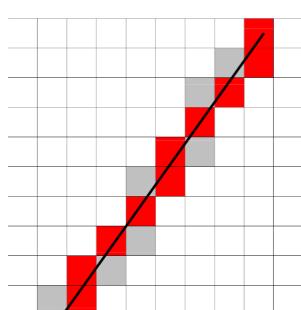


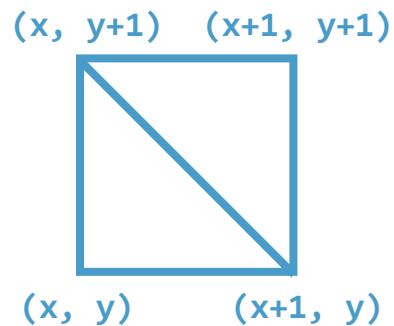
1. Heightfield-Ray Intersection 演算法

採用 DDA 的想法，當 ray 進來的時候先判斷是否有 intersect 這項物件的 Bounding Box，接著計算此 ray 第一個 Hit 在這個 box 上的點是誰，沿著這條線上（如下圖一）的點找。

首先先從第一個 Hit 的點來找，我們稱此點是 (x, y) ，找出這點的往右上延伸的格子，如圖二，這個方格的兩個三角形，分別是（下圖二）的上下兩個三角形，丟進如同 trianglemesh.cpp 的 `Intersect()` 及 `IntersectP()` 來計算，我是自己在 heightfield.cpp 裡建 `function triangleIntersect()` 及 `triangleIntersectP()` 分別對應兩個函式，就此計算和上下兩個三角形是否存在 Hit 的點，有的話就計算出來。若和三角形無交點時，會由 DDA 斜線上的點找到下一個格子，一樣分成兩個三角形去計算，直至找到 intersect 的三角形 return true 為止，或是此 ray 已超出 Bounding Box 就 return false 代表無交點。



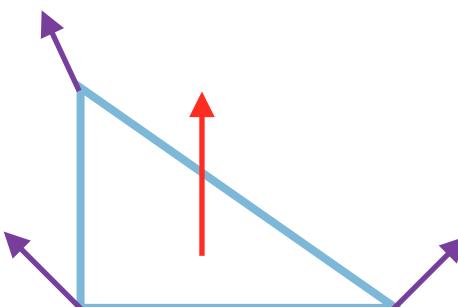
圖一：DDA



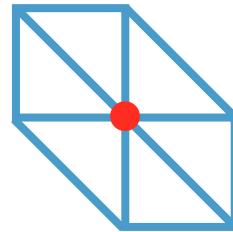
圖二：上下三角形

2. Smooth Shading 演算法

首先我會先 initial 所有點的法向量，如我的 `function computeNormal()`，但是當我們找到一個三角形並對其向量做 `Cross()` 時，只會找到三角形平面的向量（如下圖紅色箭頭），但我們的三角形的三個點的向量（如下圖紫色箭頭）才是我們要的，此時我們必須計算紫色箭頭的向量。

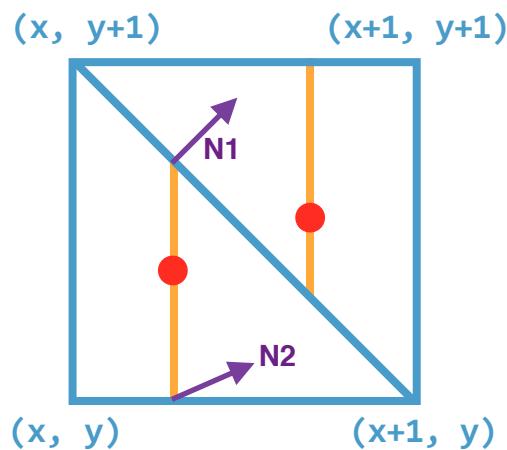


至於如何計算這點的向量，採的是找附近三角形的法向量來做平均，其中我也有試過找周圍的 2 個三角形或 4 個三角形的方法，但 2 個三角形的方法，可能因為估計得不夠準，所以最後出來的圖片會亮度差別不太明顯。而 4 個三角形的亮度分別會比較清楚，但陰影也會比較突兀。最後我採用 6 個周圍三角形（下圖藍色的 6 個三角形）來加權平均變成點（如下圖紅色點）的法向量代表。最後找到所有點的法向量。



