

國立交通大學

電子工程學系 電子研究所碩士班

碩 士 論 文

使用多重攝影機影像的立體視差估算

Disparity Estimation Using Multiple Images

研 究 生：許紹唐

指導教授：杭學鳴 博士

中 華 民 國 九 十 七 年 七 月

使用多重攝影機影像的立體視差估算

Disparity Estimation Using Multiple Images

研 究 生：許紹唐

Student : Shao-Tang Hsu

指導教授：杭學鳴 博士

Advisor : Dr. Hsueh-Ming Hang



A Thesis

Submitted to Department of Electrical & Institute of Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Industrial Technology R & D Master Program on
IC Design

July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

使用多重攝影機影像的立體視差估算

研究生：許紹唐

指導教授：杭學鳴 博士

國立交通大學

電子工程學系 電子研究所碩士班

摘要

Free viewpoint TV (FTV) 是一個很新的立體影像技術，它可以由已知的 multiple images 合成虛擬視點的影像。藉著校準過的攝影機陣列所截取到的空間中景物的影像，FTV 系統幾乎能夠合成任何位置所能拍攝到的影像，因此它可以讓觀眾自由的選擇自己所喜歡的視角去欣賞影片。而在 FTV 系統的實務上，一種快速的方式是以 ray-space representation 來當做 image-based rendering 技術的基礎。

FTV 系統一個很重要的基本原素就是估算 multiple images 中物體的視差值，而這也就是本論文所探討的焦點。在本論文中，我們的目標是發展準確的視差估算技術以增加 image-based rendering 的可靠度。我們發現，四台攝影機的架構比起傳統立體影像研究中常見的兩台攝影機架構，在視差估算以及 image-based rendering 方面

提供了較多有用的資訊，因此我們將兩台攝影機的架構推廣為四台攝影機。

為了增進遮蔽區域以及無紋理區域的視差估算的準確度，我們提出了單邊視窗對立體匹配方法。以及採用了動態規劃演算法求得整體上較佳的視差值。我們在 Middlebury College 提供的包含了擁有視差圖的 ground truth 的立體影像資料上測試我們的演算法的視差估算能力。並且以 blender 合成的虛擬場景的 multi-camera 影像來實驗我們的四台攝影機架構的 image-based rendering 演算法，結果顯示這樣的做法有不錯的成效，值得進一步發展。



Disparity Estimation Using Multiple Images

Student: Shao-Tang Hsu

Advisor: Dr. Hsueh-Ming Hang

Department of Electronic Engineering &
Institute of Electronics
National Chiao Tung University

Abstract

Free-viewpoint TV (FTV) is a novel technique that can render a virtual scene based on the given multiple images and thus it allows people to choose their favorite viewpoint freely. The 3-D objects are captured by a calibrated camera array. The suggested FTV system can synthesize the projected images of these 3-D objects from nearly any virtual camera position. In practical implementation, researchers use the image-based rendering technique based on the ray-space representation.

One essential element in the FTV synthesis procedure is to identify the object disparity information, which is the focus of this thesis. We adopt the ray-space representation and develop accurate matching techniques to increase the reliability of image-based rendering. We find that, comparing to the traditional 2-camera stereo set-up, a 4-camera structure provides a much better results on disparity estimation and thus image rendering.

To improve the accuracy on estimating the correct disparity values in the occluded and non-textured regions, we propose the single-sided window pair method. Also, we employ the dynamic programming algorithm to find the global optimal disparity estimate. We test our algorithms on the Middlebury College stereo image data sets with ground-truth disparity values. Also, we examine the 4-camera algorithm on the synthesized multi-camera scenes. The results are very encouraging.

誌 謝

能夠完成這篇論文，首先我要感謝我的指導教授，杭學鳴博士。在我兩年的碩士生涯指引我的研究方向，教導我求學和人生應有的態度，給了我研究方面最大的自主空間以及生活上各方面的關懷與幫助。接下來要感謝的是我的學長，張峰誠博士，學長給予我研究方面最直接與即時的指導與幫助。

我要感謝我的父母，雖然遠在故鄉的你們總是和我聚少離多，一年見不著幾次面。但是你們不求回報的愛與付出我都能感受得到。另外我也要感謝一路上與我互相扶持的同窗好友們，希望未來有緣能再一起奮鬥。

最後我要特別感謝，交大跆拳道社與交大跆拳道代表隊的吳竹仁教練、吳俊賢教練，還有全體的社員及隊員。在這六年當中，有你們在的新竹給了我這個異鄉學子家庭般的溫暖。

真的非常謝謝你們。



目錄

中文提要	i
英文提要	iii
誌謝	iv
目錄	v
表目錄	vii
圖目錄	viii
第一章	緒論.....	1
第二章	背景資料.....	4
2.1	基於圖像的繪製.....	4
2.2	針孔成像與相機校準.....	5
2.2.1	針孔成像與透鏡成像.....	5
2.2.2	針孔成像的公式.....	6
2.3	光線空間表示法.....	8
2.4	立體匹配.....	10
2.4.1	基於像素的立體匹配.....	10
2.4.2	基於視窗的立體匹配.....	11
2.4.3	動態規劃.....	11
2.5	Imagemagick 與 PNG	12
2.6	Blender.....	13
第三章	遮蔽對視差估算的影響.....	14
3.1	遮蔽區域產生的問題.....	14
3.2	推廣至多張輸入影像的架構.....	17
3.3	單邊視窗對立體匹配.....	19
第四章	動態規劃與總體最佳化.....	23

4.1	動態規劃的基本想法與流程·····	25
4.2	基本的總體最佳化概念·····	25
4.3	對稱立體演算法·····	29
4.3.1	對稱立體演算法的基本限制·····	29
4.3.2	以動態規劃實現對稱立體演算法·····	30
4.3.3	以四張輸入影像的架構修補對稱立體演算法··	32
第五章	實驗結果·····	34
5.1	實驗架構·····	34
5.1.1	實驗用的影像·····	34
5.1.2	實驗流程與變因·····	36
5.2	視差估算的實驗結果·····	38
5.2.1	由兩張輸入影像推廣至四張輸入影像的架構··	38
5.2.2	以總體最佳化取代 WTA ·····	41
5.3	Image-based rendering 的實驗結果 ·····	44
5.3.1	場景一的實驗結果·····	45
5.3.2	場景二的實驗結果·····	47
5.3.3	場景三的實驗結果·····	51
第六章	結論與未來工作·····	55
參考文獻	·····	58

表 目 錄

表 3.1	各種方法估算單一畫素視差的複雜度比較表	22
表 4.1	圖 4.1(a)中無紋理區域內某一畫素的 SSD 數據	24
表 5.1	各場景中三種立體匹配方法搭配 WTA 的視差估算 的均方誤差	40
表 5.2	三種立體匹配方法搭配總體最佳化在各場景視差估 算的均方誤差圖目錄	44
表 5.3	三種立體匹配方法搭配視差決策方式的 PSNR 列表	54



圖目錄

圖 2.1(a)	針孔成像模型	5
圖 2.1(b)	透鏡成像模型	5
圖 2.2	針孔成像公式的座標安排	6
圖 2.3	兩台校準過的相機與座標系統	7
圖 2.4(a)	相機陣列與場景	8
圖 2.4(b)	將輸入影像轉成離散 EPI	9
圖 2.4(c)	連續 EPI	9
圖 2.5	EPI 內插示意圖	10
圖 3.1	Image-based rendering 程式架構圖	14
圖 3.2(a)	常見的一次性 rendering 架構	15
圖 3.2(b)	階段性的 rendering 架構	15
圖 3.3(a)	場景示意圖	15
圖 3.3(b)	輸入影像(共兩張)	16
圖 3.3(c)	ground truth	16
圖 3.3(d)	基於畫素的立體匹配	16
圖 3.3(e)	基於視窗的立體匹配	16
圖 3.4	遮蔽區域的干擾示意圖	17
圖 3.5(a)	假設不存在的狀況	17
圖 3.5(b)	實際 FTV 系統示意圖	17
圖 3.6	四張輸入影像的架構	18
圖 3.7(a)	groundtruth	18
圖 3.7(b)	基於畫素的立體匹配	18
圖 3.7(c)	基於視窗的立體匹配	18
圖 3.8	基於視窗的立體匹配方法在四張輸入影像架構 之下的遮蔽區域示意	19
圖 3.9(a)	六階層 SWP 的示意圖	20
圖 3.9(b)	SWP 迴避遮蔽區干擾示意圖	20

圖 3.10(a)	SWP 搭配兩張輸入影像所得的視差圖	20
圖 3.10(b)	SWP 搭配兩張輸入影像所得的視差圖	20
圖 3.11	以加法取代乘法的 SWP 運算架構	21
圖 4.1(a)	輸入影像	23
圖 4.1(b)	ground truth	23
圖 4.1(c)	基於畫素的立體匹配	23
圖 4.1(d)	基於視窗的立體匹配	23
圖 4.1(e)	SWP	23
圖 4.2	基本的總體最佳化概念圖	27
圖 4.3(a)	輸入影像	28
圖 4.3(b)	ground truth	28
圖 4.3(c)	$K=0$ (WTA)	28
圖 4.3(d)	$K=10$	28
圖 4.3(e)	$K=20$	28
圖 4.3(f)	$K=40$	28
圖 4.4(a)	匹配狀況不佳的視差圖	28
圖 4.4(b)	不理想的輸出影像	28
圖 4.5	以基本總體最佳化尋找匹配點的不對稱狀況	29
圖 4.6	對稱立體演算法中第三條限制下不允許發生的情況	30
圖 4.7	對稱立體演算法中第四條限制式的示意圖	30
圖 4.8(a)	實際的遮蔽狀況	31
圖 4.8(b)	遮蔽代價非定值可能產生的錯誤	31
圖 4.9	以 $L1$ 為基準的立體匹配示意圖	31
圖 4.10	對稱立體演算法的子問題與其可能的解	32
圖 4.11	因輸入影像失真而產生的失誤狀況與解決方法	32
圖 4.12	對稱立體演算法與修補機制示意圖	33
圖 4.13(a)	輸入影像	33
圖 4.13(b)	ground truth	33

圖 4.13(c)	基本總體最佳化所得到的視差圖	33
圖 4.13(d)	對稱立體演算法的最佳化與遮蔽區(畫面中的白色區域)之偵測	33
圖 4.13(e)	以四張輸入影像架構修補圖 4.13 (d)遮蔽區之後的視差圖	33
圖 5.1(a)	Middlebury College 估測視差的所使用的 structured light	34
圖 5.1(b)	某一立體影像與 structured light 找到的相對應的視差圖	34
圖 5.2	Middlebury College 提供，用來進行視差估算的十五組立體影像	35
圖 5.3(a)	以 Blender 架設場景合成輸入影像以及輸出影像的理論值	36
圖 5.3(b)	場景一：會議室	36
圖 5.3(c)	場景二：祭壇	36
圖 5.3(d)	場景三：小女孩	36
圖 5.4	實驗流程圖	37
圖 5.5	基於畫素的立體匹配搭配 WTA 估算視差圖的均方誤差	39
圖 5.6	基於視窗的立體匹配搭配 WTA 估算差圖的均方誤差	39
圖 5.7	SWP 搭配 WTA 估算視差圖的均方誤差	40
圖 5.8	基於畫素的立體匹配搭配總體最佳化估算視差圖的均方誤差	41
圖 5.9	基於視窗的立體匹配搭配總體最佳化估算視差圖的均方誤差	42
圖 5.10	SWP 搭配總體最佳化估算視差圖的均方誤差	43
圖 5.11(a)	輸入影像	43
圖 5.11(b)	ground truth	43



圖 5.11(c)	以 SWP 求得的視差圖	43
圖 5.12	輸入與輸出影像的視點	45
圖 5.13(a)	WTA	45
圖 5.13(b)	基本的總體最佳化	45
圖 5.13(c)	對稱立體演算法	46
圖 5.14(a)	理論影像	46
圖 5.14(b)	合成結果	47
圖 5.14(c)	誤差	47
圖 5.15(a)	WTA	48
圖 5.15(b)	基本的總體最佳化	48
圖 5.15(c)	對稱立體演算法	48
圖 5.16(a)	理論輸出影像	49
圖 5.16(b)	實際輸出影像	49
圖 5.16(c)	誤差	50
圖 5.16(d)	視差圖	50
圖 5.17	場景二中最底的 SD 值大於一千的畫素分布圖	51
圖 5.18(a)	WTA	51
圖 5.18(b)	基本的總體最佳化	51
圖 5.18(c)	對稱立體演算法	52
圖 5.19(a)	理論輸出影像	52
圖 5.19(b)	實際輸出影像	53
圖 5.19(c)	誤差	53
圖 5.19(d)	視差圖	53

第一章

緒論

有鑑於 free-viewpoint TV (FTV)[1]系統可能被納入 MPEG 的標準，而與 FTV 類似的 light field 系統的各種研究也不曾中斷。我們預期未來，可以讓觀眾自由切換觀賞視角的影音系統將會漸漸的普及。在 FTV 系統中，最重要的技術就是能夠合成任意視點所看到的影像的，基於圖像的繪製(image-based rendering)技術。較早期的立體影像合成技術乃是基於空間場景建模，好處是只要建模工作越仔細，合成的立體影像就可以有越高的精確度。然而它的缺點是計算較為費時，而且場景建模的工作得由人工完成，沒有一套自動化處理的機制。因此，在高度要求即時性(real time)的訴求之下，人們選擇以 image-based rendering 來實作 FTV 系統。

FTV 系統中的 Image-based rendering 技術能夠由經過校準的(calibrated)相機陣列截取的影像中解讀出足夠的空間場景資訊，進而合成虛擬視點的影像，最主要是得力於 ray-space representation[1]。Ray-space representation 的原理與傳統的立體匹配(stereo matching)技術原理是相通的。立體匹配最主要的工作就是找出每個物體在兩張輸入影像當中的位置的差異，也就是視差(disparity)。知道每個物體的視差值之後就可以做更進一步的空間場景的立體架構估算，或是搭配輸入影像進行 image-based rendering。因此，我們把研究的重點放在增進視差估算的準確度上。

從各種立體匹配的文獻上我們發現，視差估算最容易發生錯誤的地方在於較遠的物體被較近的物體遮蔽的區域，或是物體花紋不明顯的無紋理區域。因此我們試著改善這兩種區域的視差估算準確度。

首先，爲了善加利用 FTV 擁有數十台攝影機的特性，我們把傳統立體匹配

領域中兩台攝影機的架構推廣至多台攝影機，基於種種考量，我們認為一次考慮四台攝影機的輸入影像對於視差估算以及 **image-based rendering** 是最佳的架構。而為了有效的運用四台攝影機架構中，新增加的兩台攝影機所帶來的資訊，我們試驗比較了各種現存的立體匹配技術，在視差估算結果並分析其優劣之後，我們綜合了各種現存的技術之優點，提出了新的單邊視窗對立體匹配法(**single-sided windows pair**)來搭配四台攝影機架構並以實驗證實它提升了遮蔽區域的視差估算準確度。

在著手處理無紋理區域的問題，我們引入了總體最佳化的概念，並且以廣為應用於各種最佳化問題中的動態規劃演算法來做加速。在比較了各種總體最佳化概念之後。我們發現 **G. Van Meerbergen** 提出的階層式對稱立體演算法[4]當中，同時對兩張影像做對稱式的最佳化概念所得到的視差估算最準確，符合 **image-based rendering** 的須求。於是我們著手將對稱立體演算法推廣至四張輸入影像的架構加以實作，並以實驗證實了它的確有效的增進了無紋理區域的視差估測準確度。於是我們決定了以四張輸入影像的架構搭配單邊視窗對以及對稱立體演算法，做為視差估算的方法。

