


Rendering HW1

資工所 - R04922078 - 吳德彥

Heightfield - ray intersection 演算法:

1. 先判斷線與Bounding Box的交點P0
2. 從P0開始Traversal
3. 依照2D DDA的Setup以及Traversal方法找出每個經過的Grid(程式碼參考grid.cpp)
4. 每個經過的Grid切成兩個三角形，如圖：
5. 檢查Ray是否與這兩個三角形有交集（參考TriangleMesh.cpp）
6. 如果有交點，得出DifferentialGeometry(參考TriangleMesh.cpp)




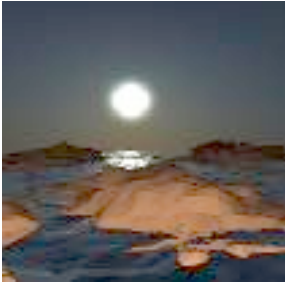
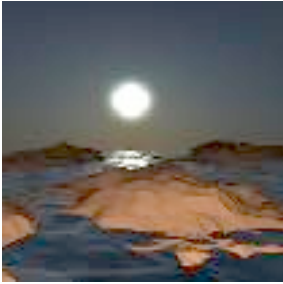
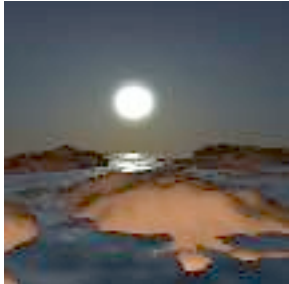
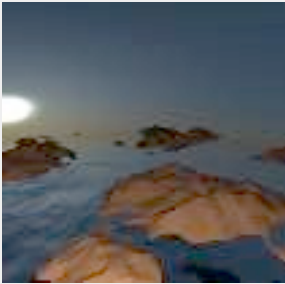
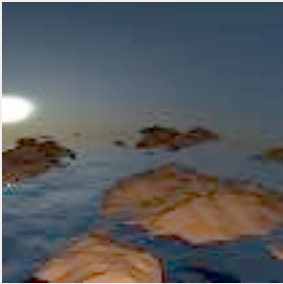

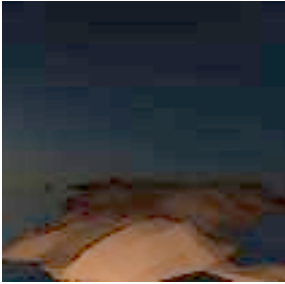
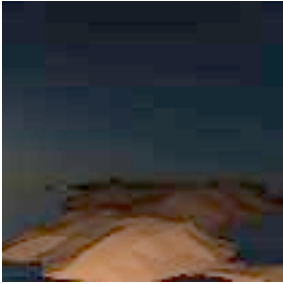
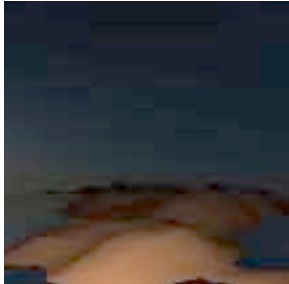
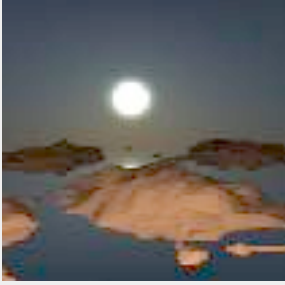
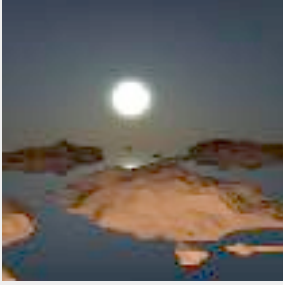

smooth shading 演算法:

1. 先在Constructor預建構好每一點的Normal
2. 每一點的Normal方法是參考上課教的方法($n = (dp/dx) \times (dp/dy)$)
3. 在GetShadingGeometry裡，得到DifferentialGeometry的U,V
4. 依照U,V算出是在哪個Grid裡的三角形裡面
5. 依據三角形的三點進行雙線性內差得到HitNormal
6. 得到HitNormal後，依據之前TriangleMesh的differential u,v，算出differential normal
7. 最後，依據HitNormal求出點的平面du, dv
8. 得出Shading DifferentialGeometry

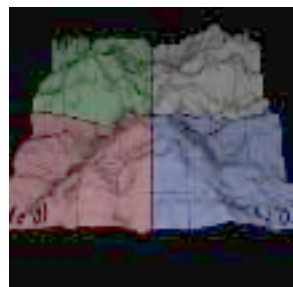
與 PBRT default heightfield 在 performance 上的比較:

	Default Heightfield	Heightfield2(without shading)	Heightfield2(shading)
hftest.pbrt	0s	0.1s	0.1s
landsea-0.pbrt	0.3s	2.1s	1.9s
landsea-1.pbrt	0.3s	2.7s	2.7s
landsea-2.pbrt	0.3s	1.5s	1.6s
landsea-big.pbrt	0.6s	250.6s	256.3s
texture.pbrt	0.2s	1.1s	1.1s

所有結果圖：

	Default Heightfield	Heightfield2(without shading)	Heightfield2(shading)
hftest.pbrt			
landsea-0.pbrt			
landsea-1.pbrt			
landsea-2.pbrt			
landsea-big.pbrt			

texture.pbrt



執行環境(Mac.OS)及配置:

Core: 8 cores, 2.2Hz, I7

Memory: 16GB

任何其他細節或者發現皆可 (e.g. 加速方法):