我的 Smooth Shading 在 `function GetShadingGeometry()` 裡實現，主要是採用 Phong Shading 來達到平滑效果，前面提到我們已經算出所有點的 Normal 法向量，但是我們的 ray 打在三角形上面並不會是一個整數點，因此必須作 Phong Interpolation 來求出 Hit 在三角形那點的法向量。

我的方法是，現在我們有 hit 的這個點（如下圖紅色點），首先先判斷這個點在格子上面的三角形還是下面的三角形，再取出那個三角形三個頂點的三個 Normal，此時先兩兩內插得到兩個 Normal（如下圖紫色的 N1 及 N2），再針對橘色的線上做內插得到紅色點的法向量，再把這個向量拿去做 Shading，即可得到 Smooth Shading 的效果。



3. 與 PBRT Default Heightfield 在 Performance 上的比較

此 Performance 計算是由指令

```
# time ./pbrt ./scene.pbrt
```

所產生的 real 時間的數據。

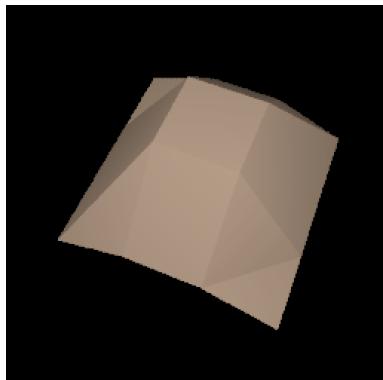
Scene	Default	Without Phong Shading	With Phong Shading
hftest	0.146s	0.152s	0.158s
texture	0.543s	2.153s	2.165s
landsea-0	0.980s	3.685s	3.730s
landsea-1	1.004s	5.646s	5.671s
landsea-2	1.023s	3.089s	3.100s
landsea-big	6.395s	9m 7.725s	8m 23.524s

由表格的內容，我們可以知道 Default 的速度非常快，在小的 hftest.pbrt 還看不出來，但較大的場景就可以看得出差異。Default 和自己寫的『Without Phong Shading』時間差得非常多，但得到的結果卻是相像的。

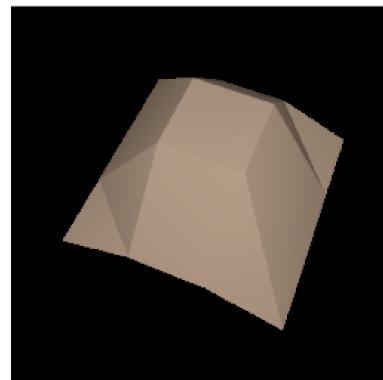
而『Without Phong Shading』和『With Phong Shading』兩種版本花費時間上就相去不遠，原因是因為在 Trace ray 找三角形，並找有沒有交點時都花了很多時間，真正Phong Interpolation 的時間並不會多花費多久。

4. 所有結果圖 default[6] , your[12]