由於 **FTV** 所使用的攝影機陣列所費不貲，而且架設與校準的工作更是困難重重，要以攝影機陣列來進行實驗並不可行。所幸，**Middlebury College** 將他們以相機陣列拍攝的立體影像，以及視差 **ground truth** 資料公開在網路上供人利用，我們因此有了相當完整的資料進行視差估算演算法的驗證。

為了測試我們的演算法在 **image-based rendering** 上的效果，我們以 **Blander** 這套自由軟體的立體場景建模軟體進行場景建構，以及虛擬立體影像合成。分別完成了三個場景中虛擬的攝影機陣列拍攝到的影像，以及 **image-based rendering** 合成影像的理論值供我們的程式進行測試。

接下來我們在第二章進行幾何光學、傳統立體匹配技術等背景知識的介紹。在第三章介紹了如何將系統推廣至四張輸入影像以及單邊視窗對的優點。第四章說明了如何修改現存的以動態規劃加速的總體最佳化演算法以搭配我們的四張輸入影像架構。第五章中我們以實驗比較了我們的演算法與其它現存的方法在視

差估算與 image-based rendering 上的差異。最後在第六章列出結論與未來工作。



第二章

背景資料

本章前半段介紹基於圖像的繪製(image-based rendering)，以及一些與它相關的背景資料。從針孔成像的模型與公式到相機校準(calibration)的介紹，到基於針孔成像公式的原理而訂定出來的光線空間表示法(ray-space representation)。後半段則簡介了本論文實驗時使用到的工具：用於儲存本實驗各項結果的無損影像壓縮格式 PNG、以及幫助 PNG 編解碼的函式庫，ImageMagick。以及用於 3D 場景建構與 3D 影像合成的 blender。

2.1 基於圖像的繪製

基於圖像的繪製(image-based rendering)在 3D 影像領域是門新穎的課題。有別於傳統的技術，須要複雜的場景建模工作。image-based rendering 只須要分析輸入的圖像資料就能夠合成特定位置與角度的 3D 影像，容易達成即時(real time)的目標。

較有名的 image-based rendering 技術有 light field 以及 free viewpoint TV (FTV) 系統所採用的 ray-space representation[8]。它們的基本原理都是先紀錄了空間中的光線的明度色調等資料，再基於針孔成像的原理，計算合成特定的視點(viewpoint)影像所須要的光線，再從已紀錄的光線資料中抽取出有用的部分合成影像。

在三度空間中，光線可以以一個五維函數 $f(X, Y, Z, \theta, \varphi)$ [1] 來表示。以 (X, Y, Z) 表示位置， (θ, φ) 表示方向。足以表示從任何位置所輻射出的任何方向的所有可能的光線。五個維度的資料的資料量相當的龐大。所幸在沒有遮蔽的情況下，可以假設從某一點出發，沿著某個方向傳遞的光線，其行進路徑上的取樣

值近乎常數。因此在 light field 的研究中，有很多方法可以把光線的資料降至四個維度。而在我們的實驗中，考慮到現行 FTV 系統的實際情況，以及能夠取得的測試資料都是將相機架設在一條線上形成的相機陣列。所以僅記錄了離散的三個維度的光線資料。

2.2 針孔成像與相機校準

2.2.1 針孔成像與透鏡成像

針孔成像是相機的前身。它的原理很簡單：現實空間中的光線遵守著光線直進的定律，穿過了黑箱上的針孔投射到了箱子底部的屏幕上，形成了影像。然而若針孔太大則會有很多不同方向的光線投射到屏幕上一個點，造成模糊；針孔太小則屏幕上無法形成夠亮的影像。所以人們以能夠匯聚光線的凸透鏡來代替針孔，漸漸的演變成現在的相機。光學系統隨著時間不斷的發展，其實只是為了修正各種相差、形變失真，以趨近針孔所成的影像。

如圖 2.1 (b)所示，透鏡只能同時讓落在某個與光軸(optic axis)垂直的焦平面(focal plane)上的物體在屏幕上產生清晰的影像，當焦平面 a 上的點 A 產生清晰的影像，則焦平面 b 上的點 B 就會模糊。然而絕大多數用來截取數位影像的工具，感光元件面積都非常小，圖 2.1 (B)的模糊效應是很輕微可以忽略的。

基於以上兩段原理說明，我們可以把 FTV 系統的輸入影像視為針孔成像的結果加以分析和應用。

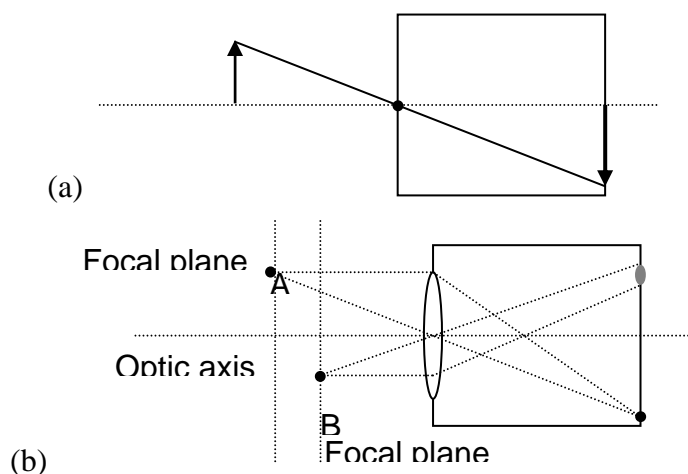


圖 2.1: 成象模型: (a) 針孔成像模型 (b) 透鏡成像模型

2.2.2 針孔成像的公式

針孔成像在數學上可以視為從三維的實體空間到二維的成像平面的一個線性映射。只要得到公式就能輕易的計算出空間中的場景投影到成像平面(retinal plane)的情形，什麼光線會投影到影像中的那個位置。這在 image-based rendering 的過程中是很重要的一個步驟。

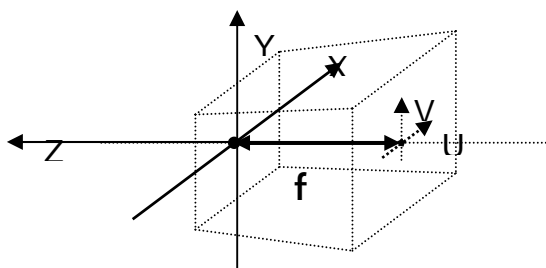


圖 2.2: 針孔成像公式的座標安排

Faugeras 在他的著作 Three-Dimensional Computer Vision [10]中提到針孔成像公式，假設空間座標與成像平面座標如圖 2.2 所定義。針孔位於實體空間的原點，光軸與空間座標的 Z 軸重合，而成像平面的座標軸(U,V)與(X,Y)平行，也就是往後論文中常提到的，經校準過的相機。則從空間中某一點(x,y,z)投影到成像平面上某一點(u,v)的映射公式如下：

$$u = f \times x / z \quad (2.1)$$

$$v = f \times y / z \quad (2.2)$$

其中 f 是針孔到成像平面的距離。若對應到實際的鏡頭成像情況，f 就是鏡頭的焦長(focus length)。

接下來我們考慮空間中某個點(x,y,z)在兩台不同的相機中的成像公式的差異。為了簡化問題，我們考慮圖 2.3 的兩台經過校準過的相機的情況。

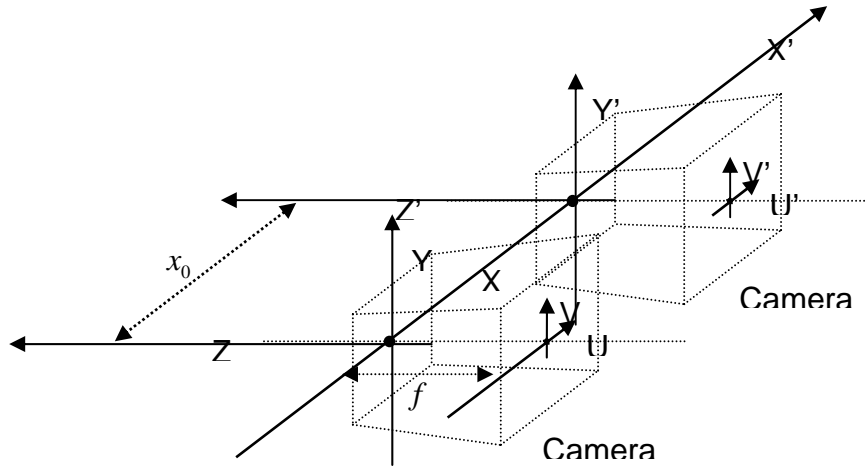


圖 2.3: 兩台校準過的相機與座標系統

Camera B 的針孔位在 $(x_0, 0, 0)$ 的位置，以這邊為原點得到一個新的空間座標系 (x', y', z') 可以寫成：

$$(x', y', z') = (x + x_0, y, z) \quad (2.3)$$

而 (x', y', z') 與 Camera B 的成像平面座標系 (u', v') 依然有著類似 (2.1)與 (2.2)的關係式：

$$u' = f \times x' / z' \quad (2.4)$$

$$v' = f \times y' / z' \quad (2.5)$$

當然，我們也可以把(2.3)重新寫成這樣：

$$(x' - x_0, y', z') = (x, y, z) \quad (2.6)$$

綜合(2.4), (2.5), (2.6), (2.7) 各式，我們得到：

$$(u', v') = (u + x_0 / z, v) \quad (2.7)$$

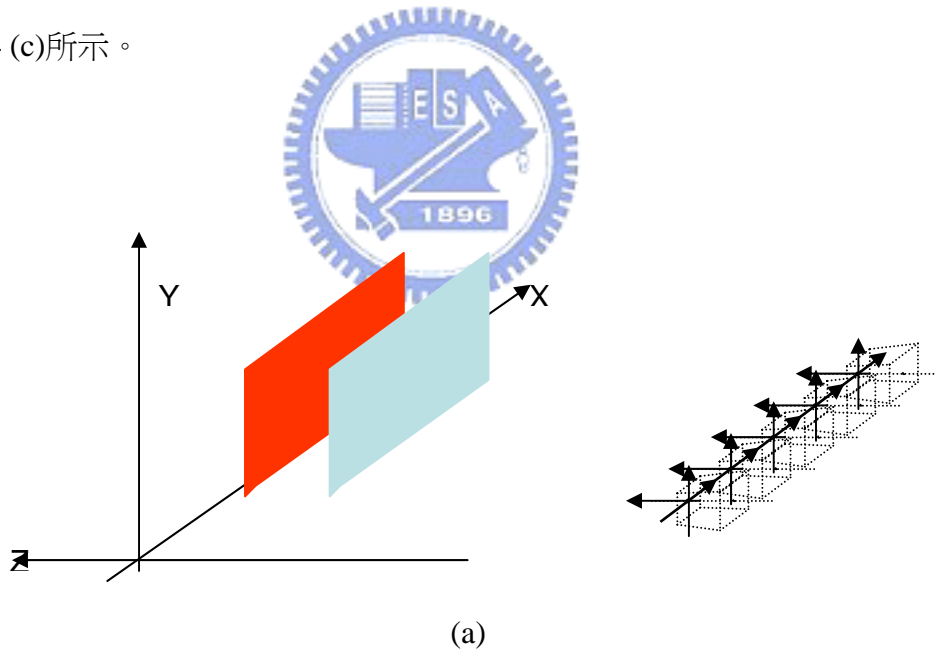
也就是說，空間中某個點 (x, y, z) 投影在兩台相機的成像平面上得到的座標值的關係式如同(2.7)所示。我們可以發現經過校準的相機之間成像的差異是一個很簡單的平移關係。雖然理論上我們能推導出各種任意安排的相機的成像關係，但是公式將不會像(2.7)一樣簡潔。因此實務上 FTV 系統所採用的也是將圖 2.3 所示的情況推廣成一系列數十台，類經過校準的相機陣列(camera array)。由(2.7)我們可以發現物體在鄰近兩台相機所成的影像的平移量為 x_0 / z ，若欲將 x_0 / z 從 (U, V) 座標系上的單位長度換算成畫素數，則須知道每單位長度上的畫素數也就

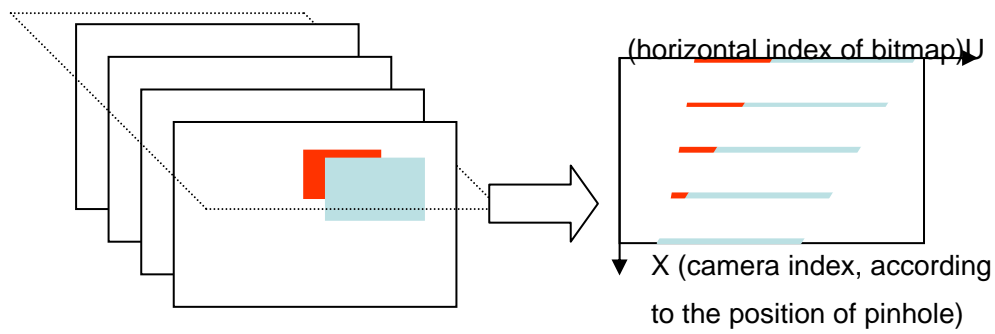
是畫素密度。假設畫素密度為 P ，則平移量就變成了 $P \times \frac{x_0}{z}$ ，也就是往後論文中一直提到，以 D 表示的視差(disparity)：

$$D = P \times \frac{x_0}{z} \quad (2.8)$$

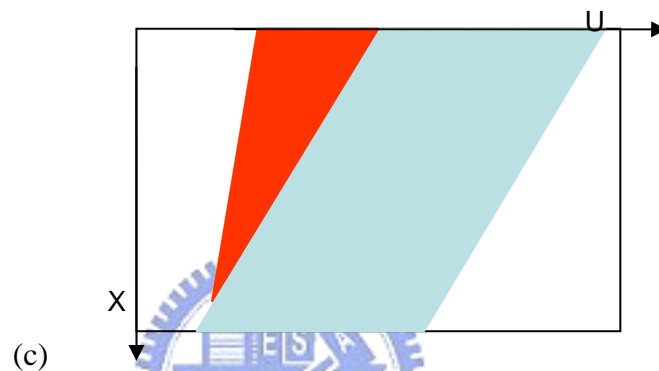
2.3 光線空間表示法

把相機陣列截取到的影像一張張儲存，並不是最適合 image-based rendering 的方式。如果要更方便合成影像與資料抽取，我們必須把影像資料重新安排，這就是光線空間表示法(ray-space representation)主要的目的。簡單來說，ray-space representation 就是把各張輸入影像垂直座標相同的水平掃描線抽取出來當成極線(epipolar line)[8]，存成各張離散的核極平面影像(epipolar plane image) (EPI)。如圖 2.4 (c)所示。





(b)



(c)

圖 2.4: 光線空間表示法: (a) 相機陣列與場景 (b) 將輸入影像轉成離散 EPI (c) 連續 EPI

假如我們能由圖 2.4 (b)的離散 EPI 重建出(c)的連續 EPI。那麼我們就能從各張 EPI 中抽出相同任意 X 值的水平掃描線，合成相機陣列所在的直線上的任意一個視點的影像，也就是 FTV 的目的。雖然以訊號處理觀點來說，離散 EPI 的內插是一個高比例的增加取樣(up-sampling)的步驟[2] [9]，理論上很難達到良好的效果，但是 EPI 有以下三種特徵：

