

國立交通大學

資訊科學系

碩士論文

互動式全域照明 – 可變動的光源位置



Interactive Global Illumination: varying positions of
the light source

研究生：林駿豪

指導教授：李嘉晃 教授

中華民國九十四年六月

互動式全域照明 - 可變動的光源位置

Interactive Global Illumination : varying positions of
the light source

研 究 生：林駿豪

Student : Chun-Hao Lin

指 導 教 授：李嘉晃

Advisor : Chia-Hoang Lee

國 立 交 通 大 學

資 訊 科 學 系



A Thesis

Submitted to Department of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

In

Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

互動式全域照明 – 可變動的光源位置

學生：林駿豪

指導教授：李嘉晃 博士

國立交通大學電機資訊學院 資訊科學研究所碩士班

中文摘要

對於互動式全域照明(Interactive Global Illumination)計算，本篇論文提出結合光子圖(Photon Map)及多項式貼圖(Polynomial Texture Map)方法用來加速繪製場景(Scene)當其光源位置變動時的照度計算。為了結合上述兩種方法，我們提出以光子密度為單位的光子密度圖(Photon Density Map)及儲存方程式係數的多項式照度圖(Polynomial Illumination Map)來達成此目的。

在場景照明計算上，我們採用光子映射法(Photon Mapping)，由光源向場景發散光子及進行追蹤，並將光子記錄於光子圖中。其後，將場景進行空間等量分割，計算空間中取樣點的光子密度並儲存於光子密度圖中。我們取樣光源移動的位置空間，完成所有光子密度圖的收集，將光子密度變化和光源位置的關係，以三次方多項式逼近其結果，並將係數記錄於多項式照度圖中。在彩現(Rendering)場景上，我們採用反向光跡追蹤(Backward Ray Tracing)方式，將與場景中物體產生交點的光子密度經由多項式照度圖還原。如此在光源變動至新位置時，便不用重新發散光子，達到加速場景照度計算之目的。

關鍵字：光子映射法、多項式貼圖法、光子密度圖、多項式照度圖、照度估計、全域照明


Interactive Global Illumination : varying positions of the light source

Student : Chun-Hao Lin

Advisor : Prof. Chia-Hoang Lee

Department of Computer and Information Science
National Chiao Tung University

Abstract

The logo of National Chiao Tung University is a circular emblem with a gear-like border. Inside the circle, there are stylized Chinese characters and the year '1896'.

Although global illumination can offer realistic effect for scene rendering, most of the techniques require vast amount of computations as well as tremendous amount of time for rendering. Thus it is still not applicable for interactive or real time applications.

In this thesis, we propose a method which can compute global illumination for complex scenes very efficiently even when the position of light source is dynamically changing. The method uses a polynomial illumination map to approximate the shading of the scene for all possible positions of light source. The construction of the polynomial illumination map is based on the photon density map which is derived from traditional photon map.

Keywords : Photon Mapping 、 Polynomial Texture Maps 、 Photon Density Map 、 Polynomial Illumination Map 、 Illumination Estimation 、 Global Illumination

誌謝

本論文得以順利完成，首先要感謝的是指導教授李嘉晃老師，他在我的研究領域上給予充分的發展空間，讓我能有感興趣的課題上潛心研究，其嚴謹認真的研究態度及細心的指導，讓我在兩年的研究過程中受益良多。其次要感謝口試委員王勝德教授、李肇林教授以及莊榮宏教授的指正與建議，使得本論文能更加的充實完善。

接著，我要感謝實驗室的學長、同窗及學弟們對我的關心及幫助，尤其是黃明超、高鳴遠、蔡沛言、張佑銘、林建良、劉開國、許志高、江柏頡等幾位，豐富了我在實驗室中的生活，和你們在一起歡樂的日子我不會忘記。此外，我還要感謝我的朋友劉先能及陳雅萍，在我研究所生活上提供很大的幫助及建議，還有微奇科技孫智豪先生給予我在工作時間很大的彈性，讓我能順心的在學術上從事研究。

最後，我要感謝我的摯愛育芳及家人，謝謝你們的支持及照顧是我努力完成學業最大的原動力。謹以此篇論文獻給育芳、我的家人和初生女兒昕穎。

目錄

中文摘要.....	i
英文摘要.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	vii
第一章、緒論.....	1
1.1 研究動機.....	1
1.2 研究目標與構想.....	2
1.3 論文章節安排.....	2
第二章、相關研究.....	3
2.1 概述.....	3
2.2 光跡追蹤法.....	3
2.2.1 正向光跡追蹤 (Forward Ray Tracing).....	5
2.2.2 反向光跡追蹤 (Backward Ray Tracing).....	5
2.2.3 雙向光跡追蹤 (Bi-directional Ray Tracing).....	6
2.3 光子映射法 (Photon Mapping).....	7
2.3.1 建立光子圖.....	7
2.3.2 場景彩現.....	9
2.4 多項式貼圖法 (Polynomial Texture Maps).....	11
2.4.1 影像拍攝.....	11
2.4.2 多項式參數計算.....	12
2.4.3 影像重建.....	13
第三章、多項式照度圖.....	15
3.1 整體架構.....	15
3.2 前置處理.....	16
3.2.1 移動光源.....	16
3.2.2 發散光子.....	16
3.2.3 建立光子圖.....	17
3.2.4 建立光子密度圖.....	19
3.2.5 建立多項式照度圖.....	24
3.3 動態執行處理.....	27
3.3.1 重建光子密度圖.....	27
3.3.2 場景繪製.....	27
3.4 多照明光源應用.....	30
第四章、實作過程與結果討論.....	31

4.1	實驗說明.....	31
4.2	多項式全域照明計算結果.....	31
4.2.1	實驗一：單一照明光源場景測試.....	31
4.2.2	實驗二：多照明光源場景測試.....	35
4.2.3	三次方多項方程式近似結果.....	36
4.3	測試場景彩現結果.....	37
第五章	、結論.....	41
5.1	研究總結.....	41
5.2	未來工作.....	42
參考文獻	43



圖目錄

圖 1 正向光跡追蹤.....	5
圖 2 反向光跡追蹤.....	6
圖 3 雙向光跡追蹤.....	6
圖 4 不同的光源類型.....	7
圖 5 漫射表面反射光子情形.....	8
圖 6 場景中不同的物體反射光子的情形.....	9
圖 7 場景進行光子收集.....	9
圖 8 投影至平面光子收集.....	10
圖 9 光子映射法的場景彩現.....	11
圖 10 多項式貼圖法取像設備.....	12
圖 11 像素照度變化的收集.....	12
圖 12 系統流程架構.....	15
圖 13 建立光子圖流程.....	18
圖 14 取樣空間內進行光子密度的計算.....	19
圖 15 場景空間等量分割.....	20
圖 16 場景計算完成光子密度圖.....	20
圖 17 光子圖彩現的結果.....	21
圖 18 光子密度圖彩現的結果.....	21
圖 19 收集所有光源取樣位置的光子密度圖.....	22
圖 20 建立光子密度圖流程.....	23
圖 21 建立多項式照度圖流程.....	26
圖 22 還原計算光子密度圖.....	27
圖 23 計算交點的光子密度.....	28
圖 24 場景彩現流程.....	29
圖 25 多照明光源的計算.....	30
圖 26 程式實驗執行流程.....	33
圖 27 場景中不同的取樣cell的多項式結果圖.....	36
圖 28 寺廟場景上前方的觀察角度的照度變化情形.....	38
圖 29 寺廟場景側方向觀察角度的照度變化情形.....	38
圖 30 寺廟場景斜方向觀察角度的照度變化情形.....	38
圖 31 修道院場景前方的觀察角度的照度變化情形.....	40
圖 32 修道院場景斜方向觀察角度的照度變化情形.....	40
圖 33 修道院場景右方向觀察角度的照度變化情形.....	40

表目錄

表 1 程式開發環境.....	31
表 2 實驗測試環境.....	32
表 3 實驗測試場景.....	32
表 4 測試場景進行光源取樣及空間分割的數據.....	34
表 5 測試場景進行前處理計算及彩現階段所花費的時間.....	34
表 6 不同的彩現方式計算時間比較.....	35
表 7 數值顯示場景光源增加所造成的繪圖速度減慢程度.....	35



第一章、緒論

1.1 研究動機

電腦繪圖的真實感是電腦圖學研究領域所追求的重要目標之一，若要提高場景(Scene)繪圖的真實度，全域照明(Global Illumination)的計算便顯得十分的重要。然而大多數的技術均需要大量的計算，且彩現(Rendering)顯像相當費時，因此，在互動性的操作或是需要即時彩現時，便產生應用上的困難。

全域照明計算係指模擬場景中所有光源發散、反射及折射等現象，以產生真實的彩現品質為一重要關鍵。常見的計算方法包含路徑追蹤(Path Tracing)方式[JEN95] [LW93] [LW96] [VG94]、能量輻射(Raiosity)方式[HSA91] [DFD93] [JMLD99]、光粒子(Photon)方式[JEN96] [JEN97] [JEN01]等。其中，以光粒子為基礎的光子映射法(Photon Mapping) [JEN96]為一有效率的方法且可以提供非常真實的場景照度效果。然而，計算過程中需要發散大量的光子並且對每一個光子的路徑進行追蹤，是一項十分費時的計算，尤其每當場景產生變動時就要對光子散發重新計算，因此並不適合應用於需即時顯現的互動操作模式。在 Image-based rendering 領域中，提出對於場景光源改變進行有效率的彩現技術如[TSH97] [DHT_00] [EHY95] [MGW01]等。其中，多項式貼圖法(Polynomial Texture Maps) [MGW01]採用方程式計算的方式，對貼圖影像上的每一個像素以一個二次多項式來表示其光源位置變化後的照度值，允許在不同的光源方向下產生高品質照片般的貼圖影像，可以應用於快速的互動操作。

為了達到互動式的全域照明計算，近年之來研究有幾個方向：[WBS03]主要在多光源應用於複雜的場景。這個方法為首先估算出每個視野中最重要光源，在彩現時僅處理這些光源的路徑追蹤即可。[WKB_02]則使用平行處理加速計算互動式全域照明所需的大量計算。另外，[PDC_03] [WS03]著重在使用繪圖硬體架構加速光子映射法的計算。除了以上所提方法，我們試著找出一個新的方法，結合全

域照明中光子映射法之方法及Image-based rendering中多項式貼圖法之技術，使得場景中光源位置變動時不需重新發散光子即可取得更新後的場景照度，以符合互動式操作的需求。

1.2 研究目標與構想

本論文研究目標即為加速全域照明的計算以達到互動式的操作。為了達到這個目標，我們計畫結合光子映射法與多項式貼圖法兩項技術。前者應用於場景照度的計算；後者取代重新發散光子的計算。為了減少採用光子映射法所產生的大量光子，資料儲存的方式必須要進行轉換。我們提出預先計算光子密度的方法，將場景中大量光子分佈轉為單位空間的光子密度，這樣的光子密度圖便十分適合應用於多項式貼圖法中所提出的多項式計算。如此一來，光源位置變動所造成場景照度的變化皆可以被許多的方程式求得其結果，而這個結果即可還原成光子密度圖。

經由上述的轉換計算，場景中最初的許多光子圖(Photon Map)變成了單一的多項式照度圖(Polynomial Illumination Map)。當應用於在使用者互動的操作環境下，本研究方法可以在光源變動至新的位置時，快速得到場景照度變化之結果。

1.3 論文章節安排

本論文第一章為緒論，敘述研究動機、研究目標與構想。第二章將對本論文相關研究作說明及介紹。第三章針對本論文所提出的方法作說明，首先介紹整體架構，接著分別說明前置處理及動態執行處理的執行步驟及相關技術。第四章敘述實作過程及結果討論，我們將展現測試場景實作後的畫面及執行數據，並針對結果進行討論及分析。第五章則對本研究作結論並提出實作後的一些心得。最後說明本論文之貢獻及其未來工作。

第二章、相關研究

2.1 概述

提高彩現場景真實度的方法，光跡追蹤為一項基本且重要的技術。所謂光跡追蹤是指在所建立的 3D 場景中進行 2D 影像合成輸出的一種技術。這樣的路徑追蹤技巧可見於其後的改良式光跡追蹤、幅射法及光子映射法中，以下便分別描述與本論文相關的成像方式。