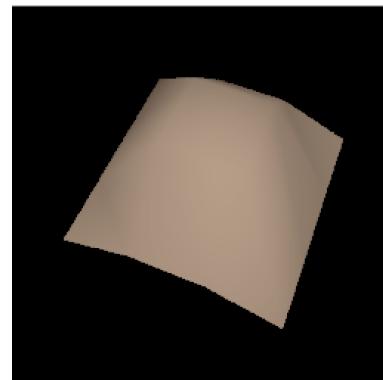
● hftest



Default
0.146s

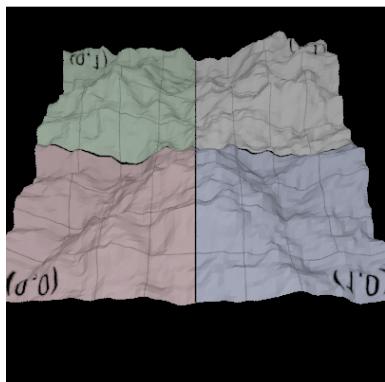


Without Phong Shading
0.152s

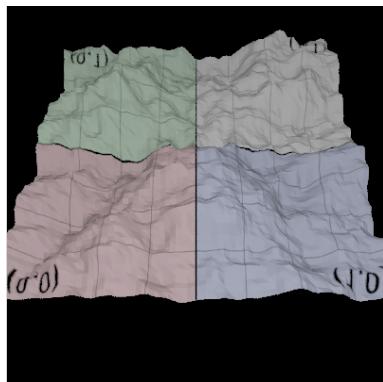


With Phong Shading
0.158s

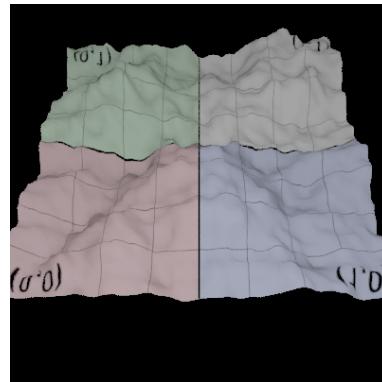
- **texture**



Default
0.543s

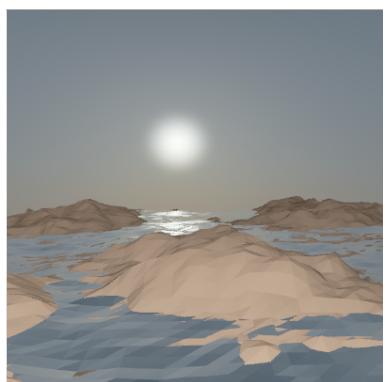


Without Phong Shading
2.153s

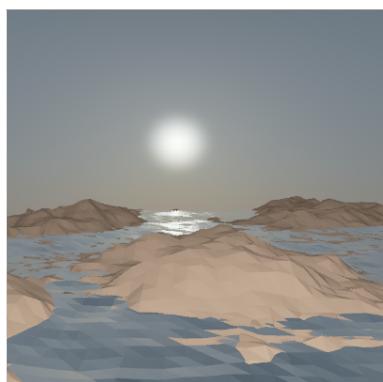


With Phong Shading
2.165s

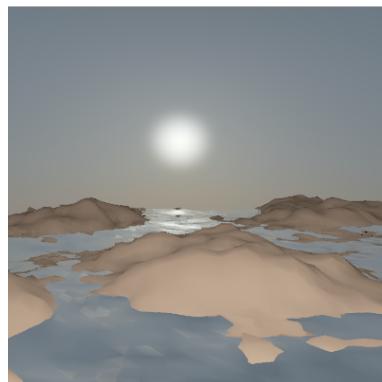
- **landsea-0**



Default
0.980s