- EPI 是由一條條不同斜率的直線所組成的
- 所有直線的斜率都是正的
- 斜率低的直線會遮蔽斜率高的直線。

這些公式很容易理解，由(2.8)式可以發現，視差值與相機位置的座標值 x_0 呈線性關係，所以空間中一個可見點在 EPI 上就會產生一條直線，而直線的斜率與 z ，也就是物體到針孔的距離成正比，固而不會有負值。最後 z 值越小也就是越靠近針孔者遮蔽遠離針孔者，所以斜率小的直線遮蔽斜率大的。善用這些特性我

們可以達到較佳的合成效果。

2.4 立體匹配

上一節提到，空間中每一個可見點都會在 EPI 上形成一條直線。而對於一個不過分光滑的材質，從不同角度所看到的色調與亮度不會有太大的差異。基於這個理由。EPI 內插的動作主要就是去判斷要內插的掃描線上的點是屬於那條直線，如圖 2.5 所示，若要內插點 P，我們可以在鄰近的兩條已知的極線上尋找連線通過點 P 的兩個點。以此二點的相似程度高低為依據，判斷 P 點與此二點在連續 EPI 上是否落在同一條直線上。亦即判斷 P 點是否與此二點在空間座標中對應到相同的點。若判斷為真，則以這兩個點的平均值去做內插。而這個判斷的工作和立體影像中發展了超過三十年的課題，立體匹配(stereo matching)的原理是相通的。以下將簡述立體匹配中的數個技術

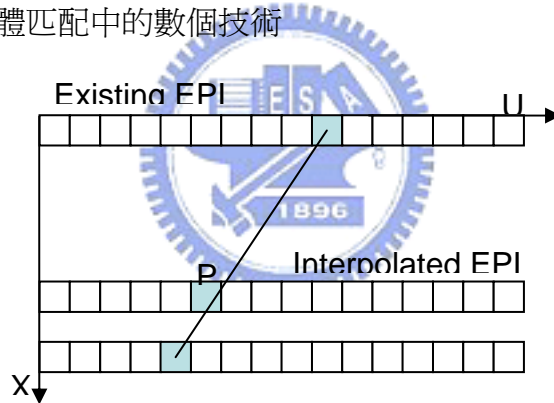


圖 2.5: EPI 內插示意圖

2.4.1 基於像素的立體匹配

基於像素的立體匹配(pixel-based stereo matching)[7]是最基本的立體匹配方法。首先我們會在相鄰的極線 x 上選一個點 u ，然後再從它相鄰的極線 $x-1$ 上選一個點 $u+d$ ， d 是某個可能的視差值。接下來用某些式子來計算匹配代價，最常用的就是兩個像素間的方差(SD)。方差表示式如下：

$$SD(u, d) = (R_{(x,u)} - R_{(x-1,u+d)})^2 + (G_{(x,u)} - G_{(x-1,u+d)})^2 + (B_{(x,u)} - B_{(x-1,u+d)})^2 \quad (2.9)$$

最後，把所有可能的 d 值的 SD 都計算出來，擁有最小的 SD 的 d 值被判斷

為該點的視差值。


雖然基於相素的立體匹配是很簡單的方法，但是它並不是一個很可靠的判斷方式。常見的問題是當遇到雜訊或是無紋理(non-textured)區域會容易出錯，因此基於視窗的立體匹配(window-based stereo matching)較常出現於實際的應用中。

2.4.2 基於視窗的立體匹配

基於”成像平面上視差變化緩和”的假設，我們在進行匹配判斷的點附近設定一個範圍，計算這個範圍內所有的點的 SD 並加總成為方差和(SSD) [6] [7]，公式如下：

$$SSD(u, d) = \sum_{k=u-w/2}^{u+w/2} SD(k, d) \quad (2.10)$$

有時候為了增加 SSD 值的鑑別度。人們會給予視窗中每個畫素不同的權重 [6]，如(2.11)所示。其中 $w(k)$ 通常是一個 $k=0$ 時有極大值且恆正的偶函數，例如高斯函數。


$$SSD(u, d) = \sum_{k=u-w/2}^{u+w/2} w(k) \times SD(k, d) \quad (2.11)$$

當無紋理區域大過視窗範圍時，基於視窗的立體匹配在判斷上還是會遇到困難，在各個可能的視差值都得到很低的 SSD。Takeo Kanade 發表的適應性視窗匹配法(A Stereo Matching Algorithm with an Adaptive Window)[3]，能夠有效的解決這個問題，但是它在視窗範圍的計算上耗費太多，不太可能達到即時。而較大的缺點是，基於視窗的立體匹配在遮蔽區及遮蔽區附近會發很大的問題。以實際的視差值代入(2.10)或(2.11)會得到很大的 SSD，在後面的章節中我們會發現這會造成視差判斷上嚴重的錯誤。這個問題我們在第三章會提出有效的解決方案。

2.4.3 動態規劃

在決定匹配的過程中，若根據特定單一基準(例如前述某一像點的 SD 或 SSD)直接判定，這種策略稱為 winner-take-all (WTA)。不過由於影像的特性或是雜訊誤差等諸多干擾，WTA 多半不能達到最佳的匹配結果。因此人們常常以總體最佳化(global optimization)的決策方式取代 WTA，而動態規劃是一套能夠有效降低

總體最佳化計算複雜度的演算法。

動態規劃(dynamic programming)[5]在數學和計算機科學中常常被用來解決包含重疊子問題的最佳化問題的方法。動態規劃的基本想法是把問題看成起始點、終點，以及中途的許多相連並且已知相互之間的距離的節點。從起點開始，從近到遠依序計算起點到達各個點的最短路程。一直到最後才選出到達終點的最短路徑。動態規劃常常被用在降低最佳化演算法的複雜度。例如以基於視窗的立體匹配計算 SSD 以判斷視差值。在無紋理區域，各個可能的視差值都會得到極低的 SSD，而使得 WTA 決策無法判斷正確值的時候，人們會引入總體最佳化的概念，幫助程式在許多可能性很高的視差值當中做出最恰當的決策。(2.12)式是一個很常見的作法：

$$E_{global} = \sum_{x=0}^{x=w-1} E_{data}(x, d(x)) + \lambda E_{smooth}(x, d(x)) \quad (2.12)[7]$$

將 $E(d)$ 最小化所得到的各個畫素的視差值，也就是所謂的視差圖(disparity map) 變化將趨於緩和，使得視差判斷在無紋理區域有較高的正確率。而欲最小化 $E(d)$ ，若使用遞迴之類的基本演算法，時間複雜度是 $O(d^{w \times h})$ 。藉由動態規劃，我們可以把複雜度降至 $O(d \times w \times h)$ 。其中 w, h 分別是水平與垂直畫素數， d 是所有可能的視差值的個數。關於動態規劃增進視差圖的準確性的細節，我們會在第四章深入的探討。

2.5 ImageMagick 與 PNG

爲了方便儲存與分析實驗數據，影像處理研究上大多選擇 PNG 這種無損壓縮的影像格式，我們也不例外。爲了處理 PNG 檔案的讀寫，我們採用了 ImageMagick 這套軟體的公開原始碼的函式庫來當做 PNG 的編解碼器。

ImageMagick 是一個強大的影像處理軟體。功能包含了讀取、轉換、寫入九十種格式的圖片。影像裁切、旋轉、組合、縮圖、銳化、顏色替換。ImageMagick 是免費軟體：全部程式碼公開，可以自由使用，複製，修改。對於 Perl, C, C++, Python, PHP, Ruby, Java 都有提供現成的 API。實作上我們下載並安裝 ImageMagick windows source 5.5.7 幫助我們在 Microsoft Visual C++ 6.0 SP5 環境

下開發的程式處理 PNG 影像的畫素直接存取以及檔案輸入輸出工作。 PNG (Portable Network Graphics)是爲了迴避 GIF 影像格式的專利問題而開發的點矩陣式影像格式。除了與 GIF 同樣採用 LZ 壓縮的修改版本 DEFLATE 進行無失真壓縮。壓縮率在各種無失真壓縮格式中表現的相當出色，然而編碼的時間複雜度較高。雖然 PNG 不支援動畫格式，然而 PNG 的 24 位元(最高可達 48 位元)彩色(GIF 爲 256 色)以及 256 階透明度十分符合實際應用的須求。

2.6 Blender

Blender 最初荷蘭的一個影片工作組 NeoGeo 與 Not a Number Technologies (NaN)設計爲內部使用的程式。但後來主要程式設計者 Ton Roosendaal 在 1998 年 6 月把它做進一步的發展，並對外發表。

在經過債權人同意後，Blender 付了十萬歐元後變爲自由軟體，並以 GNU 通用公共許可證發佈。Roosendaal 在 2002 年 7 月 18 日開始爲 Blender 募集資金，在 2002 年 9 月 7 日籌措了足夠資金，並對外公佈原始碼。Blender 現在是自由軟體，並由 Blender 基金會維護與更新。其主要功能與特色如下：

- 支持不同的幾何圖形，包括多邊形網紋，快速表層建模等
- 多功能的內部渲染及整合 YafRay 這個開源的射線追蹤套件。
- 強大的動畫工具
- 使用 Python 語言來創作及製作遊戲及工作自動化腳本。
- 基本的非線性影像編輯及製作功能。

Blender 內建的虛擬相機物件的感光元件尺寸是固定的，大小爲 32mmx 24mm。欲合成影像，可以給定鏡頭焦長或直接指定水平視角。而我們最常用的功能則是把相機設定爲特定的視點再合成虛擬的 FTV 系統攝影機所截取的影像，以及與我們用 image-based rendering 方法合成的輸出影像對照比較的理想影像。

第三章

遮蔽對視差估算的影響

本章透過一些基本匹配方法的實作來觀察遮蔽區域對視差估算造成的不良影響。並且將傳統的兩張輸入影像的架構擴展至四張輸入影像，這對於動輒數十台錄影機的 FTV 系統來說是可行的。然而實驗結果顯示，基本的匹配方法無法有效的利用多增加的兩張輸入影像當中的資訊來增加遮蔽區域的視差估算的準確度。於是我們綜合了三種基本匹配方法提出了單邊視窗對(single-sided windows pair) (SWP)來加強遮蔽區域的視差估算準確性。最後以實驗結果證明，SWP 搭配四張輸入影像的架構能夠有效的應付因遮蔽而產生的問題。

3.1 遮蔽區域產生的問題

一開始，我們實現了一個簡單的 image-based rendering 程式。它的架構如圖 3.1 所示。有別於常見的 image-based rendering 架構圖，我們把 SSD 計算與繪製的步驟分開。採用傳統立體匹配的方式先計算出整張輸入影像的視差圖，再依照視差圖與輸入影像來完成繪製的工作。原因是因為先繪製視差圖對實驗結果的觀察較有利，而且較容易以現有的動態規劃演算法套用在視差值的決策部分取代 winner-take-all (WTA)做加強。

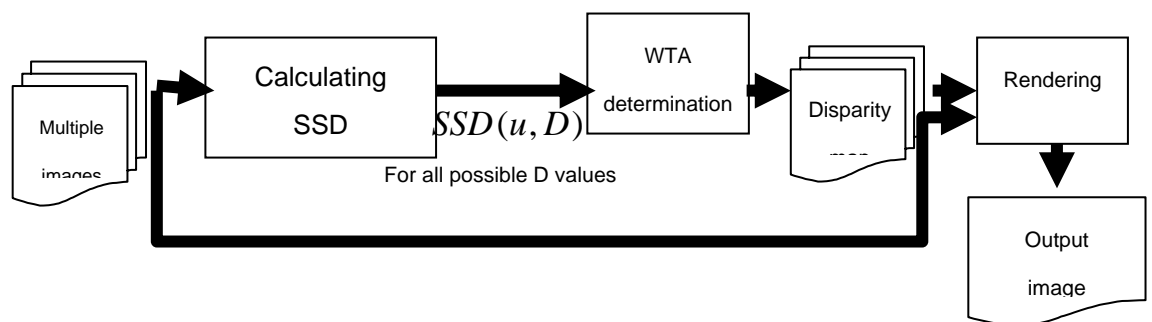


圖 3.1: image-based rendering 程式架構圖

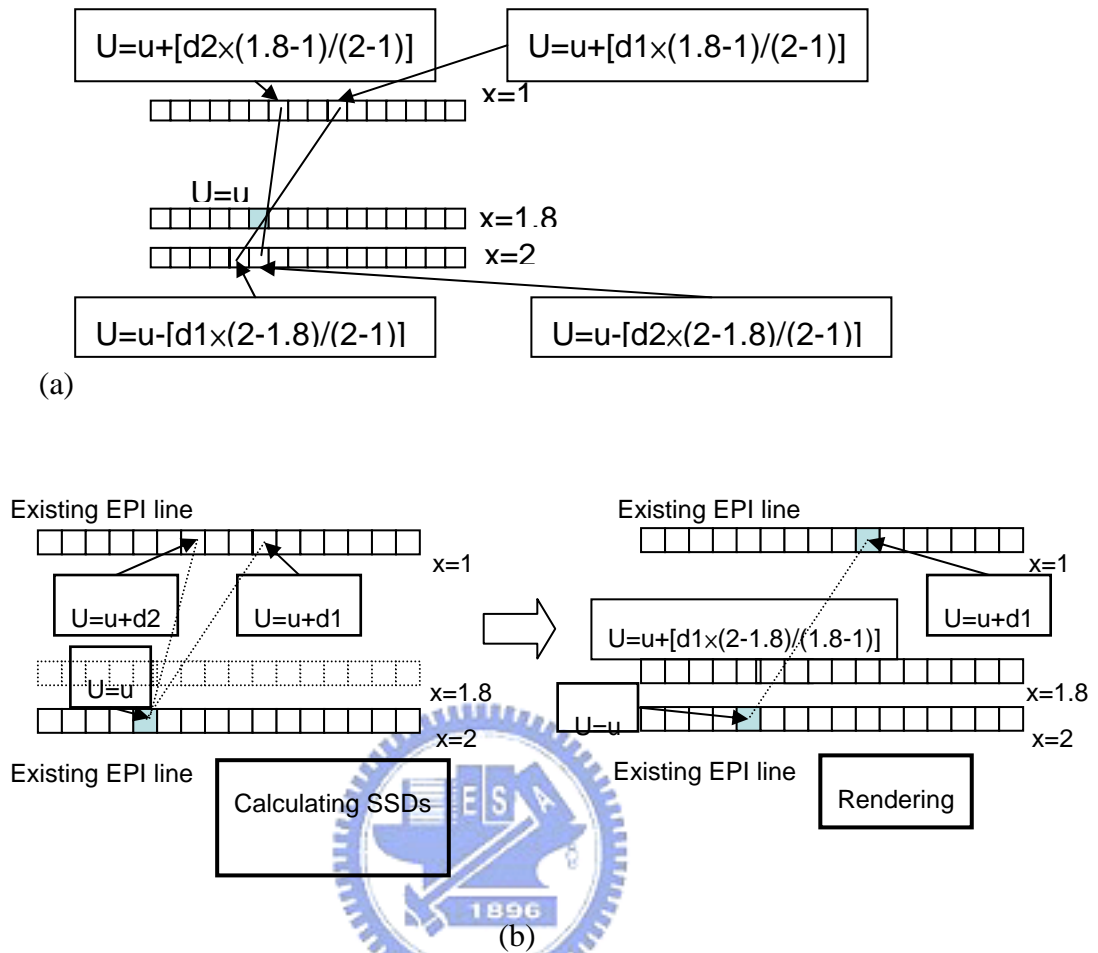
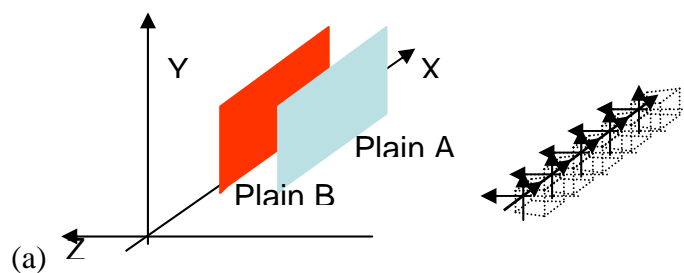