2.2 光跡追蹤法

光跡追蹤法最早是由 Turner Whitted 於 1980 年提出[TUR80]，這是第一個對於產生如相片般品質影像的演算法。其對於場景中的透明材質或鏡面反射的物體可以呈現良好的模擬效果。但由於是以單一路徑進行追蹤，對於陰影邊緣的成像過於銳利，且無法正確呈現具散射(Diffuse)特性的材質，另一個主要的問題是沒有提供間接照明的計算。對於這些限制，之後有其它研究改進的作法可以解決，如 [JOH84]對於陰影的改善、[GFR88]對於散射特性材質的成像等所提出的方法。

光跡追蹤採用遞迴式(Recursive)的演算法為[Wat00]：

ShootRay (ray structure)

intersection test

if ray intersects an object

 get normal at intersection point

 calculate local intensity (I_{local})

 decrement current depth of trace

if depth of trace > 0

 calculate and shoot the reflected ray

calculate and shoot the refracted ray

Calculate and shoot reflected ray elaborates as

if object is a reflecting object

calculate reflection vector and include in the ray structure

Ray Origin := intersection point

Attenuate the ray (multiply the current k_{rg} by its value at the previous invocation)

ShootRay(reflected ray structure)

if reflected ray intersects an object

combine colours ($k_{rg} I$) with I_{local}

Calculate and shoot refracted ray elaborates as

if object is a refracting object

if ray is entering object

accumulate refractive index

increment number of objects that the ray is currently inside

calculate refraction vector and include in refracted ray structure

else

de-accumulate refractive index

decrement number of objects that the ray is currently inside

calculate refraction vector and include in refracted ray structure

Ray Origin := intersection point

Attenuate ray (k_{tg})

if refracted ray intersects an object

combine colours ($k_{tg} I$) with I_{local}

where

k_{rg} is the global reflection coefficient

k_{tg} is the global transmitted coefficient

2.2.1 正向光跡追蹤 (Forward Ray Tracing)

正向光跡追蹤係指由光源出發，對場景發散出大量的光線進行路徑追蹤，並在每一個交點處進行色彩的計算，如圖 1 所示。這個觀念如同於一盞房間中的光源向整個房間內物體發散光子，但僅會有部份的光線最終會射入成像平面被觀察者所看見，這樣的計算量是十分龐大且沒有效率的，於是便有反向光跡追蹤方式的提出。

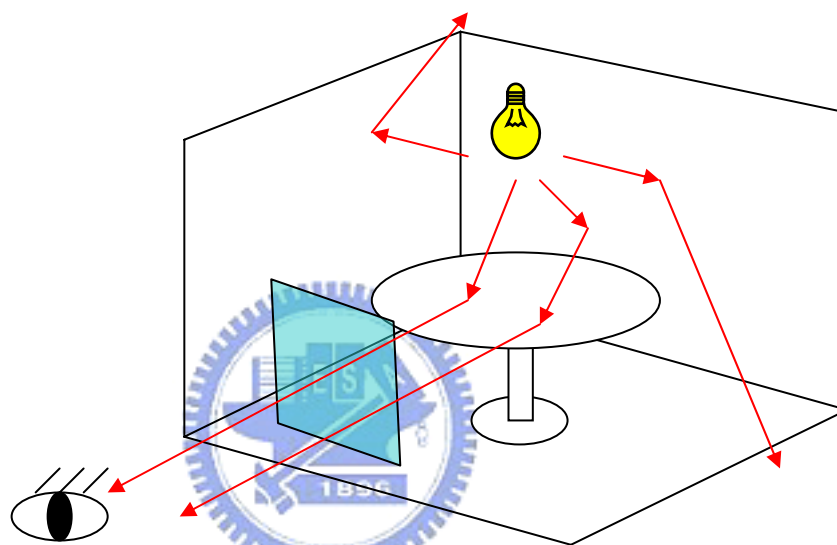


圖 1 正向光跡追蹤

2.2.2 反向光跡追蹤 (Backward Ray Tracing)

由於場景中發散的光線僅有部份會射入成像平面，若是只針對這些看得見的光線進行路徑追蹤，便可以大幅提昇執行的效率。成像平面上的每一個點，代表場景中光線最終的路徑，也是可被觀察者看到的光線，所以從成像平面上每一點出發，向場景中射出一條光線進行逆向的光跡追蹤，根據和場景中物體的交點進行反射及折射的計算，並在每一個交點處進行色彩的計算，如圖 2 所示。

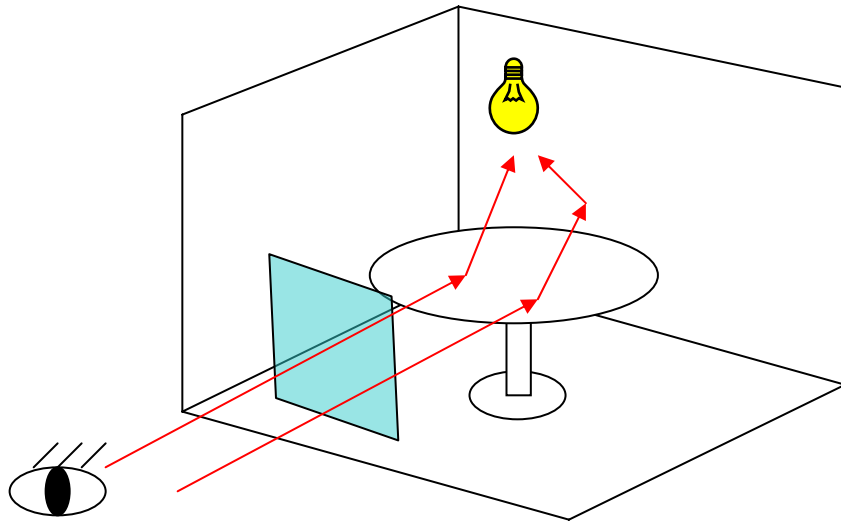


圖 2 反向光跡追蹤

2.2.3 雙向光跡追蹤 (Bi-directional Ray Tracing)

雙向光跡追蹤為一結合正向及反向光跡追蹤的方法，依據場景所需，場景的彩現可能是正向或反向的光跡追蹤，有時以雙向光跡追蹤對一個像素點進行色彩的計算，這樣的作法可求得效率及精確度上的平衡，如圖 3 所示。

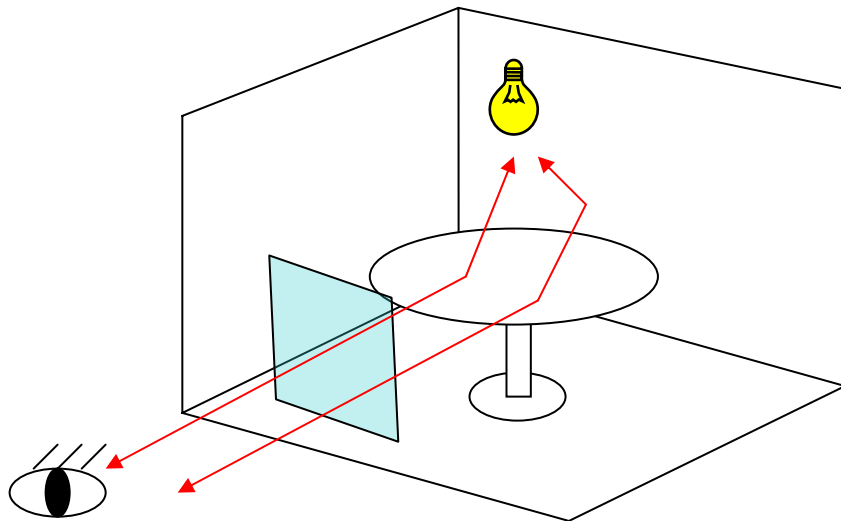


圖 3 雙向光跡追蹤

2.3 光子映射法 (Photon Mapping)

光子映射法[JEN96]為計算全域照明的方法之一，可視為一種雙向路徑追蹤的方式，其計算分為兩個階段：第一階段為前處理階段，進行光源的光子發散及儲存；第二階段為彩現成像，使用前一階段所建立的光子圖進行顯像。在第一階段建立的光子圖包含兩種：全域光子圖(Global Photon Map)及散焦光子圖(Caustics Photon Map)。當光源向場景發散光子時，這些光子的行進路徑追蹤過程可能會被物體表面吸收、反射或折射。所謂的吸收係指光子其落於表面的位置、方向及能量被儲存於 Map 中，此即為全域光子圖。而光子經過物體折射會形成散焦現象，將這些光子儲存於 Map 中，此即為散焦光子圖。第二階之彩現成像段採用反向路徑追蹤，由觀察者位置及視角向場景進行顯像，過程中將依據第一階段前處理所建立的光子圖進行照度估算。[JEN01]

2.3.1 建立光子圖

光子映射法的第一步驟為在場景中光源處發散大量光子，每一個光子皆帶有能量、位置及方向。發散光子的方式依照光源的類型而定，為了簡化光源發散光子的處理，常見的光源類型簡化表示為四種，顯示於圖4。

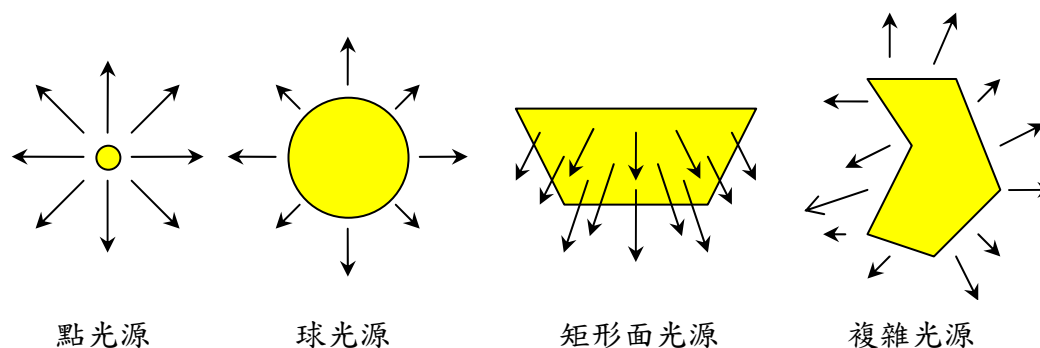


圖4 不同的光源類型

光子行進採用路徑追蹤方式，當光子打在場景中的物體表面時，採用俄羅斯輪盤法(Russian roulette)來決定這個光子為被表面吸收或反射。若光子為吸收，則將這個光子記錄在全域光子圖中；反之則依物體表面的雙向反射分佈函數(Bi-directional Reflectance Distribution Function, BRDF)決定其反射方向，並繼續光子的路徑追蹤，直至光子能量過小而終止。圖5顯示一物體其漫射表面反射光子的情形。

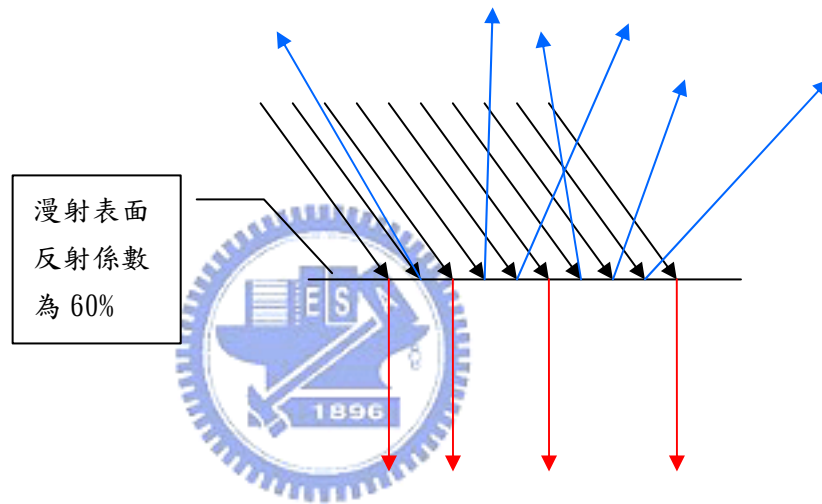


圖5 漫射表面反射光子情形

除了全域光子圖外，針對散焦的現象使用散焦光子圖來記錄這類的光子進行顯像。圖6顯示場景中不同物體反射光子的情形，圖7顯示場景進行光子收集後的情形，包含全域及散焦光子圖。