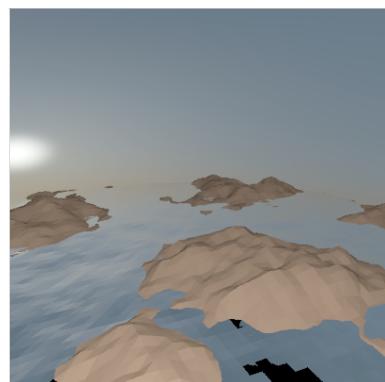


Without Phong Shading
3.685s

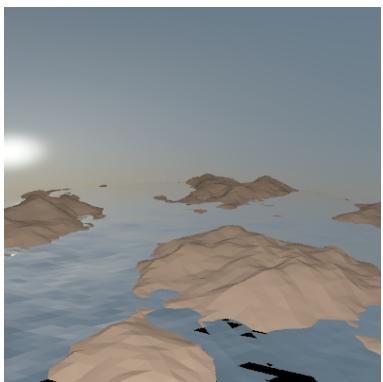


With Phong Shading
3.730s

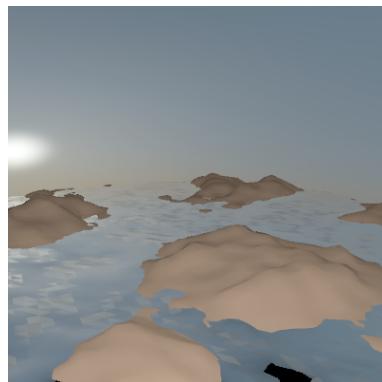
- **landsea-1**



Default
1.004s

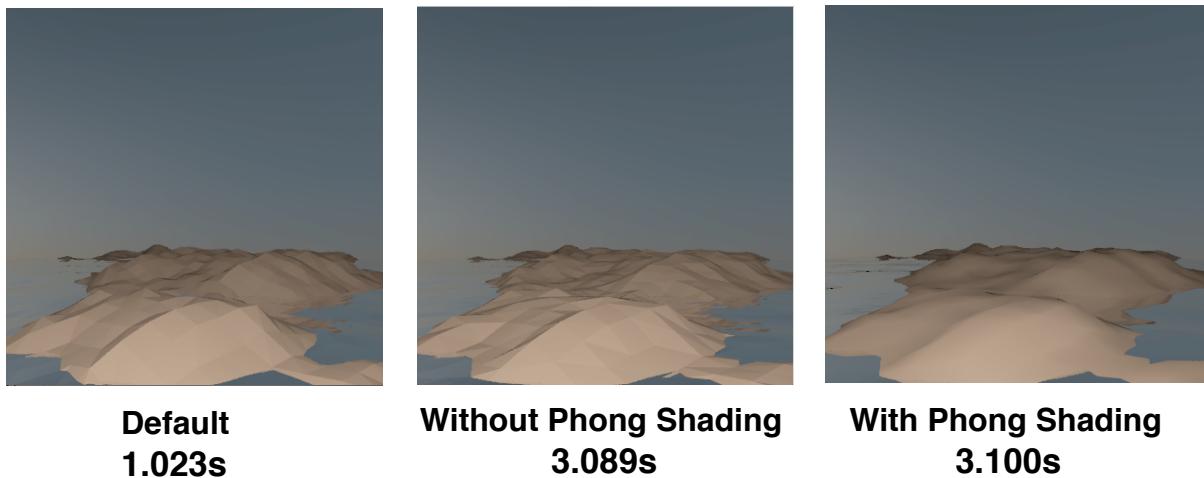


Without Phong Shading
5.646s

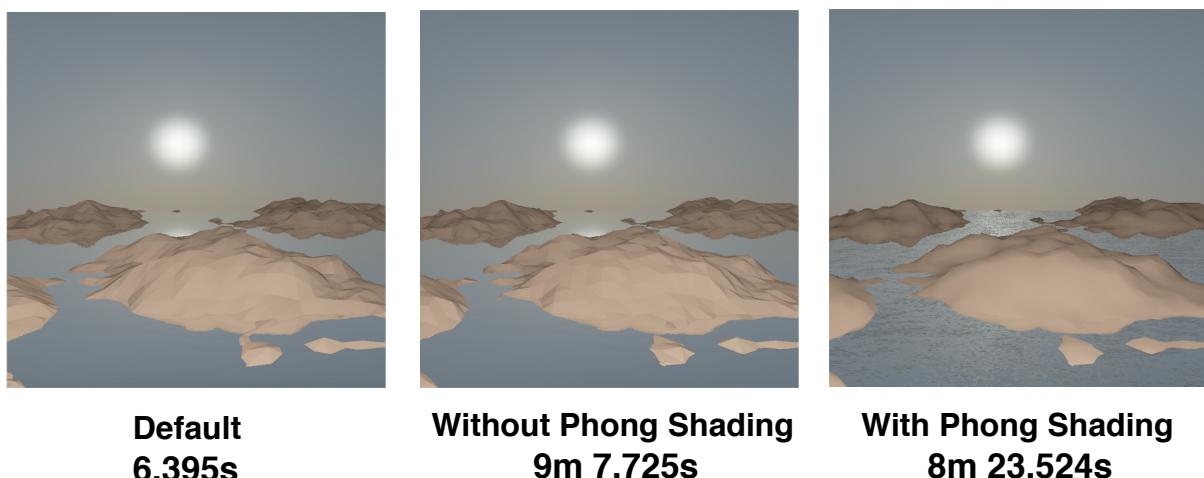


With Phong Shading
5.671s

- **landsea-2**



- **landsea-big**



5. 執行環境(Windows/Linux/Mac.OS)及配置

環境：Mac OS X

執行方面，採用兩種方式都有

1. Xcode 來執行得到結果
2. Linux 上編譯來執行得到結果