圖 3.2: (a) 常見的一次性 rendering 架構 (b) 階段性的 rendering 架構

之後我們以此程式測試了各種基本的立體匹配的技術套用於 image-based rendering 並評估其效果。測試的基本方法有基於畫素的立體匹配、基於視窗的立體匹配(視窗寬度=11)。圖 3.3(a)為輸入影像為 blender 搭建的簡單場景。3.3(b)至(f)分別為輸入影像、ground truth、以及各種基本技術所得的左邊輸入影像當中各畫素的視差值乘以 11 為灰階值所作的視差圖。



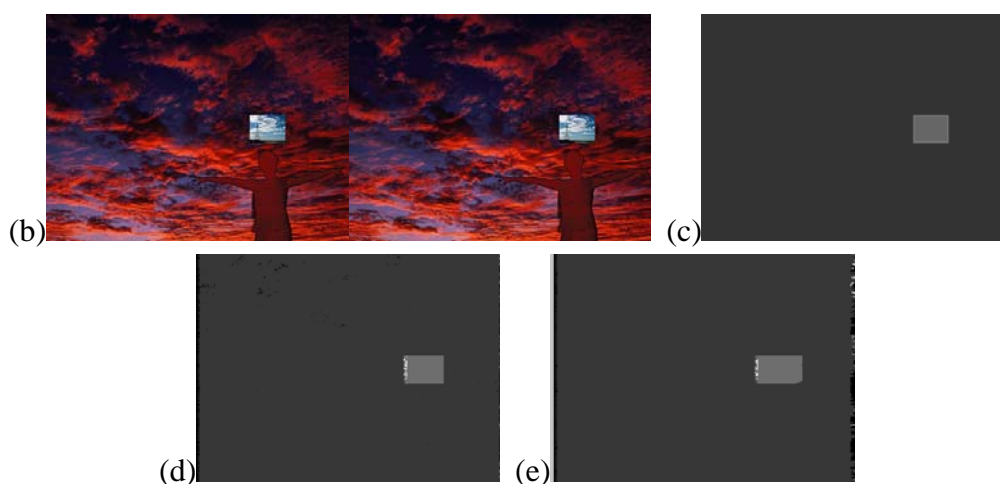


圖 3.3: 基本立體匹配套用於基於影像的繪製效果: (a) 場景示意圖 (b) 輸入影像(共兩張) (c) ground truth (d) 基於畫素的立體匹配 (e) 基於視窗的立體匹配

與 disparity ground truth 資料比較的結果，發現了以下三種誤差問題。

1· 基於畫素的立體匹配在較遠的平面上紋理變化不明顯的區域上有零星的胡椒鹽雜狀訊(pepper noise)。

2· 另外兩種方法由於須參考的影像範圍較大，在畫面的最左與最右端無法計算視差值而產生錯誤。

3· 最後一個值得注意的是在較近的平面 A 的兩側，也就是所謂的遮蔽區域。三種方法在前物左的側遮蔽區域皆發生錯誤，而兩種參考的影像範圍較大的方法則是連右側都發生錯誤。

我們特別將第三種情況提出來討論，如圖 3.4 所示。對於位於平面 A 左側的遮蔽區域內的畫素，右邊輸入影像本來就不存在與它匹配的畫素。在資訊不足的情況下無法正確的估算視差值是很合理的。而在 A 平面右側的遮蔽區域中，互相匹配的畫素其實是存在的，但是由於視窗範圍跨越了兩個實際上視差值不相同的區域，而產生干擾影響了估算的結果。

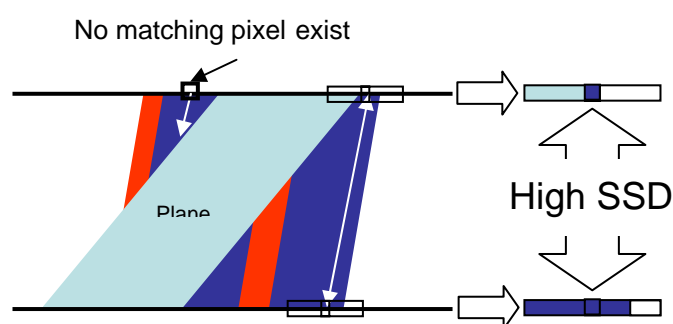


圖 3.4: 遮蔽區域的干擾示意圖

3.2 推廣至多張輸入影像的架構

由上一節我們發現了傳統立體匹配的兩張輸入影像架構在遮蔽區域產生的問題，但 FTV 系統不同於傳統系統，它擁有數十台攝影機。若一次考慮多張輸入影像，則可望增加有用的資訊量而增進估算準確度。

在架構推廣之前，我們設定了一個基本假設：場景中的任何一個物體至要要能同時被兩台相鄰的攝影機拍攝得到。也就是說，我們並不考慮像是圖 3.5 (a)，淺藍色物體出現在兩紅色物體所呈的狹縫之中無法計算視差的情況。因為對於每台攝影機相距僅數公分的 FTV 系統，只要狹縫不要太窄，淺藍色物體就能夠同時被數台攝影機拍攝到。因此這個假設是合理的，如圖 3.4 (b)所示。

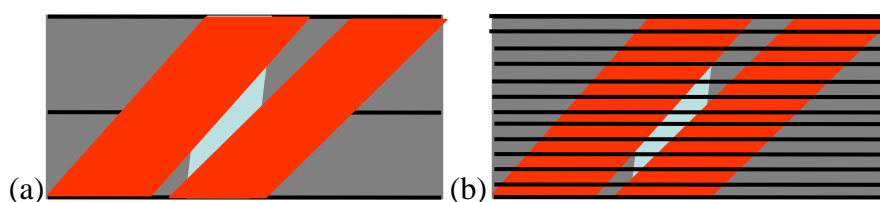


圖 3.5: (a) 假設不存在的狀況 (b)實際 FTV 系統示意圖

有了上一段的假設，我們可以確保每一張輸入影像當中的每一個畫素都能找出視差值。以圖 3.6 為例，若要內插 L3 與 L4 之間的區域，那麼根據圖 3.2(b) 的 rendering 架構，我們必須知道 L3 與 L4 所有畫素的視差值。

基於圖 3.5 所示的假設我們可以推得。L3 上的物體，若在 L2 上被遮蔽則必定會出現在 L4 上。若在 L4 上被遮蔽則必定會出現在 L2 上。也就是說若欲估算

出 L3 上所有畫素的視差，L2、L3、L4 包含了足夠的資訊。同理若要估算出 L4 上所有畫素的視差，L3、L4、L5 包含了足夠的資訊。因此若要內插 L3 與 L4 中間的區域，L2、L3、L4、L5 四條極線的資訊已經足夠。至於是否須要參考更多條極線？第一，如無雜訊或是光滑反射面反射造成反射光線不均勻等問題。四條極線已經包含非常豐富的資訊，考慮更多的極線除了不能得到更多可利用的資訊，還增加了時間複雜度。第二，我們並不確知一個物體能否同時出現在數條極線中而不被遮蔽，所以考慮更多條極線未必能減少遮蔽區域造成的干擾。目前的實驗，我們選擇以四張輸入影像的架構來取代原本兩張輸入影像的架構，而所得的結果頗佳。

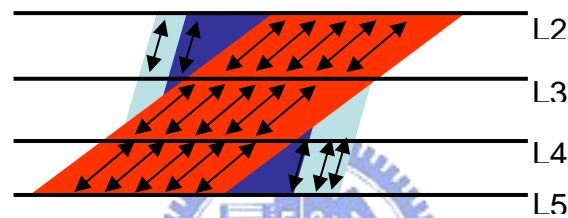


圖 3.6: 四張輸入影像的架構

我們把圖 3.6 所示的四張輸入影像的架構套用進圖 3.1 當中重做一次實驗，結果如圖 3.6 所示。我們可以發現基於畫素的立體匹配方法在遮蔽區域有了顯著的改善，然而胡椒鹽狀雜訊還是存在。而另外兩種方法在遮蔽區仍然發生了錯誤。

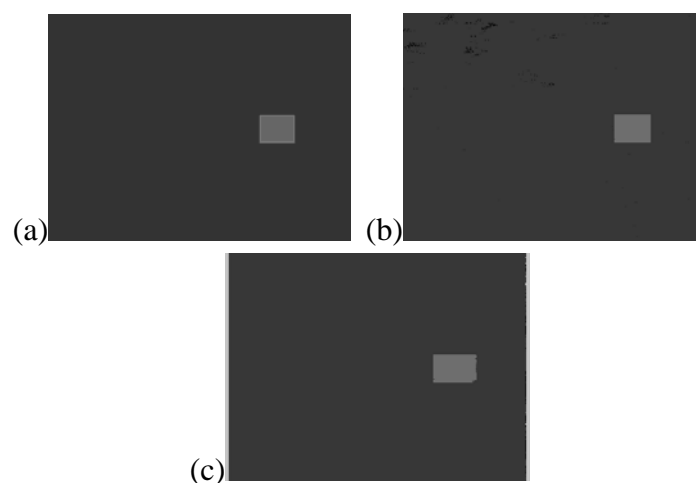


圖 3.7: 四張輸入影像的架構產生的視差圖: (a) ground truth (b) 基於畫素的立體匹配 (c) 基於視窗的立體匹配

圖 3.8 是基於視窗的立體匹配方法在四張輸入影像架構之下的遮蔽區域示意圖。我們可以發現，雖然四張輸入影像架構有了足夠的資訊，但是仍然沒辦法減少於基於視窗的立體匹配方法所受到的干擾，在遮蔽區域中真實的視差值依然會得到很大的 SSD。

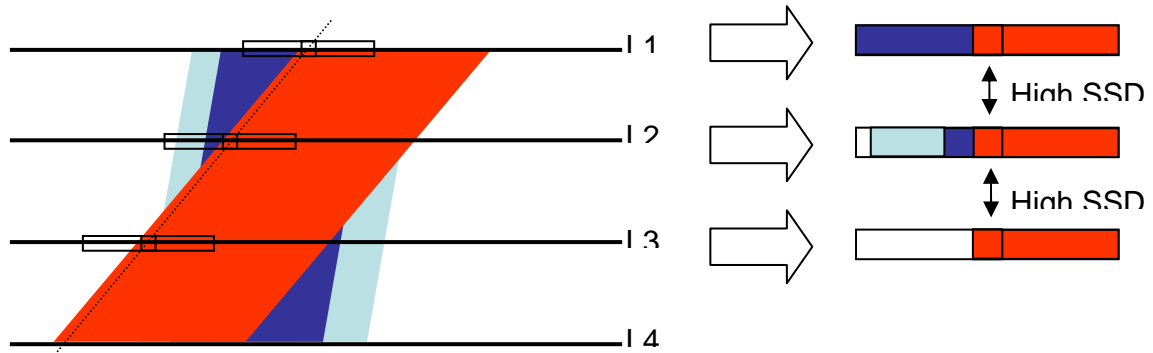


圖 3.8: 基於視窗的立體匹配方法在四張輸入影像架構之下的遮蔽區域示意圖

3.3 單邊視窗對立體匹配

跟據圖 3.6 我們發現，在四張輸入影像架構之下，資訊不足的問題已經解決。但是如圖 3.8 所示，因為遮蔽區域視差值突然的變化，對基於視窗的立體匹配方法產生的干擾還是存在的。基於視窗的立體匹配的同時考慮視窗範圍內的畫素的匹配情形的策略雖然在平滑無紋理區域有較佳的表現，卻會在遮蔽區域附近受到左半邊或右半邊的干擾資訊。因此我們希望找到一個新的立體匹配方法來解決這個問題。

這個新方法若要保有基於視窗的立體匹配方法在平滑區的可靠度，那麼必定要像舊方法一樣同時考慮某個範圍內多個畫素的匹配情形的特性。若要像基於畫素的立體匹配方法一樣在遮蔽區不受干擾，那麼就必須要能夠適時的迴避左半邊或右半邊的干擾資訊。

基於上述兩個想法，我們把基於視窗的立體匹配拆成左右兩個視窗，並且如 (2.11) 式所示，給予畫素中每個畫素不同的權重以增加鑑別度。為了能回避掉視差不連續的半邊干擾資訊，在 SSD 的計算上我們的策略是分別計算左右兩個半

邊的視窗的 SSD。乎略 SSD 值較大的，有可能是受到干擾的半邊，保留值較小的 SSD 值做為判斷是否匹配的依據。我們把這種新的立體匹配方法稱為**單邊視窗對 (SWP)**的立體匹配方法。

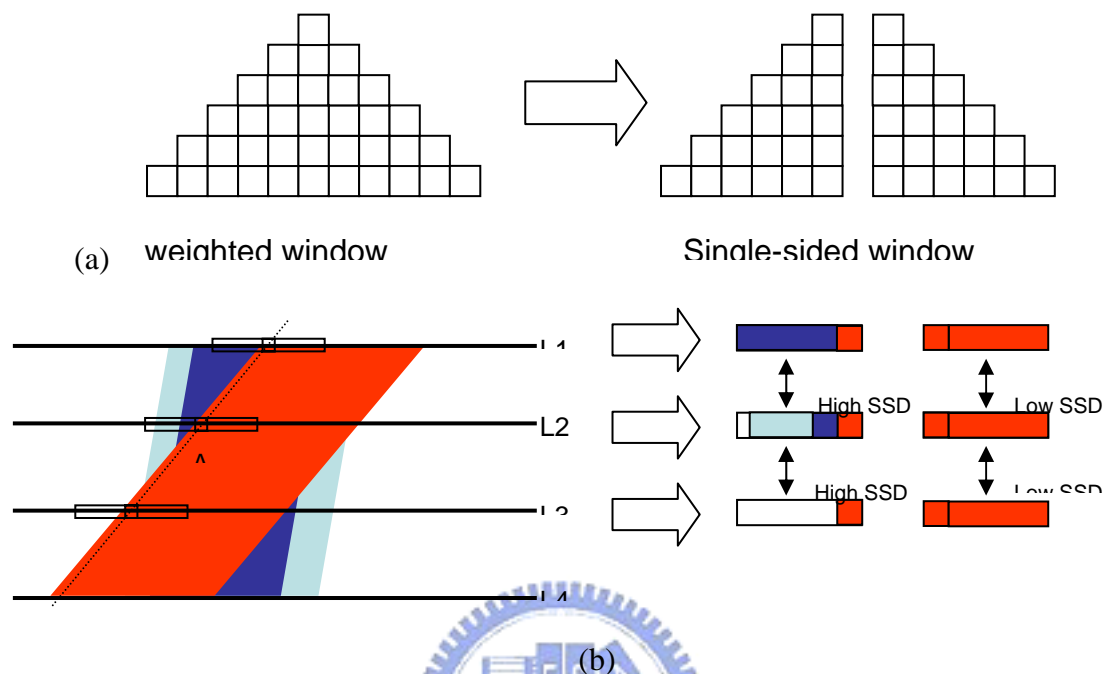


圖 3.9: SWP: (a)六階層 SWP 的示意圖 (b)SWP 迴避遮蔽區干擾示意圖

我們把 SWP 套用進圖 3.1 的程式架構取代基本的立體匹配方法，分別搭配兩張以及四張輸入影像做測試。效果如圖 3.9 所示，我們可以發現在兩張輸入影像的架構之下，SWP 在遮蔽區域中依然沒有足夠的資訊，表現大約與基於畫素的立體匹配法相同，然而 SWP 在影像的左右兩邊不會像基於視窗的立體匹配一樣，發生無法計算 SSD 的情形。在平滑區域也沒有像基於視窗的立體匹配一樣的胡椒鹽雜訊，整體而言是稍微優於幾種基本立體匹配方法的。但如果搭配四張輸入影像架構的話，就會有顯著的優勢，圖 3.10 (b)中 SWP 所得的視差圖幾乎與 groundtruth 完全吻合沒有錯誤發生。