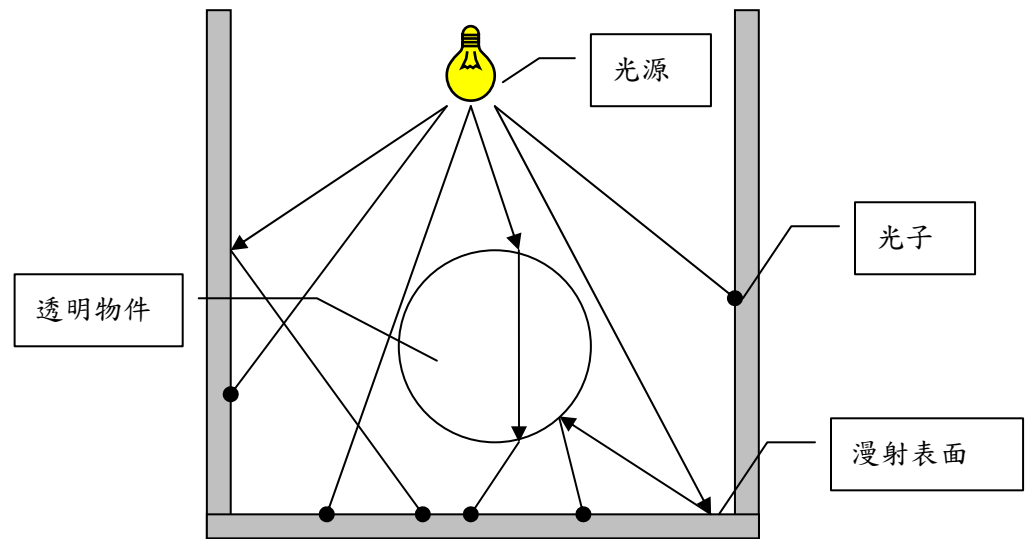


圖6 場景中不同的物體反射光子的情形

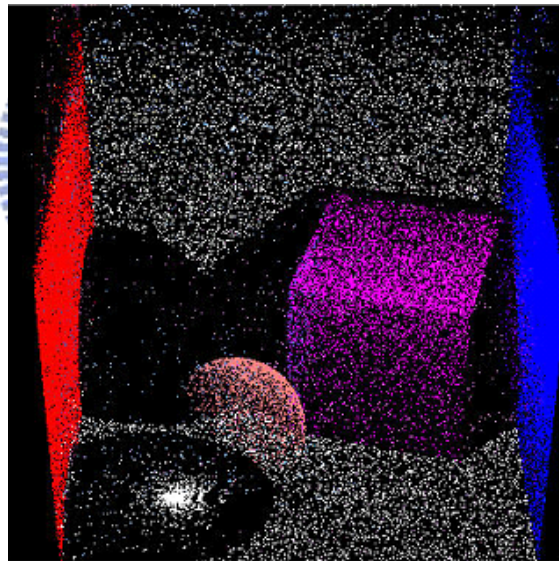


圖7 場景進行光子收集

2.3.2 場景彩現

場景成像由觀察者的視角向場景以蒙地卡羅路徑追蹤法(Monte Carlo Path Tracing)[Kaj86]進行繪製。當觀測射線與場景中物件發生相交時，這個相交點的亮度除了本身的自發性亮度外，也受其它間接照明的亮度影響，因此彩現成像公式可表示為式子(1)：

$$L_s(x, \psi_r) = L_e(x, \psi_r) + \int_{\Omega} f_r(x, \psi_i; \psi_r) L_i(x, \psi_i) \cos \theta_i d\omega_i \quad (1)$$

其中， L_e 表示自發性亮度， Ω 為入射的半球， x 表示相交點， f_r 為這個表面的BRDF， ψ_i 及 ψ_r 表示入射及反射角， L_i 表示進入的亮度。

由於在第一步驟我們已由光源向場景中發散光子，因此場景中表面的亮度資訊皆已存在光子圖中。利用這樣的結果，進入點 x 的亮度可由光子圖中取得，所以公式將可改寫為式子(2)：

$$L_r(x, \Psi_r) = \int_{\Omega} f_r(x, \Psi_r, \Psi_i) \frac{d^2\Phi_i(x, \Psi_i)}{dAd\omega_i} d\omega_i \quad (2)$$

由上述式子可知在以 x 為圓心的球體進行光子的收集並計算光子密度，由於光子皆落於物體表面上，可視為在平面上收集 N 個光子，因此，可將球體投影到以 x 為中心的平面圓面積，以圖8表示，所以公式將再改寫為式子(3)：

$$L_r(x, \Psi_r) \approx \sum_{p=1}^N f_r(x, \Psi_r, \Psi_{i,p}) \frac{\Delta\Phi_p(x, \Psi_{i,p})}{\pi r^2} \quad (3)$$

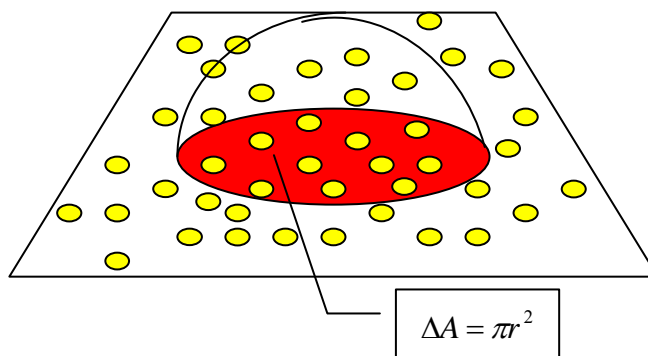


圖8 投影至平面光子收集

圖9表示以反向光跡追蹤對場景進行彩現，並進行光子資訊收集。

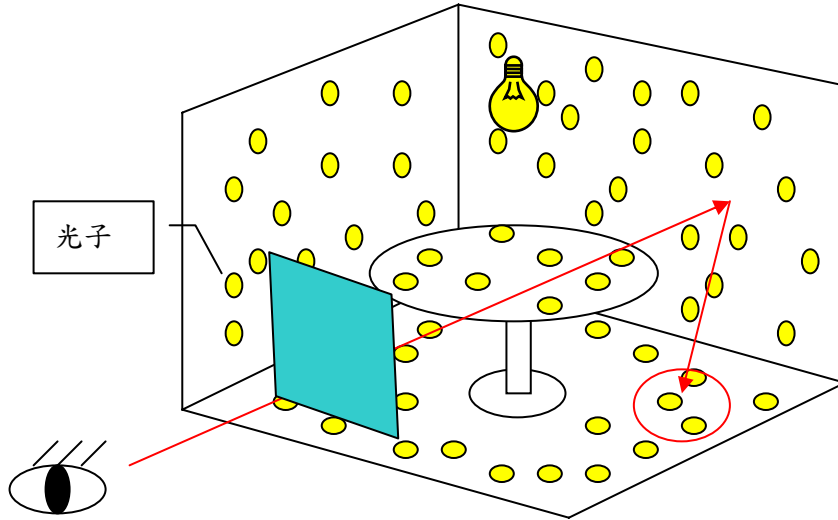


圖9 光子映射法的場景彩現

2.4 多項式貼圖法 (Polynomial Texture Maps)

在Image-based Rendering領域中，[MGW01]提出一個以影像為基礎的技術。該技術不需建立複雜的幾何模型或使用Bump Maps[BLI78]。其所要進行取樣的材質在不同光照角度的取像後，經過計算轉換成以多項式來代表影像中每一個像素照度，這樣的作法可以在有限的取樣下進行光照角度的內插計算，得到平順的轉換影像。

2.4.1 影像拍攝

首先建立一個取像攝影設備，如圖10。為了取得不同角度的光照影像，其裝置設計成一半球形空心罩，攝影機位於圓罩的頂部中心位置，這個裝置是以靜態的攝影機進行取像，並將不同光源架設於圓罩周邊的取樣點上。進行拍照時，依序點亮每一盞燈並進行取像，直到所有取樣點取像完成。

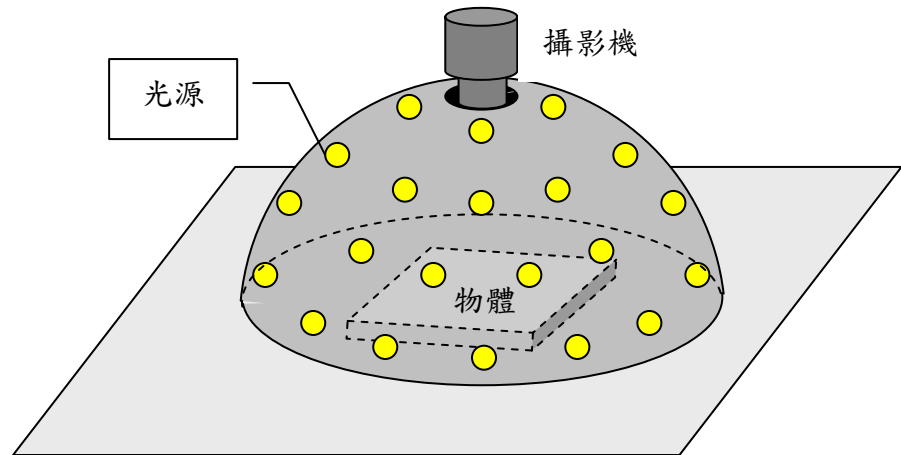


圖10 多項式貼圖法取像設備

2.4.2 多項式參數計算

經過每個光源點照射後的取像，可以得到許多不同照明角度的影像。將這些影像收集起來，在每張影像相同的座標位置，可視為該物件上此點在不同光照角度的亮度。我們以 (u,v) 來表示物件上的這一點，因此可以得到不同的亮度 $L_1(u,v)$ 、 $L_2(u,v)$ 、 $L_3(u,v) \dots L_n(u,v)$ ，如圖11所示。

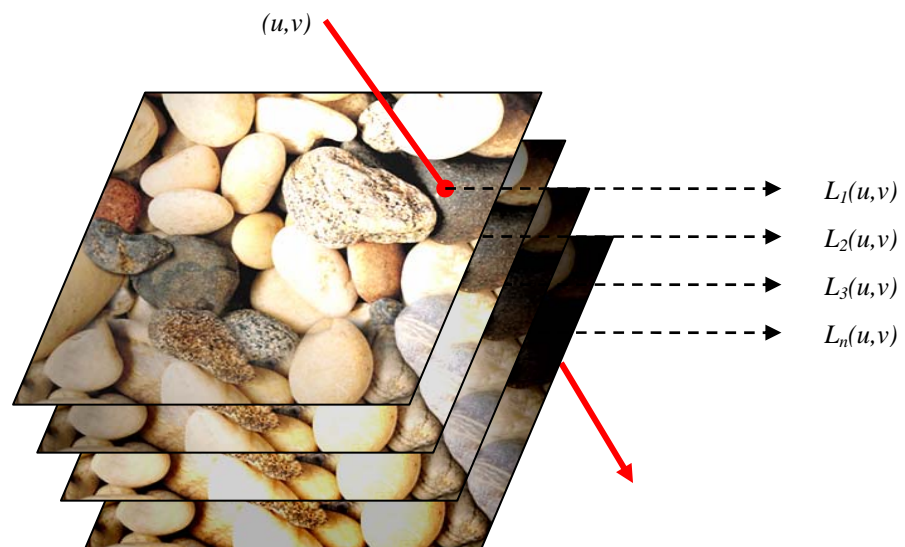


圖11 像素照度變化的收集

影像像素亮度和光源的關係式可以使用一個二次方程式來表示，如式子(4)所示。其中 l_u, l_v 為將光源投射轉換至貼圖平面的座標， u, v 為貼圖座標， $a_0 - a_5$ 為方程式係數。

$$L(u, v; l_u, l_v) = a_0 l_u^2 + a_1 l_v^2 + a_2 l_u l_v + a_3 l_u + a_4 l_v + a_5 \quad (4)$$

