

# 數位影像生成

翁振虔

## Median Cut Environment Light

根據 Paper 所述將 Environment Image 切割成我們給定的 Light 數量，每個 Light 的光亮度都相同，然後再對物件做 rendering 的動作。



### 步驟 1

從 pbrt 檔案輸入 nSample 來決定 Environment Light 的數量(切割成幾個小塊)，創立一個新的 class 叫做 MedianCutEnvironment，在 Constructure 中將 Image 轉成 GrayImage，將各 Pixel 的 Intensity 做成 Cumulative Table 加快切割時個運算，並儲存成一個一個小區塊。

各區塊中記錄著 Centroid 的位置及整個區域的 Intensity。

因為此圖是貼在球體上，因此在最上方與最下方是石上有被拉伸，因此在算 Intensity 時必須成上  $\sin(\theta)$  來修正拉伸的 Image。

## 步驟 2

改 Sample\_L:

因為此 Environment Image 是當作貼在球體上做 rendering，因此將各切割區塊所儲存的 Centroid 位置轉成 Light 給的方向  $w_i$ 。

因為 pbrt 的架構，他使用的 monte carlo 需要知道此 render 的機率，因為選到此光源的機率為  $1/nSample$ ，所以不需額外使用 cdf 等方式去估計我們的 probability。

最後給 Spectrum，因為我們利用 Centroid 代表著這整塊區域的所有 Intensity，因此我們給的光是整塊區域 rgb 個別的總和。

## 結果

### Grace\_Latlong

Infinite Light



4 Samples(15.6s)



16 Samples(16.5s)

Median Cut Environment Light



4 Samples(2.3s)



16 Samples(5.2s)



64 Samples(62.9s)



64 Samples(18.2s)



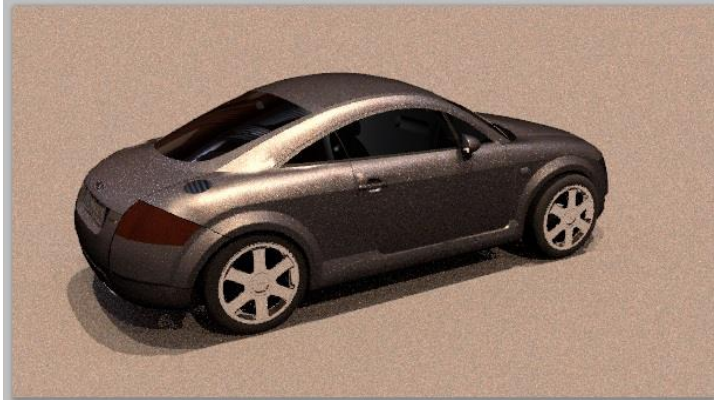
256Samples(251s)



256Samples(77.5s)

## Grace\_new\_Latlong

Infinite Light



4 Samples(6.1s)

Median Cut Environment Light



4 Samples(2.3s)



16 Samples(19s)



16 Samples(5.1s)



64 Samples(70s)



64 Samples(18.2s)



256Samples(310.4s)



256Samples(77.5s)

Spectrum 我有多加一個函式，所以除了增加新 Class 所需的東西外要把  
Spectrum.h 加入