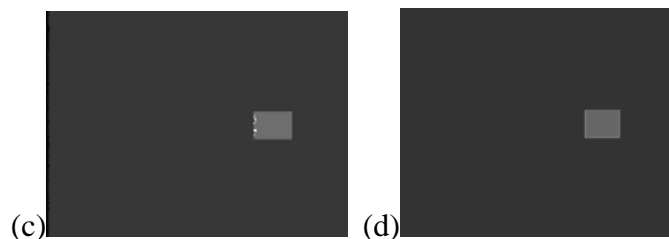


圖 3.10: SWP 產生的視差圖: (a) SWP 搭配兩張輸入影像所得的視差圖 (b) SWP 搭配四張輸入影像所得的視差圖

最後我們比較了一下各種方法的複雜度。雖然 SWP 與基於視窗的立體匹配方法同時考慮了較多的畫素，甚至每個畫素的權重不盡相同。然而如果先將所有的 $SD(u, d)$ 計算出來儲存於記憶體當中，它們不會比基於畫素的立體匹配方法耗費較多的乘法。只須要額外安排階層式的加法運算就能實現。圖 3.11 是以加法取代乘法的 SWP 運算架構。

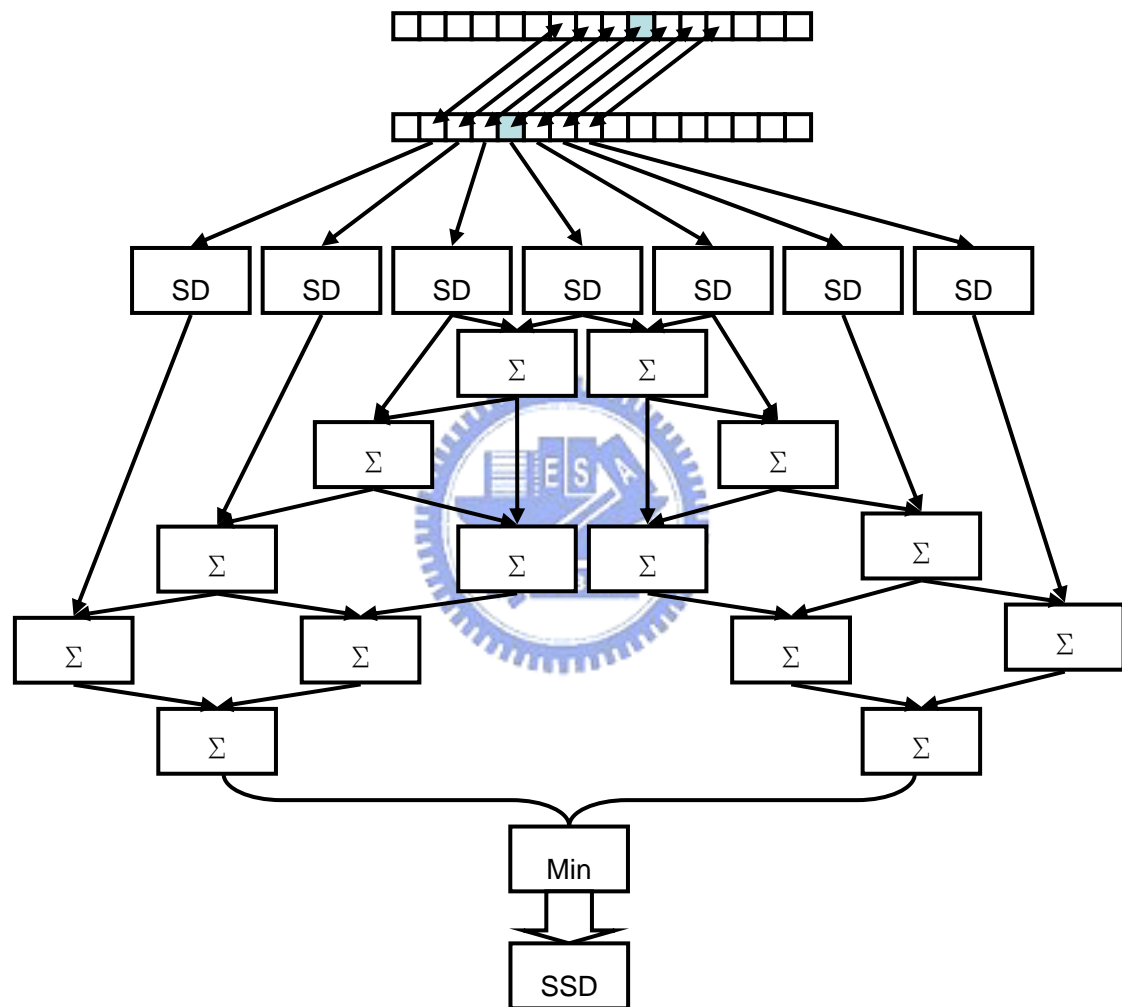


圖 3.10: 以加法取代乘法的 SWP 運算架構

表 3.1 簡單的比較了各種方法搭配 WTA，平均判斷一個畫素的視差值所須要各種運算子的個數。須注意的是 branch 指令那一列中出現的 D 是所有可能的視差值的個數，在我們的實驗中它的值是 22，另外表 2.1 中基於視窗的立體匹配同時考慮了十一個畫素，六階層的 SWP 同樣考慮十一個畫素並將考慮的範圍

拆解成左右兩個六個畫素的視窗，如圖 3.9 (a)所示。藉由表 3.1 我們可以發現 SWP 比起基於視窗的立體匹配只有在加法以及 branch 兩個較不費時的指令上多了一倍，因此還是很有希望開發一個即時的 FTV 系統。

表 3.1: 各種方法估算單一畫素視差的複雜度比較表

	Pixel-based	Window-based (window size=11)	SWP (6-level)
Addition, subtraction	5	15	27
Multiplication	3	3	3
Branch	D	D	D*2
Memory consuming	1	6	12



第四章

動態規劃與總體最佳化

雖然 SWP 在第三章的簡單測試當中有良好的表現。然而那只是一個視差變化簡單，物體紋理複雜的場景，這在真實環境當中並不常見。於是我們把 data set 中的兩組場景放到程式中進行實驗。結果如圖 4.1 所示。

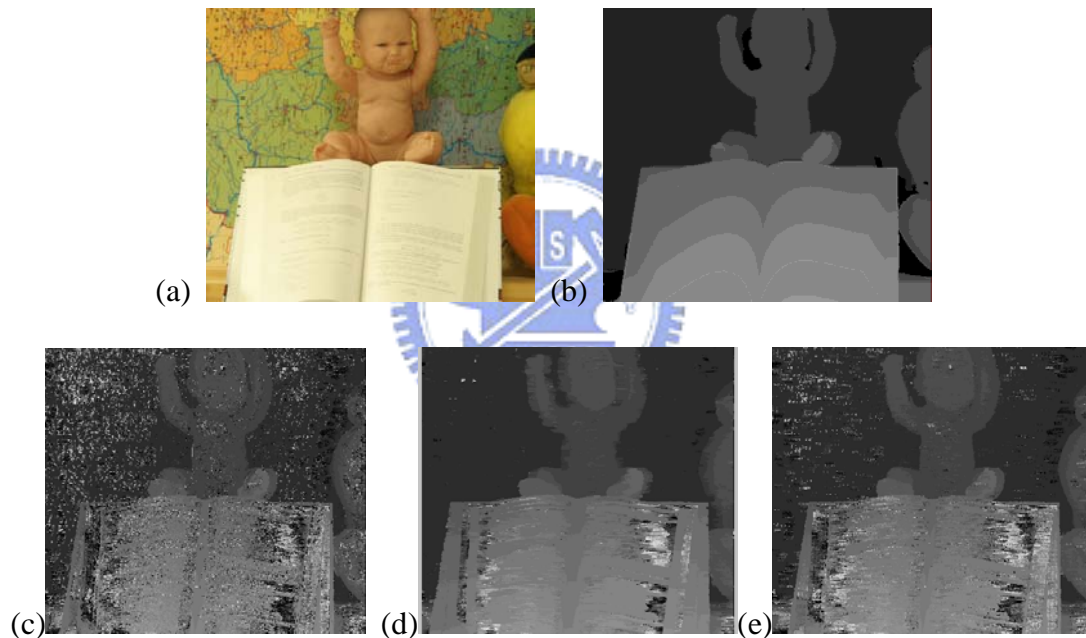


圖 4.1: 真實場景的視差圖: (a) 輸入影像 (b) ground truth (c) 基於畫素的立體匹配 (d) 基於視窗的立體匹配 (e) SWP

基於畫素的立體匹配在實際狀況之下的 pepper noise 比起 blander 搭建的簡單場景要嚴重許多。而基於視窗的立體匹配以及 SWP 雖然分別同時參考了十一個以及六個相鄰畫素的匹配情形，但是對於一張寬達四百畫素的輸入影像來說還是不足以準確判斷紋理不明顯的區域的視差值。表 4.1 是輸入影像當中(300,240)位置，也就是輸入影像當中書本右上角無紋理區域的某一個畫素。由表 4.1 的數據可以發現各種方法求出的 SSD 列表。包括 ground truth 的實際視差值 11 在內，

所有可能的視差值以三種方法計算都得到極小的 SSD 值。

表 4.1: 圖 4.1 (a)中無紋理區域內某畫素的 SSD 數據

SSD \ Methods Disparity	Pixel-based	Window-based	SWP
0	0	0	0
1	3	9	19
2	3	14	23
3	2	13	18
4	3	13	24
5	3	10	21
6	3	13	18
7	3	14	21
8	3	19	23
9	3	22	24
10	3	20	24
11	5	22	36
12	6	22	19
13	3	22	24
14	4	23	25

有別於遮蔽區域內，實際的視差值得到極大的 SSD 而被判斷為不匹配，面對這種各種視差值都得到很小 SSD 的困境，人們一般有兩種策略來應付，一是以適應性視窗立體匹配法(A Stereo Matching Algorithm with an Adaptive Window) [3]來得到對於視差值更有鑑別度的 SSD，二是以動態規劃(dynamic programming)來加速總體最佳化(global optimize)的過程，協助視差圖最後的決策，使得無紋理區域當中有較平緩的視差估算結果。然而前者在視窗範圍評估的步驟中，大量運用了諸如開平方根之類的浮點運算子。考慮到未來即時性系統的發展，我們決定採用以動態規劃加速的總體最佳化，來改善無紋理區域的視差估算結果。

在本章中我們先介紹了動態規劃的基本想法與流程。接著實作了能夠對(2.12)式進行最佳化的動態規劃函式，並將其套用到圖 3.1 的程式架構中取代 WTA 進行視差值的決策。因為效果不盡理想，而且不能同時對基於影像的繪製所須的兩條極線上的視差值做最佳化。於是我們又參考了 G. Van Meerbergen 所提出的，

使用動態規劃的階層式對稱立體演算法(A Hierarchical Symmetric Stereo Algorithm Using Dynamic Programming) [4]。並修改它，加上了以四張輸入影像架構修補演算法中無法判斷的遮蔽區域，最後得到與 ground truth 非常吻合的視差圖。

4.1 動態規劃的基本想法與流程

基本上我們可以把動態規劃想像成資料結構中，graph 的最短路徑規劃問題。被規劃的 graph 須要有以下的特性：

- 1.具有一個起點與一個終點以及許多分布於起點終點之間的節點。
- 2.各個節點都有若干個節點直接通往它，也都直接通往其它若干個節點。
- 3.節點與節點之間的距離是固定已知的，連結是單向並且沒有迴圈發生。

假如 graph 具有這些特性，我們就可以用動態規劃來找出最短路徑。動態規畫的基本想法是，假如我們知道從起點到達每一個可以直接通往終點的節點的最短路徑與路程，那麼我們只要把到達這些節點的最短路程與這些節點與終點之間的距離相加，從中找出最小值。便可以找出從起點到達終點的最短路徑與路程。

接下來我們就可以把原本的問題分解成從起點到達每一個可以直接通往終點的節點的最短路徑與路程的動態規劃子問題。每個子問題都利用上一段所敘述的方法分解到最後，最基本的子問題就變成了從起始點到其可以直接通往的相鄰節點的距離，是較容易處理的問題。最基本的子問題解決了以後，我們只須要尋序的找出包含它們的母問題的最短路徑與路程，最後便可以找出從起點到終點的最短路徑與路程。

4.2 基本的總體最佳化概念

基本的總體最佳化是以去除無紋理區的胡椒鹽狀或斑狀雜訊為出發點，而提出(2.12)所示的函式並且以動態規劃的方式將它最小化。基於時間複雜度的考量，在實作上人們總是把單獨一條掃描線而非整張影像視為總體，而對單獨對一條掃描線進行最佳化決策。有的演算法會把已最佳化過的相鄰的掃描線的決策結

果當成參考的基準。

在此我們先考慮最佳化的總體目標，如下式(2.12)所示。

$$E_{global} = \sum_{x=0}^{x=w-1} E_{data}(x, d(x)) + \lambda E_{smooth}(x, d(x)) \quad (2.12)$$

將 E_{global} 最小化的工作其實就是求出 $E_{data}(x, d(x))$ 與 $\lambda E_{smooth}(x, d(x))$ 總和的最小值。其中 $d(x)$ 是各畫素決策的視差值。 $E_{data}(x, d(x))$ 為各畫素的匹配代價，在我們的程式當中採用的是最簡單的 SSD。

$$E_{data}(x, d(x)) = SSD(x, d(x)) \quad (4.1)$$

而 $E_{smooth}(x, d(x))$ 則是所謂的不連續代價(discontinuity cost)。通常是相鄰的畫素決策的視差之差值的平方。

$$E_{smooth}(x, d(x)) = (d(x) - d(x-1))^2 \quad (4.2)$$

λ 是一個可調的平滑化參數，大小取決於程式設計者偏向較平緩的變化或是較低的匹配代價。對各個畫素來說它不一定是個固定的值。在我們的程式當中， λ 取決於相鄰畫素間 RGB 差值的絕對值。若 RGB 差值的絕對值大於某個門檻，則代表該畫素位於一個有紋理的區域，那麼我們不須要 $E_{smooth}(x, d(x))$ 的幫助也能夠做出正確的決策，所以將 λ 設為零。若 RGB 差值的絕對值小於某個門檻則代表該畫素可能位於無紋理區域內，則將 λ 設為一個非零的定值 K，在我們的程式中，六階層的 SWP 搭配 $\Lambda=20$ 能夠得到較佳的效果。

$$\lambda = 0 \quad \text{if } |R_x - R_{x-1}| + |G_x - G_{x-1}| + |B_x - B_{x-1}| > threshold$$

$$\lambda = K \quad \text{if } |R_x - R_{x-1}| + |G_x - G_{x-1}| + |B_x - B_{x-1}| \leq threshold \quad (4.3)$$