當我們有 n 個光源照射位置，每個像素就有 n 個亮度值 $L(u, v)$ ，將式子以矩陣方式表示，如式子(5)所示。

$$\begin{bmatrix} l_{u0}^2 & l_{v0}^2 & l_{u0}l_{v0} & l_{u0} & l_{v0} & 1 \\ l_{u1}^2 & l_{v1}^2 & l_{u1}l_{v1} & l_{u1} & l_{v1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{un-1}^2 & l_{vn-1}^2 & l_{un-1}l_{vn-1} & l_{un-1} & l_{vn-1} & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{bmatrix} = \begin{bmatrix} L_0 \\ L_1 \\ \vdots \\ L_{n-1} \end{bmatrix} \quad (5)$$

將 N 個方程式以 SVD (Singular Value Decomposition) 法求其逼近解以得到係數 $a_0 - a_5$ 的值，再將這些值隨著像素逐一記錄下來。在 Polynomial Texture Map 當中，儲存每一個像素的 RGB 值及方程式係數。

2.4.3 影像重建

多項式貼圖法係在每一個貼圖像素中使用六個多項式係數來重現像素的光影色彩。換句話說，每一個像素的光影色彩是由光照的方向及六個多項式係數計算而成的。我們可以將光源位置及式子(5)所求得的係數 $a_0 - a_5$ 代入式子(4)中求得該像素的亮度值 L ，此像素的色彩可由公式(6)求得。

$$\begin{aligned}R &= LR' \\G &= LG' \\B &= LB'\end{aligned}\tag{6}$$

由每一個像素的方程式係數來還原該照明光源位置下的色彩，在未來新的光源位置改變時便能很快的計算出其貼圖影像。



第三章、多項式照度圖

3.1 整體架構

本論文的研究方法包含了三個步驟：(一) 建立光子密度圖、(二) 計算多項式照度圖、(三) 影像彩現。前兩個步驟為前置處理，第三步驟為提供使用者互動操作的計算及顯像，系統的執行流程及步驟顯示於圖 12。在圖 12 左邊區塊為第一步驟，執行目的為建立光子密度圖。其中又分為四個程序：(一) 取樣光源位置、(二) 發散光子、(三) 建立光子圖、(四) 建立光子密度圖。這四個程序形成一個執行的循環，直到完成所有光源位置取樣為止。在圖 12 中間區塊則為第二步驟，執行目的為建立多項式照度圖，使用第一步驟所建立的資料進行計算，並提供下一步驟更新場景照度時使用。在圖 12 右邊區塊的為第三步驟，主要負責場景彩現顯像，其中又分為三個程序：(一) 移動光源、(二) 計算場景照度及 (三) 光跡追縱場景。

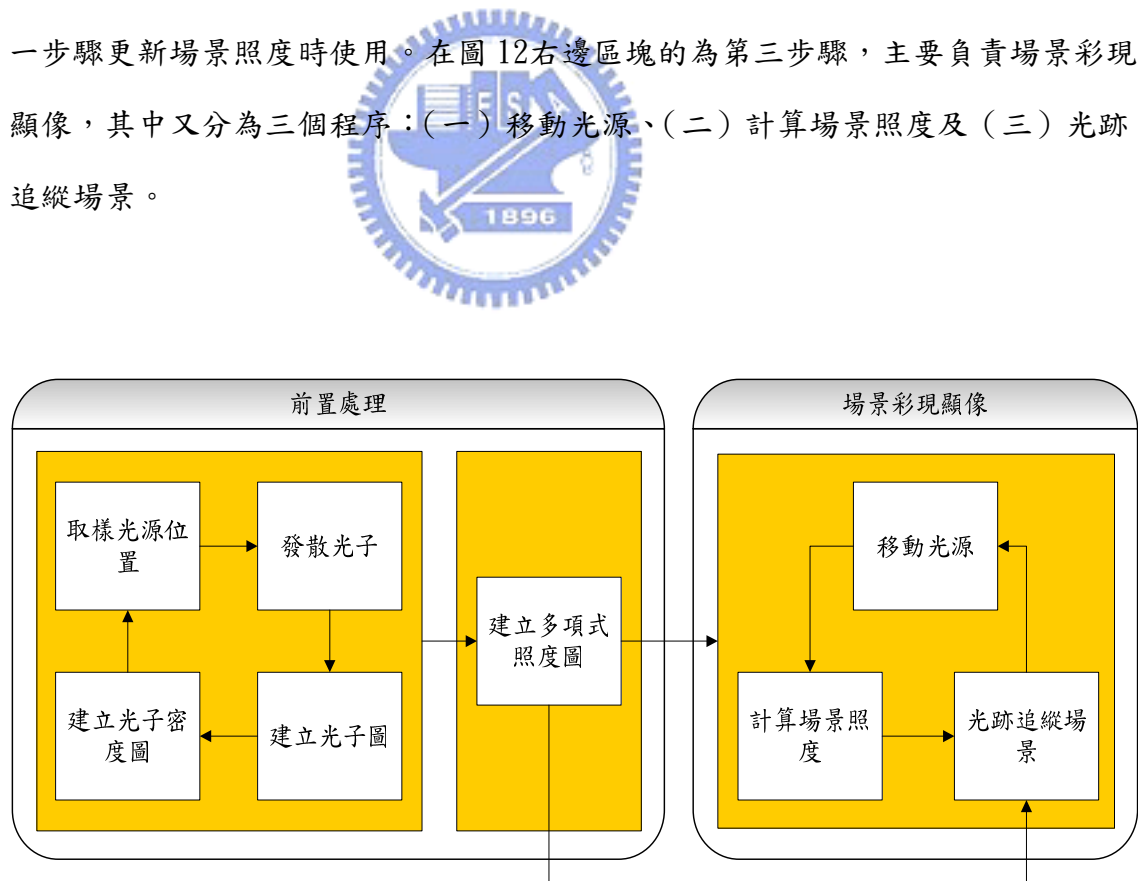


圖 12 系統流程架構

3.2 前置處理

前置處理主要目的為產生多項式照度圖供光源位置變動後計算場景照度使用，整個處理過程主要分為建立光子密度圖及計算多項式照度圖。以下先針對建立光子密度圖的步驟及方法作說明，最後將說明多項式照度圖的建立及計算方式。

3.2.1 移動光源

在場景中光源位置可移動的空間範圍內，設定位置的取樣。這些取樣點會在每一循環執行中將光源逐一的移到這些位置上，並依光源的類型進行下一程序的發散光子處理。

3.2.2 發散光子

發散光子係指在光源的位置依其不同類型向場景發射光子，並進行光子路徑追蹤及記錄。為了簡化光源發散光子的處理，我們僅進行點光源及矩形面光源的模擬。

光子發散的計算方式相同於光子映射法，每個光子發散是依光源的放射分佈函數(radiant distribution function)所決定，在我們所測試的場景中，使用了點光源及矩形面光源來進行場景的照明。光子的資料結構定義如下：

```
struct Photon
{
    float power;           // photon power
    float pos[3];         // photon position
    unsigned char theta, phi; // incoming direction
    short plane;          // splitting plane for kd-tree
};
```

在每一個光子資料結構內儲存了其能量(變數power)、位置(變數pos[3])及方向

(變數theta及phi)三個變數，光子透過放射分佈函數向場景射入，行進方式採路徑追蹤，當光子與場景內之物體表面產生碰撞，我們採用俄羅斯輪盤法的機率方式來決定其被表面吸收或反射。例如：一個物體表面其具有反射及漫射兩個屬性，以 ρ_d 表示反射係數， ρ_s 表示漫射係數，且 $\rho_d + \rho_s < 1$ 。我們設定一個均勻的亂數函數產生0至1的亂數以 z 表示。光子與表面碰撞後的處理表示如下：

```
if ( z <  $\rho_d$  )
    do reflect diffuse
else if ( z <  $\rho_d + \rho_s$  )
    do reflect specular
else
    do absorb
```

3.2.3 建立光子圖

若光子決定被表面吸收，則將這個光子資料存於全域光子圖中。全域光子圖可視為一個陣列的儲存空間，為了加速臨近光子的搜尋，全域光子圖使用KD-Tree結構來進行資料的儲存。反之，若光子決定被表面反射，則依物體表面的反射函數決定其反射方向，並繼續光子的路徑追蹤，直至光子能量過小而終止。當所有的光子都發散完畢即完成全域光子圖的建立。

圖13顯示建立光子圖的流程。建立光子圖執程序前，我們先將光源移動的位置空間分割為M個取樣點，光源發散P個光子。待本程序執行完成後，可得到場景的光子圖M張。

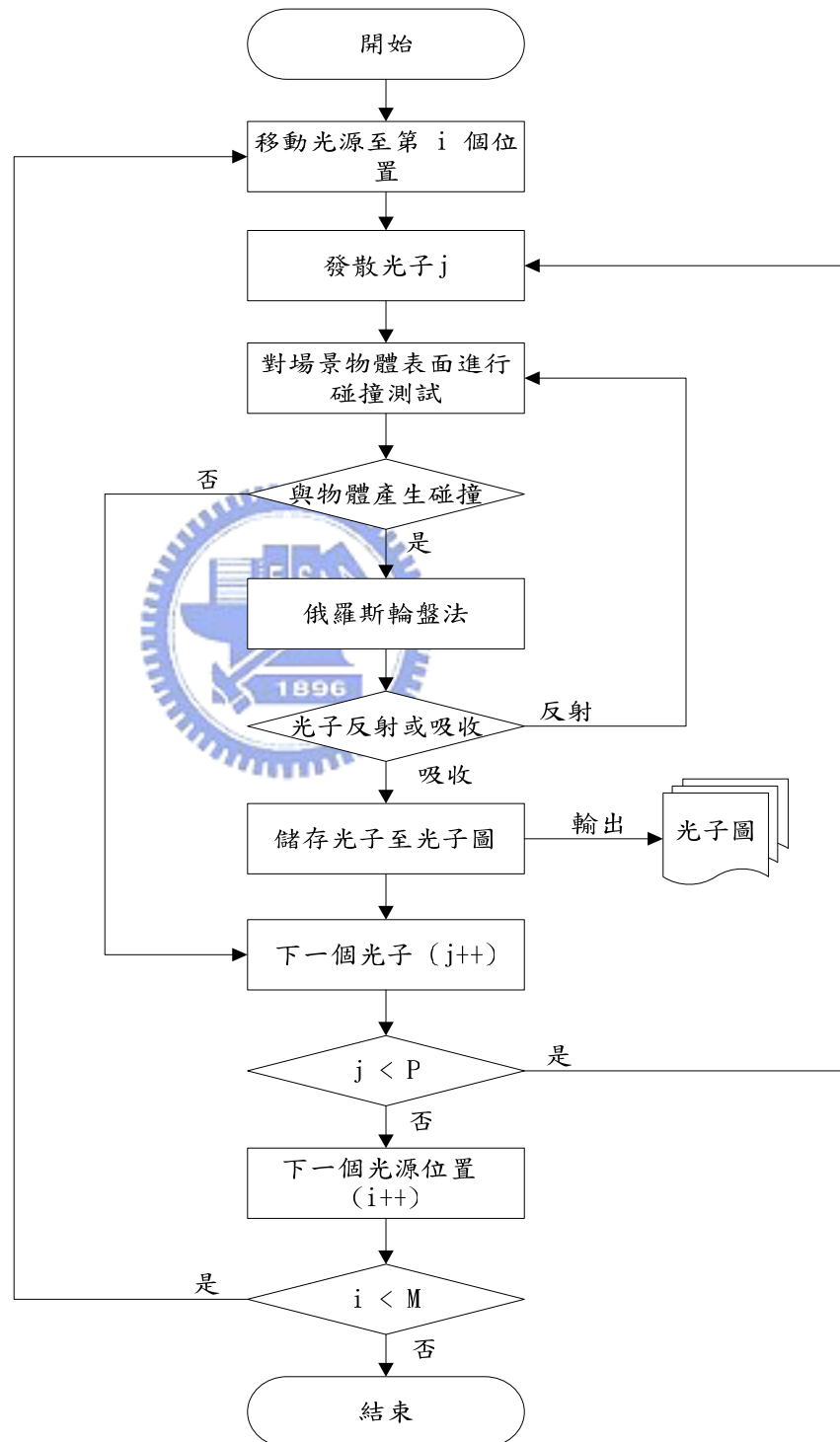


圖13 建立光子圖流程

3.2.4 建立光子密度圖

我們首先從光源取樣位置計算光子圖，其次將每一張光子圖轉換至光子密度圖，此光子密度圖相對於場景中不同光源位置的照度。

欲將光子圖轉換至光子密度圖，首先需將場景等間距的進行空間分割成為許多小的取樣空間，描述於圖14。然後使用每個小取樣空間的中心點，以固定的半徑進行光子數目的收集，在此空間中可計算出光子密度值並且儲存至一個光子密度圖中：

