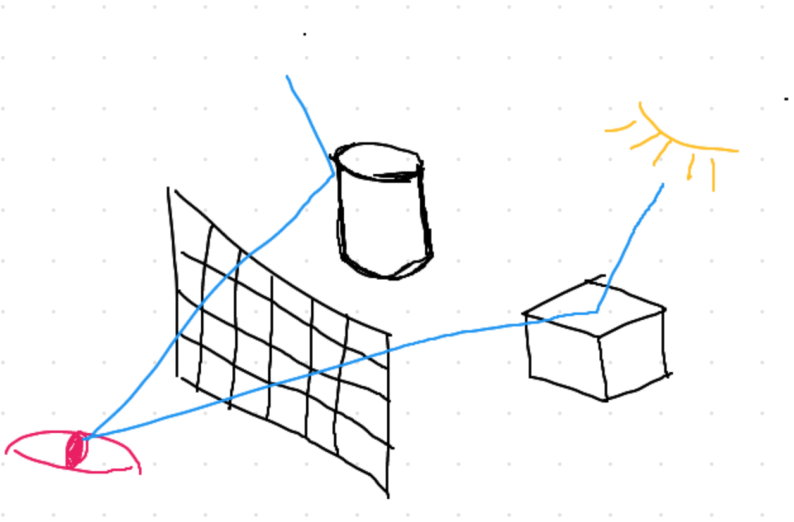
Ray Tracing on GPU

網媒所 碩一 楊子由 R06944013

Introduction

給定一個場景裡面的物件資訊坐標和一個相機，要畫出這相機所「拍」到的影像。Ray Tracing 就是其中一種做法。



這橫著一格一格的是像素，眼睛在後面，從眼睛連往任一像素都可以打出一條射線，跟個這條射線走，確認這條射線的顏色。如此，就可以確認像素的顏色，進而畫出整張影像。

Method and Algorithm

實作方式是實作 CPU 和 GPU 版本。GPU 程式使用 OpenCL 。

Ray Trcaing

```
Color rayTracing(ray, 反射層數)
    假如反射層數 > 4 ，回傳 黑色
    Cube = 取得射線相交方塊(ray)
    If (Cube != NULL)
        Color ret;
        For j = 1 ~ 4
            ret += rayTracing(反射 ray, 反射層數+1)
        End For
        Return ret/4*(Cube 顏色)
```

```
    End If
End Function
```

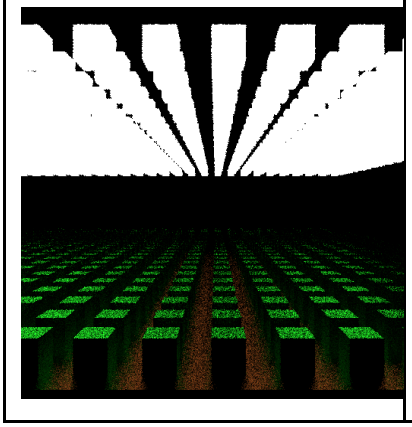
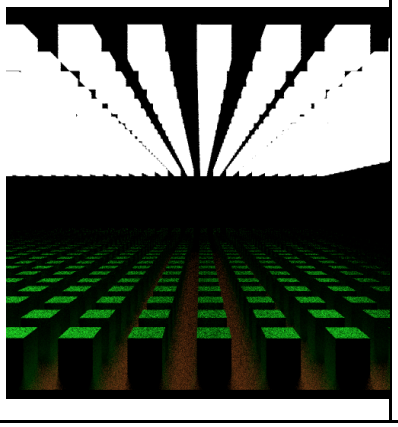
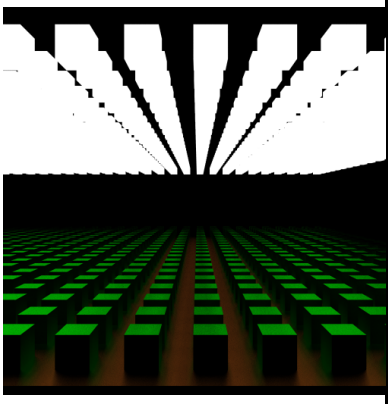
Result

CPU、GPU Information

- CPU
  - CPU(s): 40
  - Thread(s) per core: 2
  - Core(s) per socket: 10
  - Socket(s): 2
- GPU
  - Tesla K80

Render

Sample數是指每個像素打出幾條Ray，越多 samples 會讓畫面越乾淨。

1 sample	4 samples	256 samples
		

CPU Threading

Samples\ Threading	1	2	4	8	16
1	1.42 (1)	0.82 (1.7)	0.44 (3.2)	0.25 (5.7)	0.21 (6.8)
4	5.28 (1)	2.74 (1.9)	2.08 (2.5)	1.11 (4.8)	0.62 (8.5)

9	11.95 (1)	6.27 (1.9)	3.45 (3.5)	1.81 (6.6)	1.29 (9.3)
16	21.00 (1)	11.41 (1.8)	5.96 (3.5)	4.28 (4.9)	2.22 (9.5)
25	32.46 (1)	18.50 (1.8)	9.64 (3.4)	4.72 (6.9)	3.44 (9.4)

括弧裡面是和 1 sample 的速度倍率。

CPU vs GPU

Samples\Type	CPU(1 Thread)	GPU
1	1.42 (1)	0.23 (6.2)
4	5.28 (1)	0.85 (6.2)
16	21.00 (1)	3.24 (6.5)
256	348.10 (1)	51.21 (6.8)

括弧裡面是和 CPU 1 Thread 的速度倍率。

Samples\Type	CPU(16 Threads)	GPU
1	0.21 (1)	0.23 (0.9)
4	0.61 (1)	0.85 (0.7)
16	2.26 (1)	3.24 (0.7)
256	23.86 (1)	51.21 (0.5)

Conclusion

CPU Threading 測試部分之所以沒有按照倍率上升能是因為每個 Ray 的遞迴深度並不一樣。

GPU 並沒有顯著的加速，甚至比 CPU 16 Threading 慢。

Code

在 Github 上有 repo : [cube-gpu-raytracing]