E_{global} 可以視為 $d(x)$ 的函數，維度跟一條掃描線的畫素數一樣高，不靠動態規畫的幫助最佳化工作是不可行的。要將 E_{global} 最小化就必須把一條掃描線上各畫素的視差值決策改寫成符合 4.1 所提出的三個條件的 graph。因此，如圖 4.2 所示，我們把各個畫素所有可能的視差值決策都視為一個節點。然後再訂定虛擬的起始點與終點，起始點連接第一個畫素的每個節點，之後每個節點連接下一個

畫素的所有節點，最後一個畫素的所有節點連接終點。相鄰節點間的距離則是 $E_{data}(x, d(x)) + \lambda E_{smooth}(x, d(x))$ ，計算細節如(4.1)、(4.2)、(4.3)所示。

如圖 4.2 所示，直接與起始點相連的節點為第一個畫素的所有節點。也就是整個最佳化問題最基本的子問題。到達這些節點的路徑都是唯一的，路程分別是它們的 SSD 值。第一個畫素的節點路程計算完了之後，到達第二個畫素的節點的最短路徑路程便可以計算得出來了。依照各個畫素的順序逐步推算之後我們便可以求出整條掃描線的最佳視差決策。

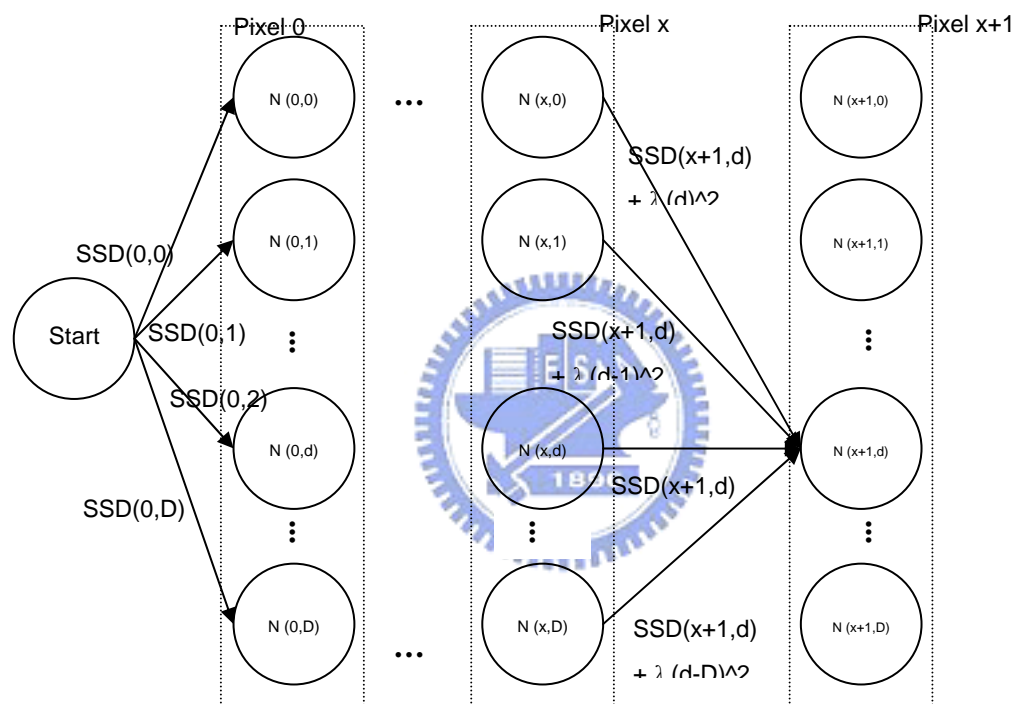


圖 4.2: 基本的總體最佳化概念節點圖

我們依照本節敘述的流程實作了一個動態規劃函式，把它套用進圖 3.1 所示的程式架構，取代 WTA 來幫助視差圖估算，並且以 SWT 計算的 SSD 做為匹配代價。圖 4.3 是視差圖估算的結果，我們可以比較藉改變(4.3)式中的 K 值大小所造成結果的差異。

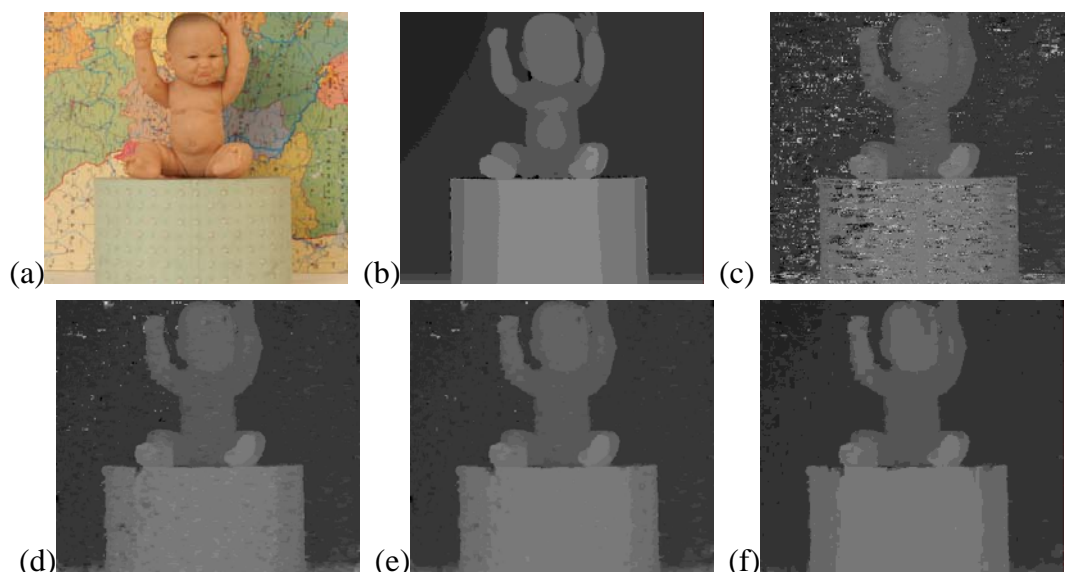


圖 4.3: 基本總體最佳化的視差圖估算結果: (a)輸入影像 (b) ground truth
(c) $K=0$ (WTA) (d) $K=10$ (e) $K=20$ (f) $K=40$

由圖 4.3 我們可以發現總體最佳化的確有助於得到較平整的視差圖，平整的程度依照平滑參數的強度而增加。但是平滑參數並不能無限度的增加，從圖 4.3 我們可以發現嬰兒右腳邊的區域，由於前景與背景的色彩差異不大，平滑參數不會為零，而使得背景的視差影響到前景造成錯誤，影響範圍隨著平滑參數的增加而擴大。另外，如圖 4.4 所示。用來幫助 rendering 的兩張視差圖，各別進行最佳化。因此兩張圖在無紋理區域中的視差值並不是很匹配，容易造成輸出影像的斑駁或某些畫素沒有被填入資料的情形。因此我們參考了 G. Van Merbergen 所提出的，使用動態規劃的階層式對稱立體演算法當中的對稱概念，期望得到較佳的效果。[4]

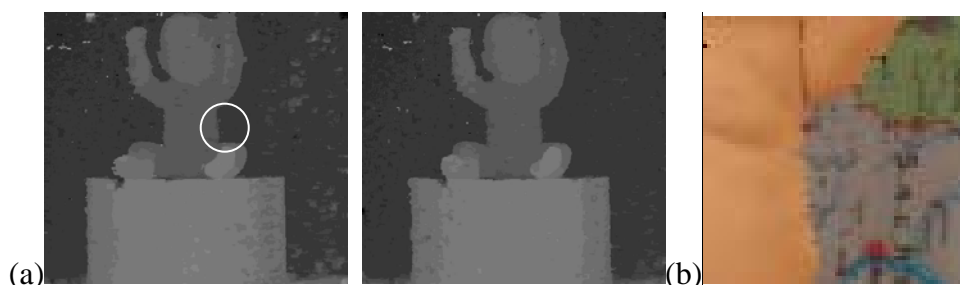


圖 4.4: 視差圖的不匹配以及 rendering 不理想的情形:
(a) 匹配狀況不佳的視差圖 (b) 不理想的輸出影像

4.3 對稱立體演算法

對稱立體演算法是一種對兩張輸入影像的立體匹配問題做最佳化的演算法。由於基本總體最佳化只針對一張視差圖的匹配代價以及不連續代價做最佳化，因而可能產生如圖 4.5 所示的情況：畫素 A 經過最佳化之後判斷畫素 B 為最有可能與它匹配的畫素，但是以畫素 B 的觀點，在 L1 中還存在著比畫素 A 更有可能匹配的畫素。而對稱立體演算法針對這種失誤做了改善。藉著幾項嚴格但實際的限制，對稱立體演算法能夠有效的確保兩張輸入影像之間尋找每一對最有可能匹配的畫素。而面對兩張輸入影像架構中無可避免的遮蔽區域問題，對時垃圾演算法的策略是將遮蔽區域偵測出來加以標記，並不估算視差。將對稱立體演算法推廣至四張輸入影像的架構，我們的做法分成兩階段：第一階段先以兩張輸入影像的架構找出各對匹配的畫素以及無法匹配的遮蔽區域。第二階段再以四張輸入影像的架構修補被偵測並標記為遮蔽區域的畫素的視差值。

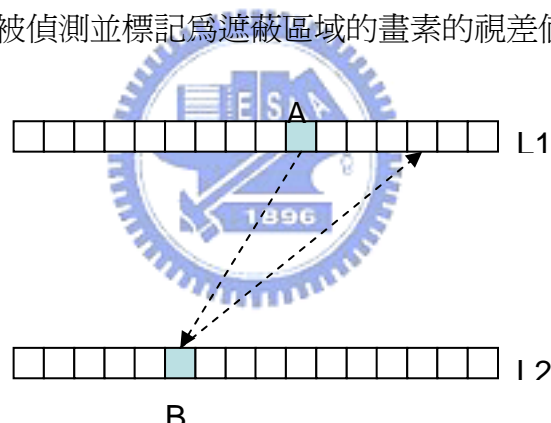


圖 4.5: 以基本總體最佳化尋找匹配點的不對稱狀況

4.3.1 對稱立體演算法的基本限制

兩張輸入影像的對稱立體演算法的基本理論乃建構於下列幾條限制之上：

1. 可能的視差值落在一特定範圍之內。
2. 一條極線中的一個畫素最多只能和另一條極線中的一個畫素匹配。
3. 如圖 4.6 所示，A 為極線 L1 中的一個畫素，B 為 L1 當中位於 A 左側的另一畫素，若 A 與 L2 中的 C 畫素匹配，則 B 不可能與 L2 中 C 右邊的畫素匹配。
4. 如圖 4.7 所示，若在極線 L1 中產生一遮蔽區域，A 與 B 為該區域邊緣

的兩個未遮蔽的畫素，則 L2 中與 A、B 匹的兩個畫素必定相鄰。

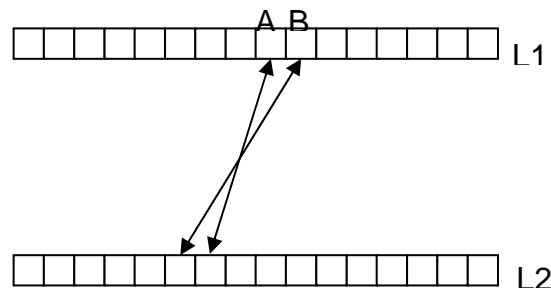


圖 4.6: 對稱立體演算法中第三條限制下不允許發生的情況

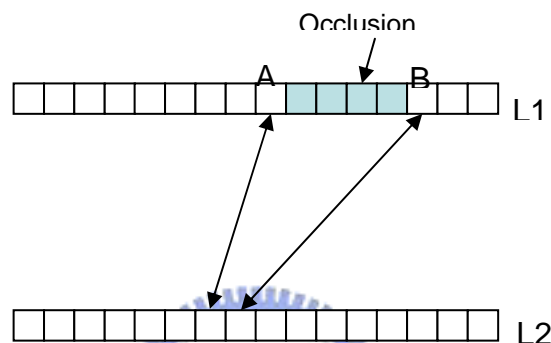


圖 4.7: 對稱立體演算法中第四條限制的示意圖

4.2.2 以動態規劃實現對稱立體演算法

和基本的總體最佳化概念一樣，對稱立體演算法也針對了一條掃描線列出一條函式，並且將它最小化：

$$E_{global} = \sum_{x=0}^{x=w-1} E_{data}(x, d(x)) + E_{occlusion}(x) \quad (4.4)$$

(4.4)式當中的 $E_{data}(x, d(x))$ 與(2.12)是完全相同的，而 $E_{occlusion}(x)$ 則是所謂的遮蔽代價(occlusion cost)。除了輸入影像兩端的畫素必定無法找到匹配的畫素，所以不必付出遮蔽代價之外，對於其它每一個受到遮蔽的畫素，遮蔽代價為一定值。雖然如圖 4.7 所示，一條極線上越長的遮蔽區域對應到另外一條極線就是越大的視差不連續，然而與遮蔽區域長度平方成正比的遮蔽代價，在將(4.4)最佳化的目標之下會導致如圖 4.8 所示的錯誤，所以我們給予每個被遮蔽的畫素一個固定值，使得總體的遮蔽代價隨著遮蔽畫素數線性成長。

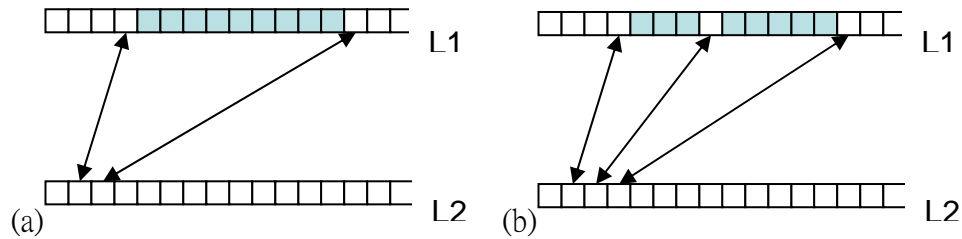


圖 4.8: (a)實際的遮蔽狀況 (b) 遮蔽代價非定值可能產生的錯誤。

接下來我們將對稱立體演算法的最佳化以動態規劃的方式實現。對稱立體演算法的節點圖與圖 4.2 所示的基本總體最佳化的節點圖非常類似。一樣是以各個畫素的所有可能的視差值決策做為節點。差別在於各節點連接的方式與距離的計算。如圖 4.9 所示，考慮以 L1 為基準，從左邊出發的對稱立體演算法。一開始必定會先遇到一段無法匹配的區域之後，A 畫素才與 L2 的第一個畫素匹配。若在 L2 發生遮蔽，以 L1 的觀點而言相當於連續的畫素之間產生了視差值的遞減，遞減量可以為任意值。若在 L1 發生遮蔽，則可以視為不相鄰的畫素間產生視差值遞增，遞增量為兩畫素之間的距離減一。