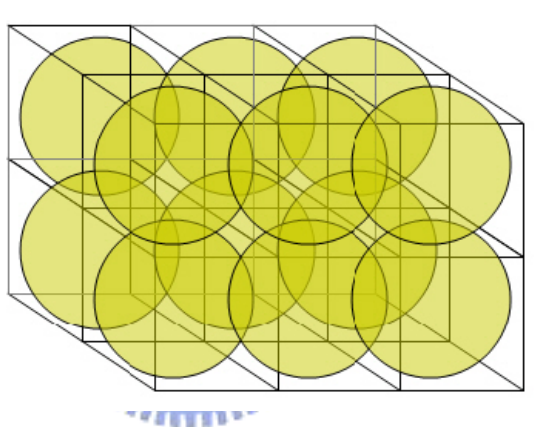


圖14 取樣空間內進行光子密度的計算

圖15 顯示寺廟場景經過空間分割後的情形，空間分割的範圍包含整個場景物件，其中紅色點為每個取樣空間的中心點。

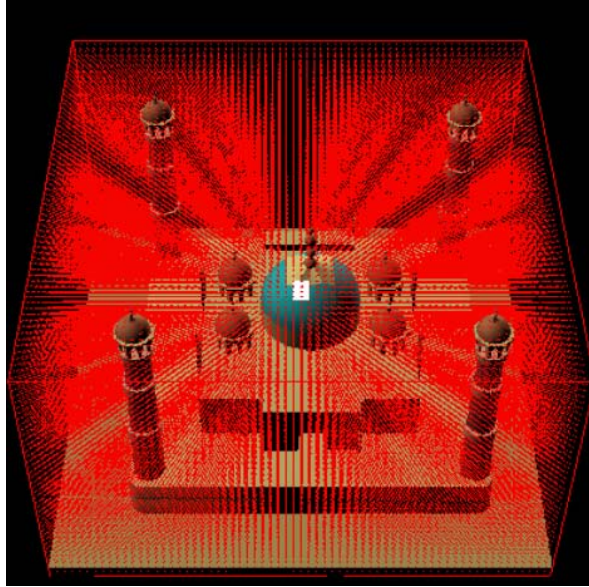


圖15 場景空間等量分割

圖16 顯示寺廟場景中一個光子圖經過轉換後成為光子密度圖的結果，其中取樣空間內之顏色愈接近黑色即表示愈低的光子密度；愈接近白色則為愈高的光子密度。光子密度的計算皆會落於物體的表面上。

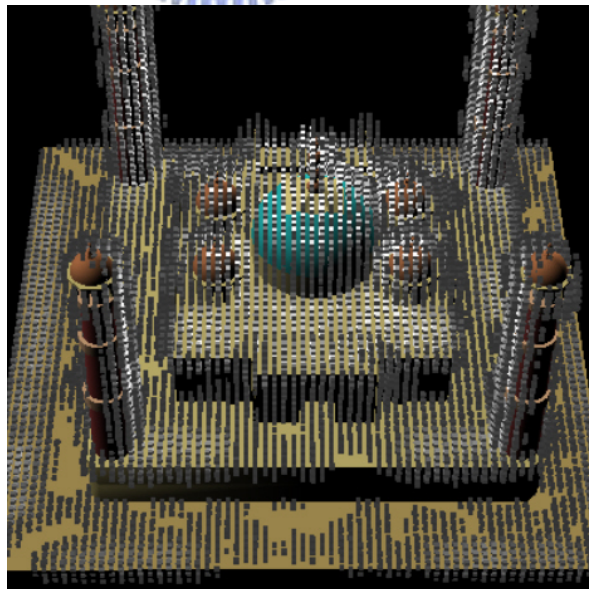


圖16 場景計算完成光子密度圖

圖17顯示經過發散光子程序後，以光子圖進行場景彩現之結果；圖18顯示將光子圖轉換為光子密度圖後，以光子密度圖進行場景彩現之結果。比較兩個結果，可觀察到後者顯示的影像較為平滑。



圖 17 光子圖彩現的結果



圖 18 光子密度圖彩現的結果

每一個光源其照明資訊的收集，均需要經過上述四個程序來完成。假設所有光源的移動位置取樣為 n ，在完成所有取樣點的計算後，會得到 n 組光子密度圖，表示於圖19。

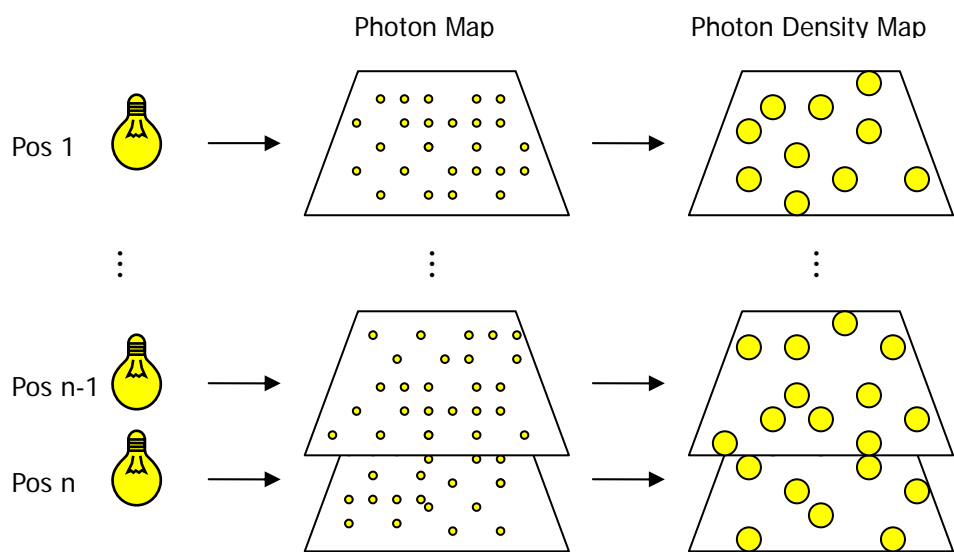


圖19 收集所有光源取樣位置的光子密度圖



圖20顯示建立光子密度圖的流程。建立光子密度圖執程序前，我們已將場景空間分割為N個取樣空間，並將光源移動的位置空間分割為M個取樣點。待本程序執行完成後，可得到場景的光子密度圖M張。

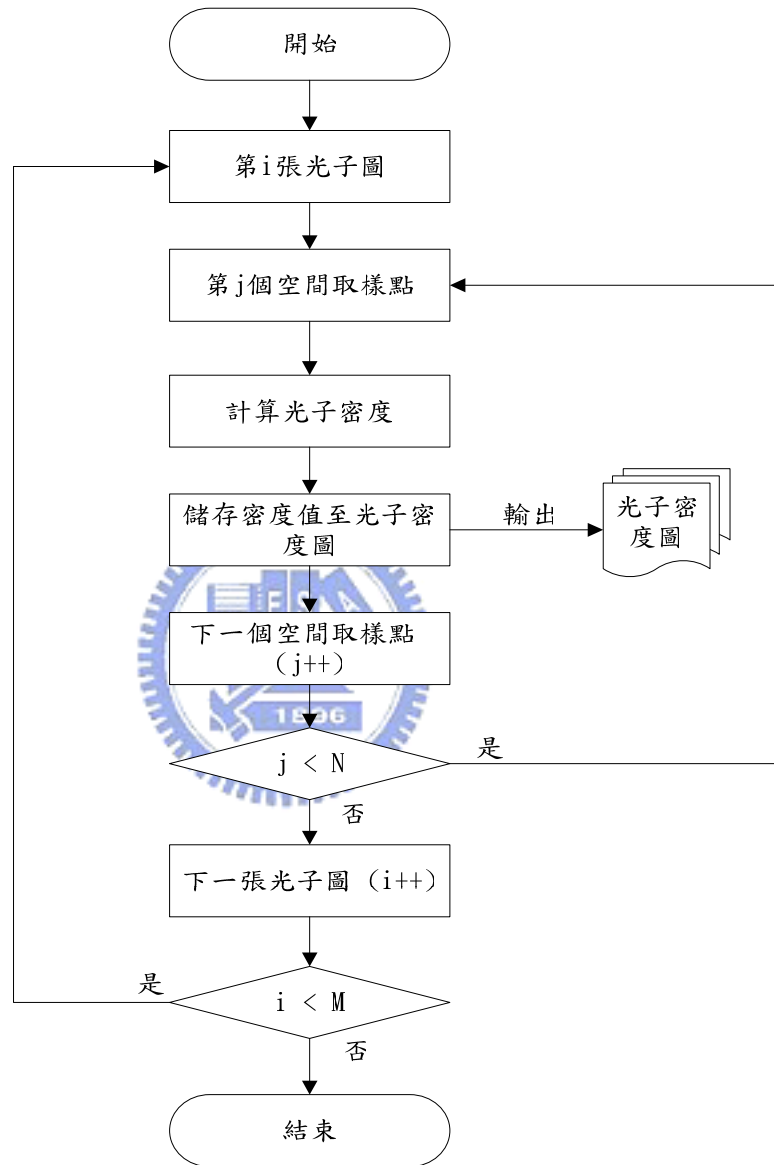


圖20 建立光子密度圖流程

3.2.5 建立多項式照度圖

多項式照度圖是用來建立光源位置及場景中照度的關係。在建立此關係之後，即可透過方程式的係數還原新光源位置的場景照度，藉由此方法可免除在光子映射法中若光源位置改變時就需重新發散光子、追蹤光子等繁複的計算過程。而且此方程式僅是簡單的乘法及加法計算，對於全域照明的計算非常有效率。

每一取樣點，使用一個三次方程式來表示光子密度和光源位置的關係。計算式描述於式子(7)。方程式中 $D = (D_x, D_y, D_z)$ 表示在取樣空間位置 (x, y, z) 的光子密度； $L = (l_1, l_2, l_3)$ 表示光源的位置。 a_0 - a_{19} 表示三次方程式的係數，透過這二十個係數，可計算出取樣空間在光源位置 L 時的光子密度 D 。

$$\begin{aligned}
 D = & a_0 l_1^2 l_2 + a_1 l_2^2 l_3 + a_2 l_3^2 l_1 + a_3 l_1 l_2^2 + a_4 l_2 l_3^2 \\
 & + a_5 l_3 l_1^2 + a_6 l_1 l_2 l_3 + a_7 l_1^3 + a_8 l_2^3 + a_9 l_3^3 \\
 & + a_{10} l_1^2 + a_{11} l_2^2 + a_{12} l_3^2 + a_{13} l_1 l_2 + a_{14} l_2 l_3 \\
 & + a_{15} l_3 l_1 + a_{16} l_1 + a_{17} l_2 + a_{18} l_3 + a_{19}
 \end{aligned}
 \tag{7}$$

使用相同於[MGW01]的方法，在矩陣式子(8)中的係數 $X = [a_0 \ \dots \ a_{19}]^T$ 將可用來計算每一個取樣空間的光子密度資料並儲存於光子密度圖中。 A_1 表示十個光源位置之三次項不同的組合， A_2 表示六個光源位置之二次項不同的組合， A_3 表示三個光源位置之線性項， 1 表示為皆為1的列。

$$[A_1 | A_2 | A_3 | \mathbf{1}] X = \underline{D}
 \tag{8}$$

其中

$$A_1 = \begin{bmatrix} l_{01}^2 l_{02} & l_{02}^2 l_{03} & l_{03}^2 l_{01} & l_{01} l_{02}^2 & l_{02} l_{03}^2 & l_{03} l_{01}^2 & l_{01} l_{02} l_{03} & l_{01}^3 & l_{02}^3 & l_{03}^3 \\ l_{11}^2 l_{12} & l_{12}^2 l_{13} & l_{13}^2 l_{11} & l_{11} l_{12}^2 & l_{12} l_{13}^2 & l_{13} l_{11}^2 & l_{11} l_{12} l_{13} & l_{11}^3 & l_{12}^3 & l_{13}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{k1}^2 l_{k2} & l_{k2}^2 l_{k3} & l_{k3}^2 l_{k1} & l_{k1} l_{k2}^2 & l_{k2} l_{k3}^2 & l_{k3} l_{k1}^2 & l_{k1} l_{k2} l_{k3} & l_{k1}^3 & l_{k2}^3 & l_{k3}^3 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} l_{01}^2 & l_{02}^2 & l_{03}^2 & l_{01} l_{02} & l_{02} l_{03} & l_{03} l_{01} \\ l_{11}^2 & l_{12}^2 & l_{13}^2 & l_{11} l_{12} & l_{12} l_{13} & l_{13} l_{11} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{k1}^2 & l_{k2}^2 & l_{k3}^2 & l_{k1} l_{k2} & l_{k2} l_{k3} & l_{k3} l_{k1} \end{bmatrix}$$

$$A_3 = \begin{bmatrix} l_{01} & l_{02} & l_{03} \\ l_{11} & l_{12} & l_{13} \\ \vdots & \vdots & \vdots \\ l_{k1} & l_{k2} & l_{k3} \end{bmatrix}$$

且 k 表示光源位置的取樣數目，

$$X = [a_0 \quad \cdots \quad a_{19}]^t$$

$$\underline{D} = [D_0 \quad \cdots \quad D_k]^t$$



圖21顯示建立多項式照度圖的流程。建立多項式照度圖執行程序前，我們已將場景空間分割為N個取樣空間，並將光源移動的位置空間分割為M個取樣點。待本程序執行完成後，可得到場景的多項式照度圖1張。

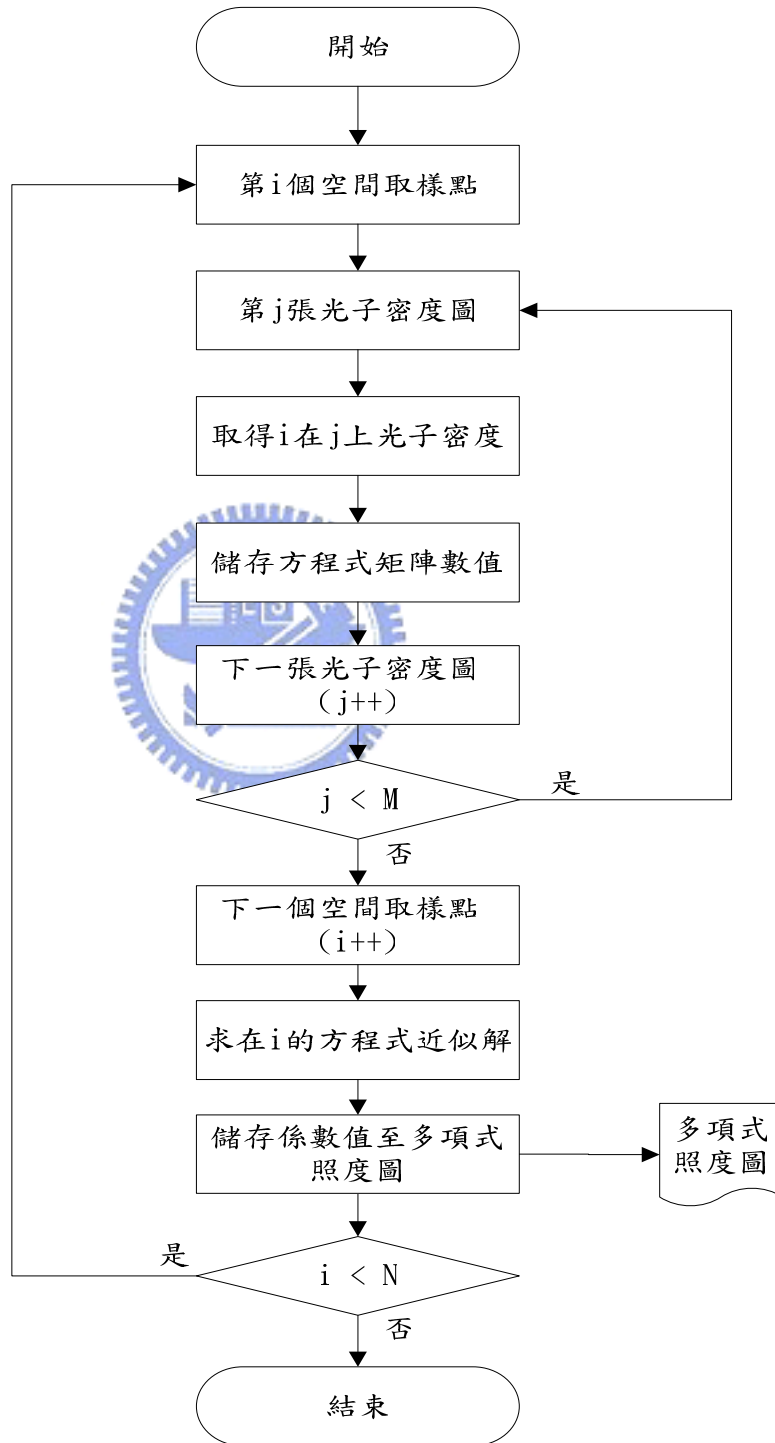


圖21 建立多項式照度圖流程

3.3 動態執行處理

3.3.1 重建光子密度圖

在光源移動位置範圍內，亦即不超出先前在光源位置取樣範圍，將新的位置代入多項式照度圖內每一組係數，求其相對應的光子密度值，即可得到新的光子密度圖，如圖 22所示。

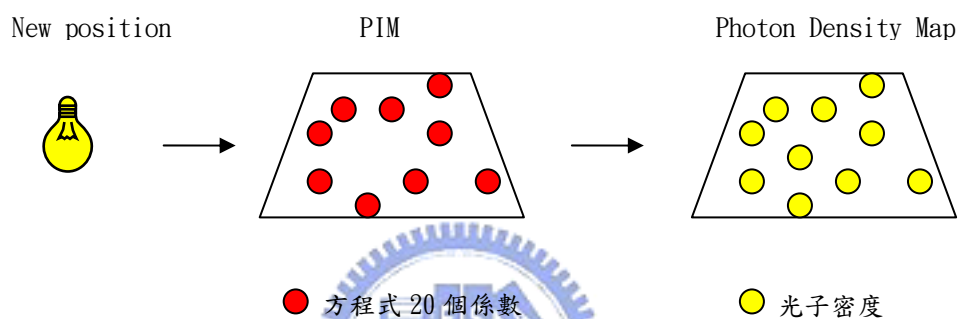


圖 22 還原計算光子密度圖

3.3.2 場景繪製

場景繪製係指以光子密度圖進行反向光跡追蹤的方式，將其所成的照度值進行場景明暗混色。根據觀察者的位置角度，我們以視角射線向場景進行路徑追蹤及彩現顯像。對每一條Ray計算其與表面相交點，以此相交點在光子密度圖中找出臨近的光子密度值，以距離為加權計算其對這個相交點的貢獻，該影響程度隨著距離而遞減，因此距離相交點近的供獻度較高，如圖 23所示。

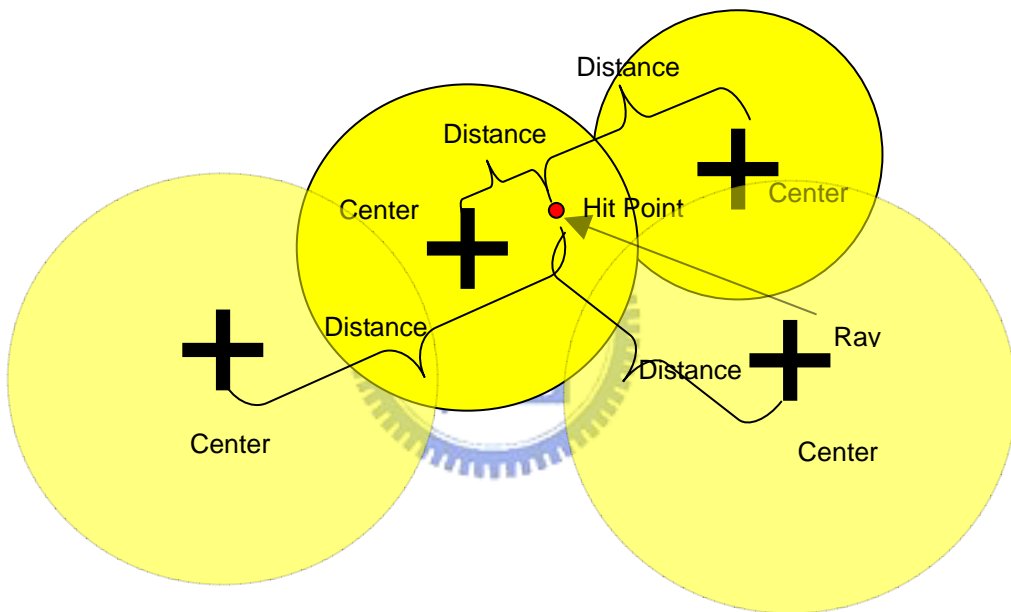
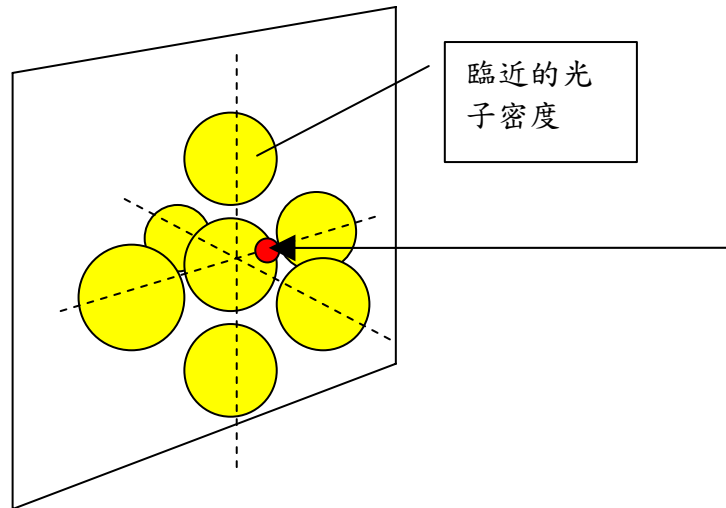


圖23 計算交點的光子密度

場景的明暗彩現係依光子密度值來顯示。光子密度值愈高，代表在這個空間內光源的光子愈多，因此呈現出較亮的顏色，故以白色代表最亮，黑色代表最暗。

當Ray與物體表面的交點計算出其光子密度後，再與物體表面顏色進行混色即可得到這個點的色彩。為了加速運算，在視角不變動的情形下，我們可以記下這個交點供下次彩現時使用。