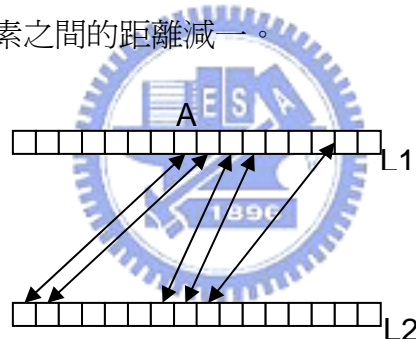


圖 4.9: 以 L1 為基準的立體匹配示意圖

跟據上一段。欲在 L1 的某畫素做了一個視差值的決策，則必定是上一個畫素做了大於或等於此畫數的視差值決策，或是前幾個畫素做了小於它的視差值決策。這個結論就是對稱立體演算法在動態規劃的節點圖中各節點之間連結建立的依據。依此我們得到了圖 4.10 的連結示意圖。

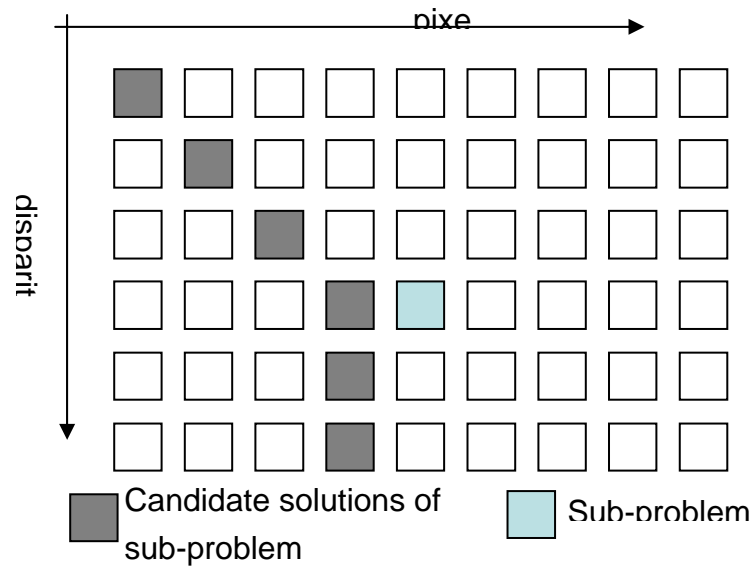


圖 4.10: 對稱立體演算法的子問題與其可能的解

4.2.3 以四張輸入影像的架構修補對稱立體演算法。

以兩張輸入影像架構搭配對稱立體演算法求出了兩張互相匹配的視差圖之後，我們所要進行的工作是改以四張輸入影像的架構修補被標記出來的遮蔽區域。這個工作基本上和四張輸入影像架構的基本的總體最佳化方法是一樣的。唯一要注意的是如圖 4.11 所示的失誤狀況。當場景複雜度較高，且攝影機解析度不足的情況下，輸入影像容易發生取樣失真。在失真的區域當中，實際的視差決策也會有很高的匹配代價。當匹配代價高於遮蔽代價的時候，動態規劃就會做出錯誤的決策。因此，為了演算法的抗誤能力，當我們進行遮蔽區域的視差值修補的時候，必須將 L1 的遮蔽區域左邊的一個畫素以及 L2 的遮蔽區域右邊的一個畫素一起修補。而圖 4.12 是對稱立體演算法與修補機制的示意圖。

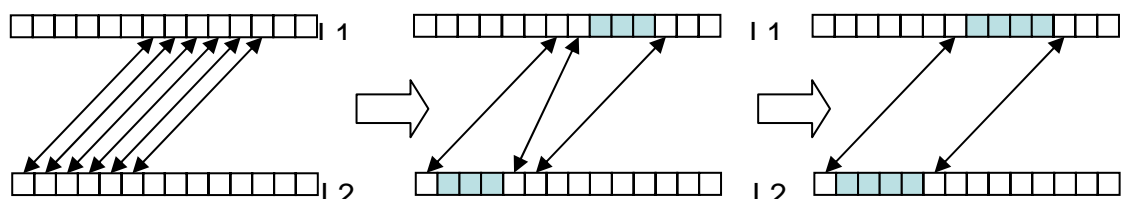


圖 4.11: 因為輸入影像失真而產生的失誤狀況與解決方法

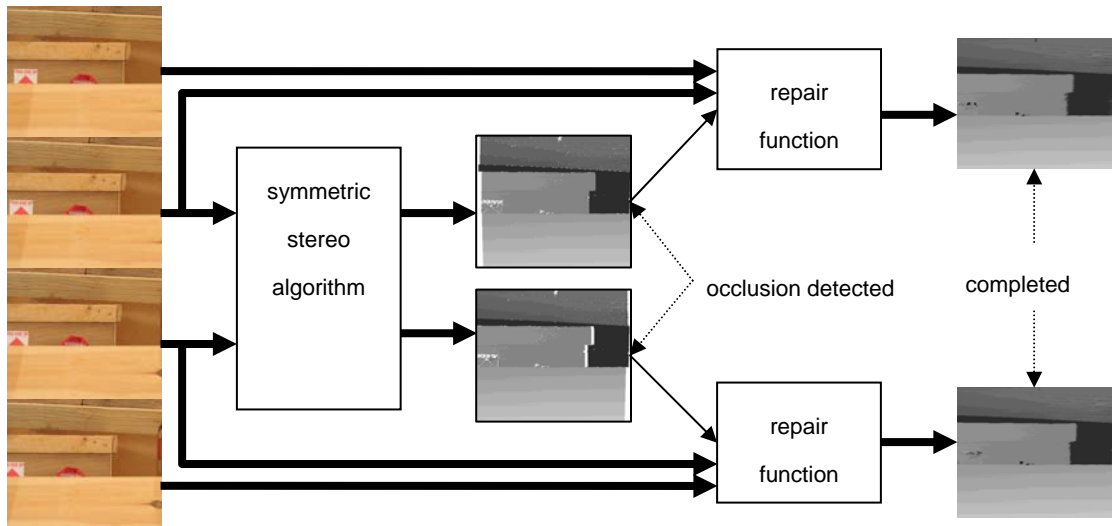


圖 4.12: 對稱立體演算法與修補機制示意圖

圖 4.13 是以 SWP 計算匹配代價、以對稱立體演算法進行總體最佳化，並且把偵測到的遮蔽區域加以標記(圖 4.12 中的白色區域)、最後再搭配四張輸入影像的架構做遮蔽區域的視差值修補。比較圖 4.12 的(c)與(e)可以發現對稱式立體演算法比起基本的總體最佳化在視差估算上的改善。

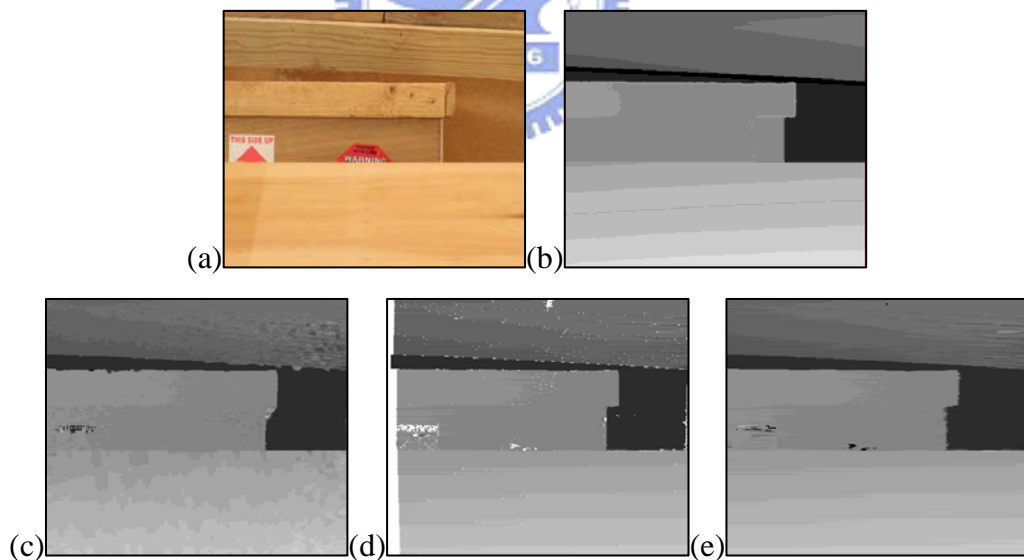


圖 4.13: 對稱立體演算法與基本總體最佳化的比較

- (a)輸入影像 (b)ground truth (c)基本總體最佳化所得到的視差圖
- (d)對稱立體演算法的最佳化與遮蔽區(畫面中的白色區域)之偵測
- (e)以四張輸入影像架構修補圖 4.13 (d)遮蔽區之後的視差圖

第五章

實驗結果

本章以一些測試影像實驗我們的演算法。並且以理論值檢驗所提的方法估算出來的視差圖以及 image-based rendering 所得到的輸出影像並與其它現有的方法做比較。以它們的誤差量來評估與比較各種方法之間的優劣。

5.1 實驗架構

本節將對我們用來進行實驗的輸入影像，以及實驗流程和各種變因加以簡介。

5.1.1 實驗用的影像

本論文的實驗採用了 Middlebury College 的立體影像進行實驗，並以他們提供的視差圖做為 ground truth 驗證我們的演算法以及現存方法對於視差估算的效能。接著又以 Blender 合成的影像做為輸入影像以及輸出影像的理論值，來測試我們的演算法與現存方法合成的影像的 PSNR。

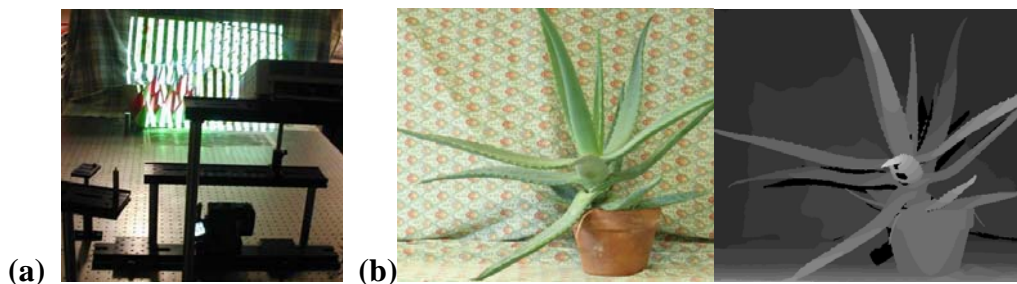


圖 5.1: (a) Middlebury College 估測視差的所使用的 structured light

(b) 某一立體影像與 structured light 找到的相對應的視差圖 [11]

特別要注意的是 Middlebury College 所提供的立體影像與視差圖。如圖 5.1(a) 所示，有別於一般視差評估的立體匹配方法，Middlebury College 的策略並非背動的截取靜態影像加以估算視差，而是主動的投射 structured light 到場景當中藉以更準確的估測場景的視差值。雖然以 FTV 系統的角度來看，這種估測視差的

點，兩個黑色角錐是輸出影像的視點，用來合成輸出影像的理論值，圖 5.3 (b) 是產生本次實驗所使用的影像的三個場景。

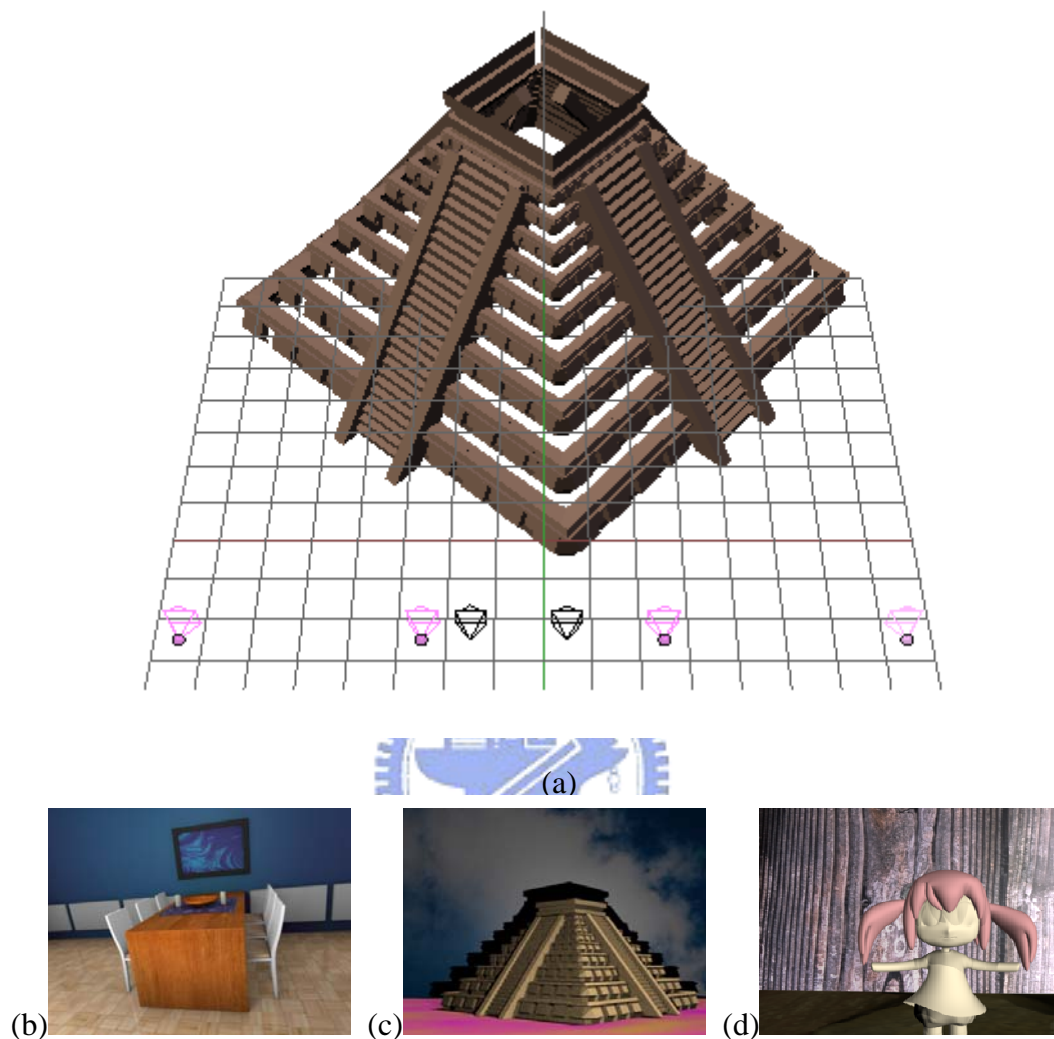


圖 5.3: (a) 以 Blender 架設場景合成輸入影像以及輸出影像的理論值

(b)場景一:會議室 (c)場景二:祭壇 (d)場景三:小女孩

5.1.2 實驗流程與變因

圖 5.4 是本次實驗的流程示意圖。基本上實驗總共分爲四個步驟，輸入影像取得、匹配代價計算、視差值決策以及 rendering。其中 rendering 的方法只有一種，因此我們每次針對一個步驟進行各種方法的比較，則總共有三組操作變因對實驗結果的影響須要探討。然而須注意的是，個個步驟之間的方法並非完全獨立可以任意更動的。例如從第四章我們可以了解，對稱立體演算法必須搭配四張輸