圖24顯示場景彩現的流程。在執行場景彩現程序前，其投影在螢幕上的顯示平面寬度為 W ，高度為 H 。光源位置在 $L(x,y,z)$ 。

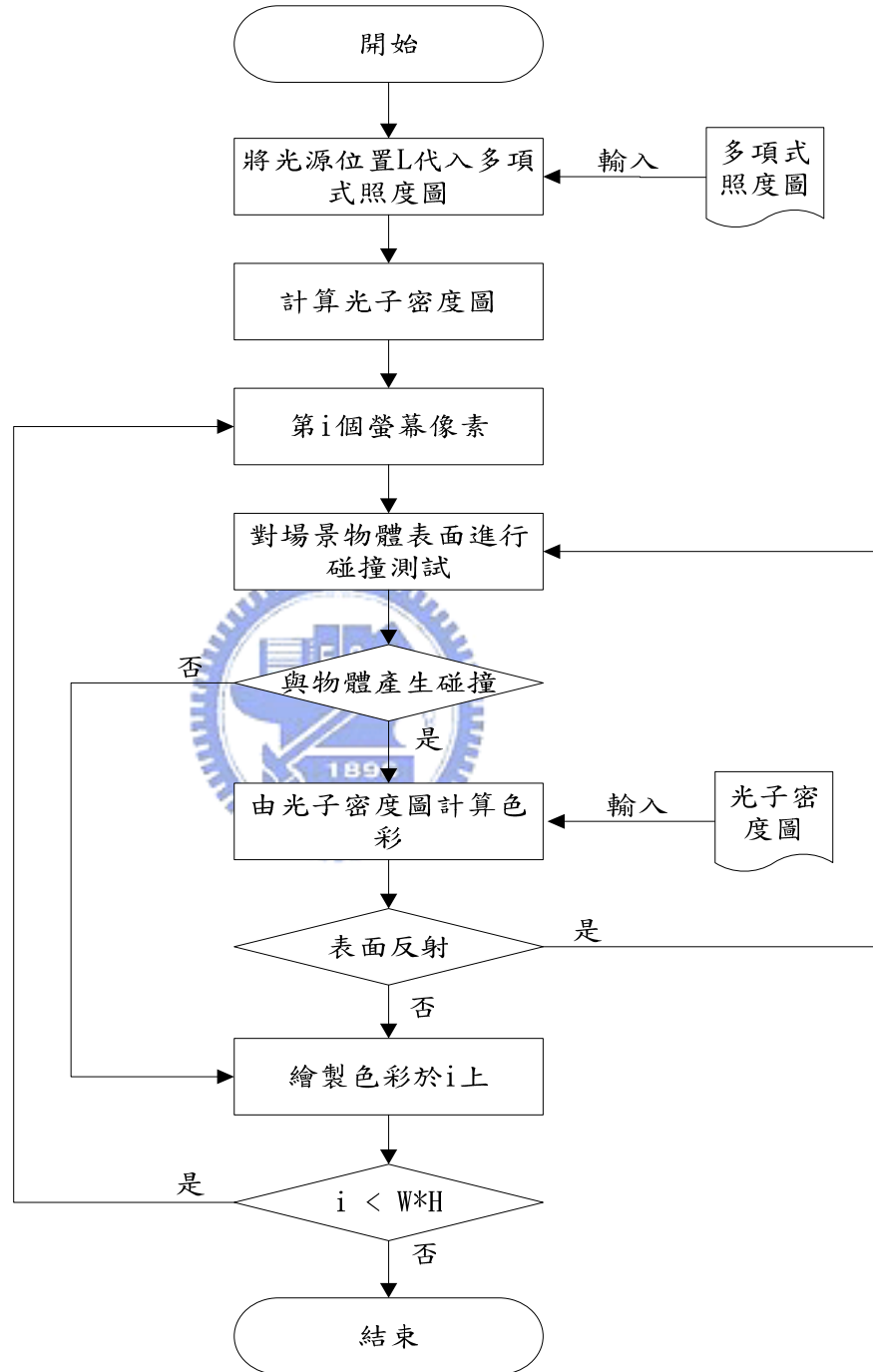


圖24 場景彩現流程

3.4 多照明光源應用

場景中多光源照明是常見的。我們可以很簡單的使用相同的技術，將場景中單一光源延伸至多光源的應用。其方法為在前處理的程序中，將每一個光源的計算結果以獨立的係數存於多項式照度圖中，例如：若有三個光源應用於場景中，則每一個cell即存在著三組係數。在進行場景彩現時，分別依每個光源取其所屬的係數計算後的結果進行疊合，如圖 25所示。

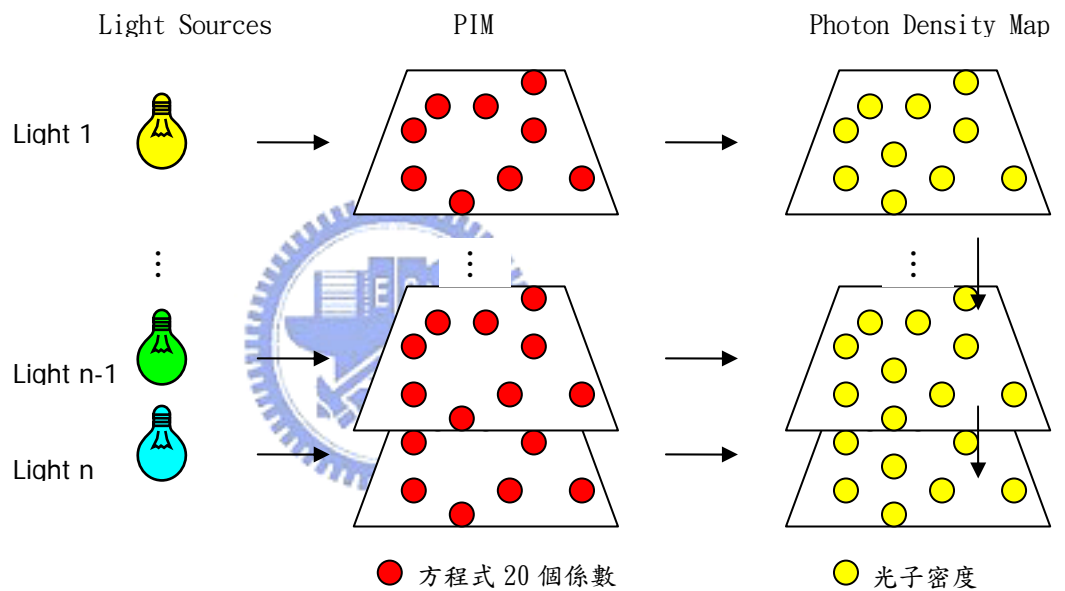


圖25 多照明光源的計算

第四章、實作過程與結果討論

4.1 實驗說明

本論文實驗為實作場景光源位置變動的照度計算，並展現執行時間及彩現畫面，並在實驗過程中針對結果作說明及討論。本章節的安排先就多項式全域照明方法進行實作，第一個實驗設定為單一光源，其中說明執行及開發環境、場景設定資訊及程式實驗執行流程，並列出實驗執行的各項測試時間。第二個實驗設定為多照明光源，在相同的執行環境下增加場景光源進行測試，並列出實驗執行的測試時間。其後我們呈現三次方多項方程式進行照度近似計算的結果。最後，我們將場景彩現的畫面列出並針對光源移動的照度變化進行說明。

4.2 多項式全域照明計算結果

4.2.1 實驗一：單一照明光源場景測試

我們將前述所提的構想及演算法進行實作，表 1 為程式開發環境，所有實驗計算過程及結果的彩現成像完全透過軟體方式進行計算，並沒有使用硬體加速圖形計算，而表中所列 3D API 僅用來呈現場景安排、光源位置變動及視角更改等使用者互動操作部份。表 2 為程式執行實驗環境。

程式工具	Microsoft Visual C++ 6.0
3D API	Microsoft DirectX 9.0 SDK
作業系統	Microsoft Windows XP SP2

表 1 程式開發環境

中央處理器	AMD Athlon 1.3GHz
記憶體容量	768MB RAM
作業系統	Microsoft Windows XP SP2

表 2 實驗測試環境

我們收集了兩個場景進行實驗：寺廟(Temple)及修道院(Cloister)，場景模型由 [WWW]處下載取得，表 3 為場景的幾何資訊。實驗執行時，我們採用視角固定的方式，並且只針對場景中之可視點進行彩現成像。

項目 實驗場景	Object 數目	Vertex 數目	Polygon 數目
Temple	97	9674	19230
Cloister	81	40939	81410

表 3 實驗測試場景

對於每一個測試場景，實驗執行的流程如圖 26所示。主要分為兩部分，第一部分為前處理，首先進行光子發散及收集並進行光子密度圖轉換，直至取樣完成所有光源位置，並計算出多項式照度圖。第二部分為互動操作及彩現顯像計算，由新的光源位置透過多項式照度圖還原光子密度圖，接著進行場景彩現計算。

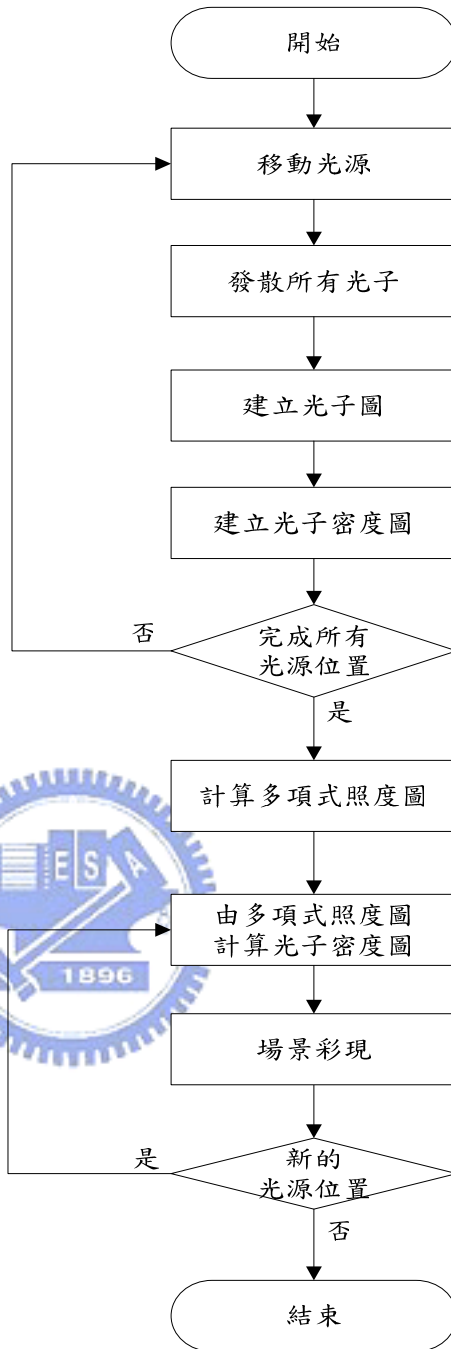


圖 26 程式實驗執行流程

實驗進行前，我們需先設定測試參數值，表 4 之數值分別表示測試場景在每個光源取樣點所發散光子數量，同時也顯示場景空間分割後的空間取樣點數目。

項目 實驗場景	光源發散光子數	場景空間取樣點	光源取樣位置
Temple	40000	44100	182
Cloister	30000	61712	96

表 4 測試場景進行光源取樣及空間分割的數據

表 5 中，(A)項目之數值表示測試場景由每個光源取樣位置進行光子發散、追蹤及收集，並轉換至光子密度圖花費的時間；(B)項目之數值表示將這些光子密度圖轉為多項式照度圖其計算多項式係數所花費的時間。將這兩個計算時間加總即為前處理所花費的總時間，如表格中(A)+(B)項目。表格中最後一項數值為場景使用多項式照度圖進行彩現顯像時每秒繪圖的畫面格數。

項目 實驗場景	(A) 光源發散光子 至建立光子密度圖	(B) 計算多 項式照度圖	(A) + (B) 前處理 時間	彩現顯像 (Frames / Sec)
Temple	2760 秒	47 秒	2807 秒	2.2916
Cloister	3120 秒	56 秒	3176 秒	1.375

表 5 測試場景進行前處理計算及彩現階段所花費的時間

表 5 中，我們在相同場景採用不同的彩現方式進行光源位置變動時計算速度比較。可以從表中數值觀察到採用本論文的多項式照度計算節省大量的計算時間。

項目 實驗場景	Ray Tracing	Photon Mapping	Polynomial Illumination Map
Temple	12 秒	10.6 秒	<1 秒
Cloister	21.3 秒	20.3 秒	<1 秒

表 6 不同的彩現方式計算時間比較

4.2.2 實驗二：多照明光源場景測試

為了在多照明光源場景中應用我們的方法並加以測試其執行時間，我們以寺廟場景來實驗，將光源數目由一個逐步增加至四個，並且分別測試其場景彩現的速度。表 7 為在寺廟場景進行實驗，數值顯示當光源數目增加時，其繪圖速度與僅有一盞光源的比較，由表中顯示當光源數由一盞增為二盞時，其繪圖速度下降了 22.586%，若再增加為三盞其繪圖速度則下降了 33.317%，當場景中光源數為四盞時，其繪圖速度下降了 41.450%。由表列數值可以得知，光源的增加僅為多項式計算次數的增加，所增加的繪圖時間可視為一固定的數值。這是由於在相同場景下，每盞光源所要處理的多項式照度圖計算量幾乎相同。

項目 光源數目	彩現顯像(Frames / Sec)	相對繪圖速度
1	2.041	-
2	1.580	-22.586%
3	1.361	-33.317%
4	1.195	-41.450%

表 7 數值顯示場景光源增加所造成的繪圖速度減慢程度

4.2.3 三次方多項方程式近似結果

當場景中光源移動的範圍較大時，其每一個取樣 Cell 其光子密度的變化也變得複雜，使用三次方程式逼近較二次方程式有著更佳的结果，因此我們使用二十個係數儲存於每個 Cell 中，並使用多項式及光源位置來還原其所對應的光子密度。

如圖 27 所示，顯示場景中的一個取樣 Cell 的照度變化，橫向表示所有取樣光源的位置，縱向表示光子密度。紅色表示原始光子密度值，藍色表示使用三次方程式逼近的結果，綠色表示與原始值的誤差值。

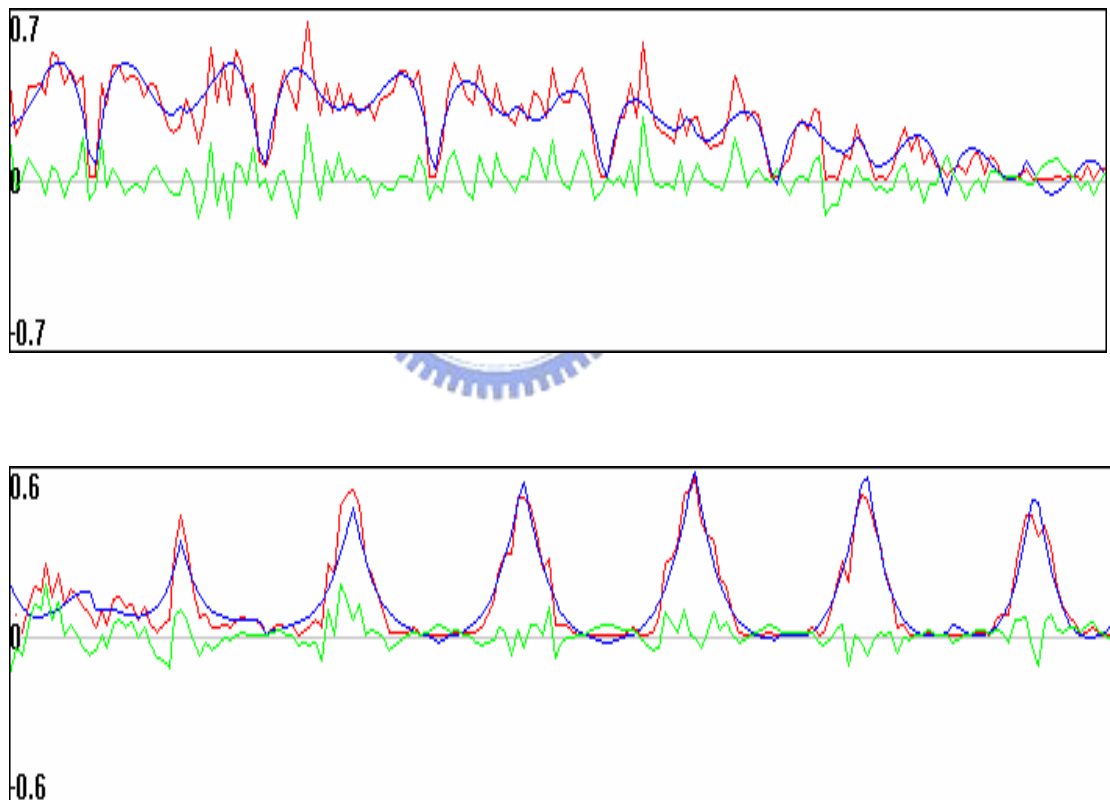


圖 27 場景中不同的取樣 cell 的多項式結果圖

4.3 測試場景彩現結果

第一個測試的場景為寺廟，畫面解析度設為 393 x 393 像素，執行畫面更新速度約為每秒 2.2 個畫面，實驗執行的畫面顯示於圖 28、圖 29 及圖 30 中。

圖 28 為取自於上前方觀察角度的連續畫面中的三張影像，左邊的圖片顯示光源位置移至場景的右下角，中間的圖片顯示光源位置移至場景中間上方的位置，右邊的圖片顯示光源位置移至場景的左上角。圖 29 為採用側角度來觀察場景照度變化的情形，光源移動的方向同前描述。圖 30 為採用斜角度來觀察場景照度變化的情形，光源移動的方向由場景中右上角移至左下角。





圖28 寺廟場景上前方的觀察角度的照度變化情形



圖29 寺廟場景側方向觀察角度的照度變化情形

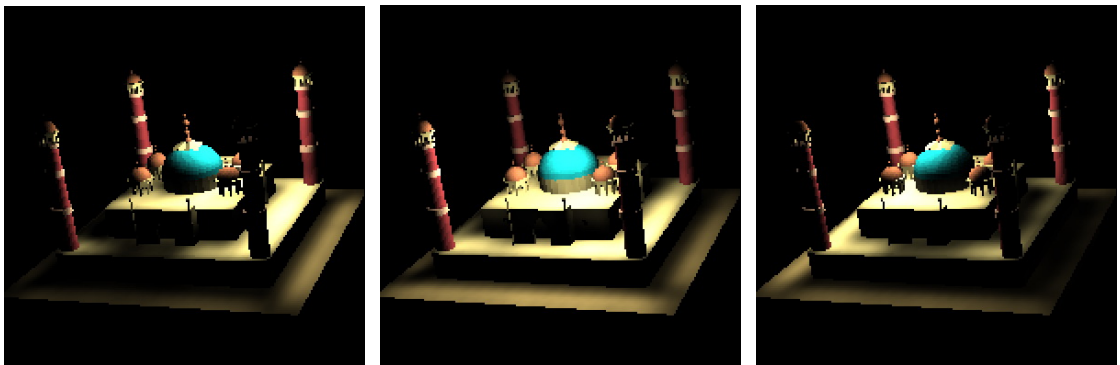


圖30 寺廟場景斜方向觀察角度的照度變化情形

第二個測試的場景為修道院，畫面解析度為 393 x 393 像素，執行畫面更新速度約為每秒 1.3 個畫面，實驗執行的畫面顯示於圖 31、圖 32 及圖 33 中。

圖 31 為取自於前方觀察角度的連續畫面中的三張影像，左邊的圖片顯示光源位置移至場景的右下角，中間的圖片顯示光源位置移至場景右上角，右邊的圖片顯示光源位置移至場景的左邊。圖 32 為採用斜角度來觀察場景照度變化的情形，光源移動的方向同前描述。圖 33 為採用場景右方角度來觀察場景照度變化的情形，光源移動的方向由觀察角度的左邊移向右邊。





圖31 修道院場景前方的觀察角度的照度變化情形



圖32 修道院場景斜方向觀察角度的照度變化情形



圖33 修道院場景右方向觀察角度的照度變化情形

第五章、結論

5.1 研究總結

在本論文中，我們使用了以光子映射法為基礎的全域照明方法及多項式貼圖法，對場景進行光源照度的計算，並提出了光子密度圖，將場景照度結果轉為以空間光子密度的表示方法，這使得數量龐大的光子資料得以減少，以利於多項式的計算。之後將光源移動的位置及其對應的光子密度圖之變化關係，經由三次方多項式計算以求得空間中每一取樣點之二十個多項式係數，並將其儲存於多項式照度圖中，用來還原對應於光源位置的光子密度圖。有了這些準備，便可在給定新的光源位置後，很快的計算出整個場景的照明亮度。由實驗中得知，經過上述的程序處理，可加速取得全域照明計算之結果，省略光源發散光子計算的步驟，縮減大量計算時間。

此外，經由我們觀察測試場景的結果，在場景中以較細的空間分割而形成的取樣cell，可得到較精確的空間照明數值，但過多的取樣cell卻會導致較長的計算時間。另外，隨著光源的取樣點增加，空間中每一取樣cell之光子密度變化也隨之複雜，由於是使用多項方程式來進行結果逼近，因此從圖 27中可以看到如此將會損失一些細微的數值變化，這樣的結果會造成彩現時陰影部份的細節消失，而呈現出較為模糊的影子。在加速計算方面，於建立多項式照度圖時，我們可以將未與物體表面接觸的取樣cell忽略，這些取樣cell通常像是空間中的間隙，如此便可加速多項式照度圖的計算及處理。

5.2 未來工作

我們提出一個新的方法採用多項式近似的方式來進行全域照明計算，延著這個方向，將有許多課題可以被提出研究。其中一個為加入可變動的視角，這需要加快處理本篇論文所提的方法。此外，由於多項式計算方式單純，亦可將整體執行架構於硬體上，應用GPU硬體加速計算，相信對速度上的提昇會有很大的助益。另一方面，相對於場景中光源位置變動，物體位置改變更常見於實際的應用，因此，加入對物體變動的全域照明計算亦為一重要的方向。



參考文獻

- [Bli78] BLINN J. F.: *Computer display of curved surfaces*. PhD thesis, 1978.
- [DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 145–156.
- [DMG99] DISCHLER J.-M., MOSTEFAOUI L., GHAZANFARPOUR D.: Radiosity including complex surfaces and geometric textures using solid irradiance and virtual surfaces. *Computers & Graphics* 23, 4 (1999), 507–524.
- [EHY95] EPSTEIN R., HALLINAN P. W., YUILLE A. L.: 5 +/- 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models, 1995. IEEE Workshop on Physics-Based Vision.
- [HSA91] HANRAHAN P., SALZMAN D., AUPPERLE L.: A rapid hierarchical radiosity algorithm. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1991), ACM Press, pp. 197–206.
- [Jen95] JENSEN H. W.: Importance driven path tracing using the photon map. In *Rendering Techniques '95* (1995), pp. 326–335. (Proceedings of the Sixth Eurographics Workshop on Rendering).
- [Jen96] JENSEN H. W.: Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96* (London, UK, 1996), Springer-Verlag, pp. 21–30.
- [Jen97] JENSEN H. W.: Rendering caustics on non-lambertian surfaces. *Comput. Graph. Forum* 16, 1 (1997), 57–64.
- [Jen01] JENSEN H. W.: *Realistic image synthesis using Photon Mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

- [LTG93] LISCHINSKI D., TAMPIERI F., GREENBERG D. P.: Combining hierarchical radiosity and discontinuity meshing. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1993), ACM Press, pp. 199–208.
- [Kaj86] KAJIYA J. T.: The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)* (Aug. 1986), vol. 20, pp. 143–150.
- [LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bidirectional path tracing. In *3rd International Conference on Computational Graphics and Visualization Techniques (Compugraphics) '93* (Alvor, Portugal, 1993), Santo H. P., (Ed.), pp. 145–153.
- [LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)* (New York, NY, 1996), Springer-Verlag/Wien, pp. 91–100.
- [MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, pp. 519–528.
- [PDC*03] PURCELL T. J., DONNER C., CAMMARANO M., JENSEN H. W., HANRAHAN P.: Photon Mapping on programmable graphics hardware. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 41–50.
- [TSH97] TEO P. C., SIMONCELLI E. P., HEEGER D. J.: *Efficient Linear Rerendering for Interactive Lighting Design*. Tech. rep., 1997. Stanford Computer Science Department Technical Report STAN-CS-TN-97-60.
- [VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Proceedings of the 5th Eurographics Workshop on Rendering '94* (Darmstadt, Germany, jun 1994), pp. 147–162.
- [WBS03] WALD I., BENTHIN C., SLUSALLEK P.: Interactive global illumination

in complex and highly occluded environments. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 74–81.

[WKB*02] WALD I., KOLLIG T., BENTHIN C., KELLER A., SLUSALLEK P.: Interactive global illumination using fast ray tracing. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2002), Eurographics Association, pp. 15–24.

[Wat00] WATT A.: *3D Computer Graphics*. Addison-Wesley, 2000, pp. 349.

[WS03] WAND M., STRASSER W.: Real-time caustics. *Comput. Graph. Forum* 22, 3 (2003), 611–620.

[WWW] architecture models. <http://www.3dcafe.com>.