入影像的架構才能估算出完整的視差圖。

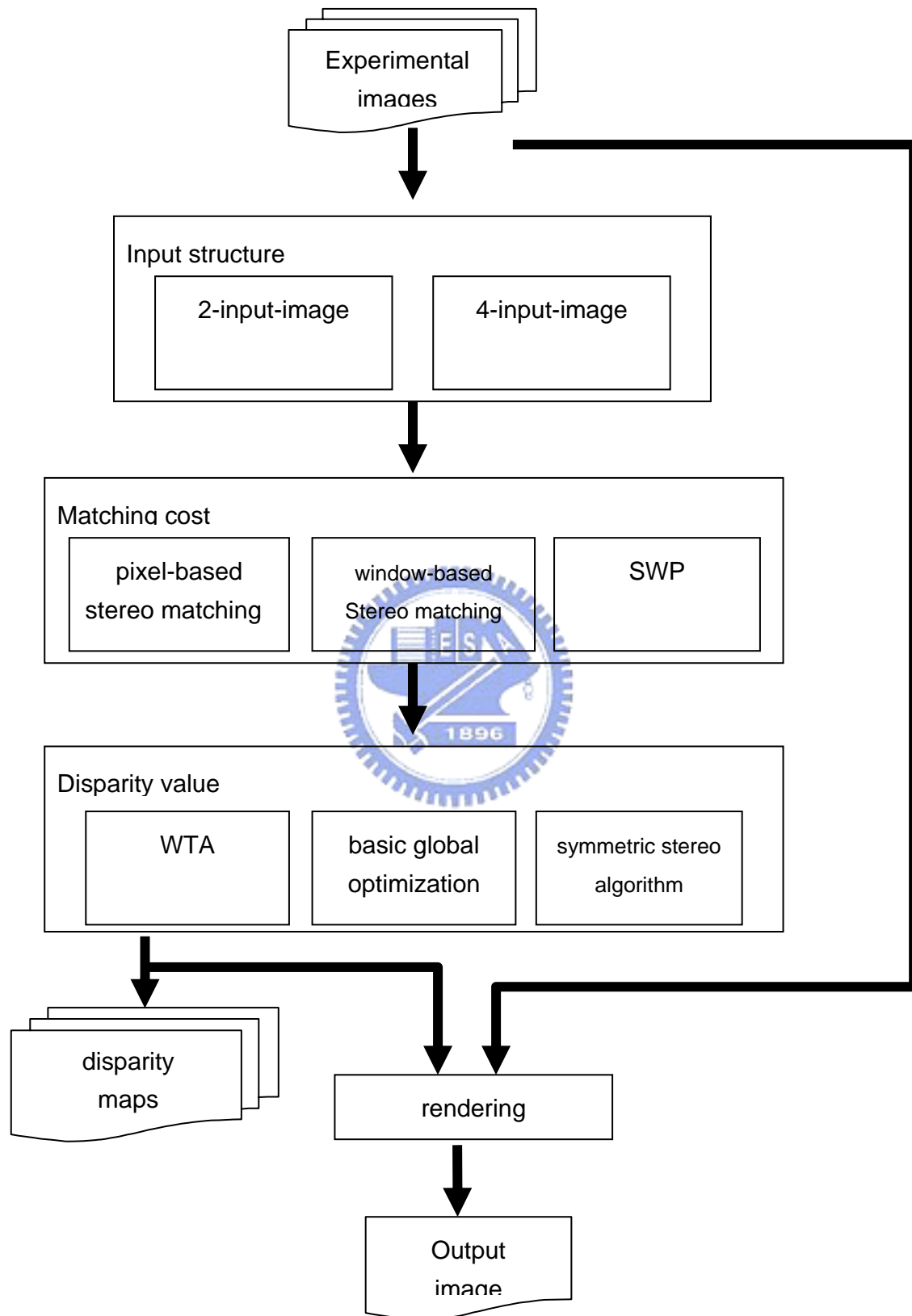


圖 5.4: 實驗流程圖

5.2 視差估算實驗結果

在這一節當中，我們採用 Middlebury College 的立體影像進行實驗。比較了採用兩張與四張輸入影像的架構、以不同的立體匹配方法計算 SSD、以 WTA 與動態規劃進行的視差值決策，對於視差圖正確性的影響。

爲了於評估一張程式估算出來的視差圖的正確性，我們的方法是採用 Middlebury College 提供的視差圖當中所有視差值非零的畫素與我們的程式所估算的視差圖當中相對應的畫素計算方均誤差。均方誤差越小則越準確。

5.2.1 由兩張輸入影像擴展至四張輸入影像架構

在第三章中，我們觀察了 EPI 在遮蔽區域的特性，並且以簡單的輸入影像對各種立體匹配方法進行測試。最後提出了以四張輸入影像的架構取代傳統立體影像域中常見的兩張輸入影像架構。因此在這一小節當中，我們以圖 5.4 中 input structure 區塊爲操作變因進行實驗，並以實驗數據證實四張立體影像在視差估算上的優勢。另外我們也比較了不同的立體匹配演算法從兩張輸入影像擴展至四張輸入影像之後，視差估算準確率提升的比例。

圖 5.5 到圖 5.7 是基於畫素的立體匹配、基於視窗的立體匹配以及 SWP 在兩張與四張輸入影像架構之下搭配 WTA 決策，在十五組輸入影像當中的視差值均方誤差所作的直方圖。表 5.1 則是三種立體匹配方法在十五組輸入影像中的均方誤差的平均值。

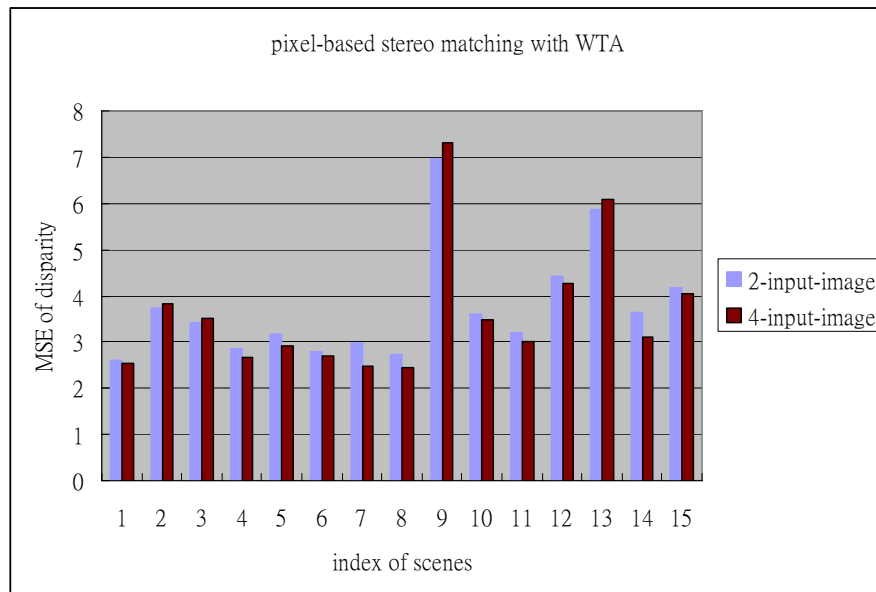


圖 5.5: 基於畫素的立體匹配搭配 WTA 估算視差圖的均方誤差

從圖 5.5 我們發現兩組資料非常的吻合。這表示若是搭配基於畫素的立體匹配方法，由兩張輸入影像擴展至四張輸入影像的架構，對視差圖的準確度幫助非常有限。原因是在兩張輸入影像的架構之下，基於畫素的立體匹配所受到遮蔽區域的干擾，本就較基於視窗的立體匹配要輕微。然而由於一次只參考一個畫素的特性，只要紋理稍微不明顯的區域，基於畫素的立體匹配就很容易發生錯誤。所以誤差偏高。

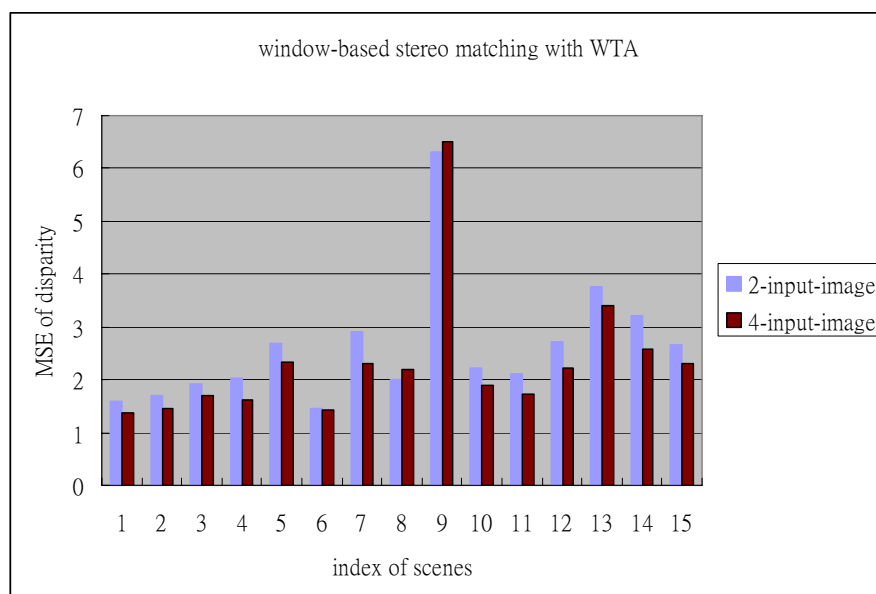


圖 5.6: 基於視窗的立體匹配搭配 WTA 估算差圖的均方誤差

比較 圖 5.6 與圖 5.5 我們發現，由於基於視窗的立體匹配由於參考的範圍

大於基於畫素的立體匹配，在無紋理區域中抗誤力較高。整體而言誤差小於基於畫素的立體匹配。然而基於視窗的立體匹配無法迴避掉半邊雜訊的特性，使得它在推廣至四張輸入影像的架構下準確度比起兩張輸入影像的架構並沒有顯著的提升。

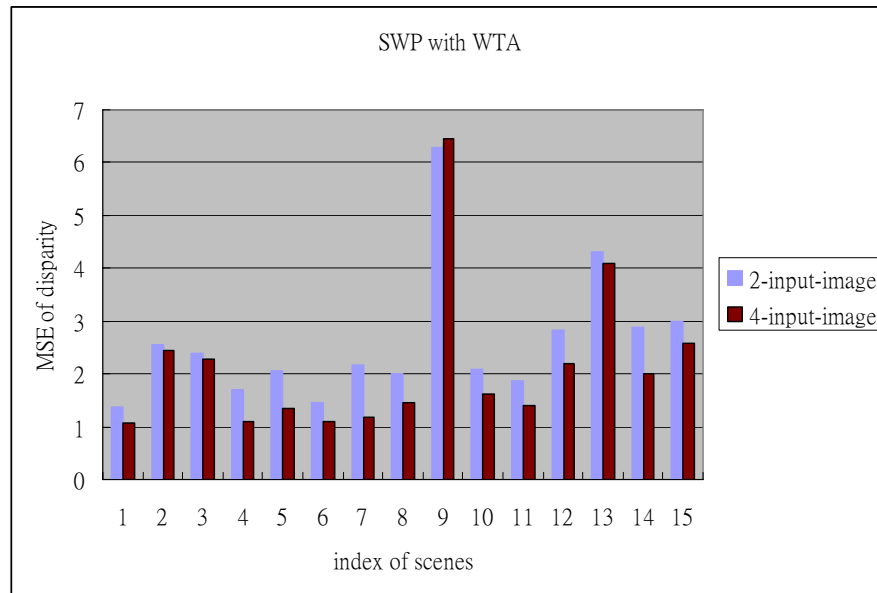


圖 5.7: SWP 搭配 WTA 估算視差圖的均方誤差

比較圖 5.7 與圖 5.6，SWP 在兩張輸入影像的架構下表現大約與基於視窗的立體匹配相同，但是在推廣至四張輸入影像的架構之後，因為 SWP 能夠有效的利用額外的資訊來估算遮蔽區的視差。所以誤差下降的比例較其它兩種方法都來得大。

表 5.1: 各場景中三種立體匹配方法搭配 WTA 的視差估算的均方誤差

	Pixel-based	Window-based	SWP
2-input-image	3.744	2.618	2.595
4-input-image	3.630	2.338	2.154

由表 5.1 的數據我們發現，就整體表現而言，基於畫素的立體匹配是誤差最大的，不適合搭配 WTA 來估算視差圖。而基於視窗的立體匹配與 SWP 在兩張

輸入影像的架構下表現相若。那是因為在參考相同的畫素數的前題之下，基於視窗的立體匹配不像 SWP 把考慮的範圍拆解成兩個長度較短的視窗，所以在無紋理區抗誤能力較強；而 SWP 則是在遮蔽區附近占有優勢。然而在四張輸入影像的架構之下，基於視窗的立體匹配因為遮蔽區干擾的問題，使得準確度無法大幅的提升，誤差因而大於 SWP。

由這一小節的實驗數據我們可以發現，四張輸入影像的架構確實可以增進視差估算的準確度，而且搭配 SWP 這種立體匹配方法計算 SSD 能夠更有效利用多餘兩張輸入影像的資訊。**5.2.2 以總體最佳化取代 WTA**

在第四章當中我們採用了兩種以動態規劃加速的總體最佳化演算法：基本的總體最佳化與對稱立體演算法來增進視差圖的準確度。在這一小節當中，我們以這兩種總體最佳化來幫助各種架構做視差值的決策以得到較 WTA 決策更準確的視差值。然而如 5.1.2 所提的，對稱立體演算法無法在兩張輸入影像的架構下估算完整的視差圖，於是本節的實驗的操作變因為兩張輸入影像的基本總體最佳化、四張輸入影像的總體最佳化以及對稱立體演算法。我們控制了圖 5.4 中 matching cost calculation 區塊的變因對此三者進行實驗，計算比較三者的均方誤差。

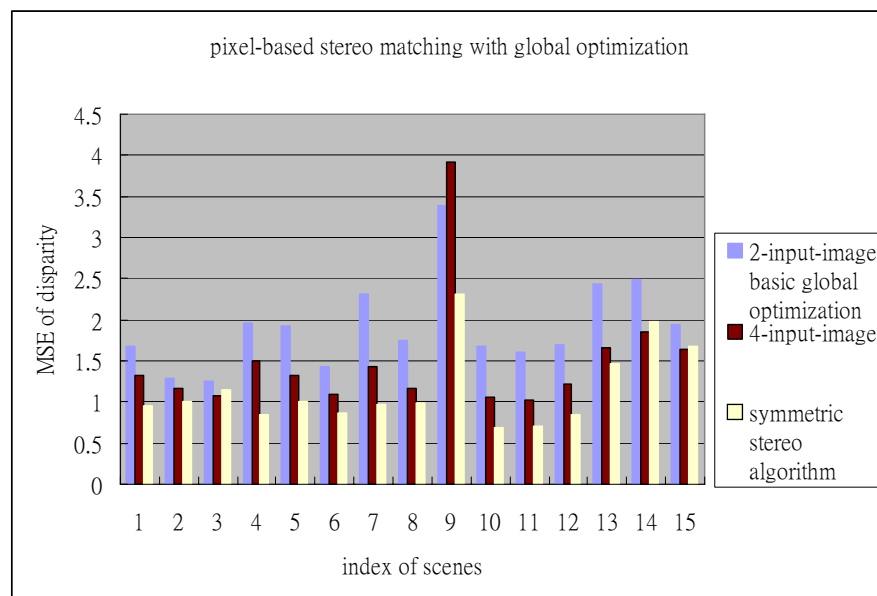


圖 5.8: 基於畫素的立體匹配搭配總體最佳化估算視差圖的均方誤差

圖 5.8 是以基於畫素的立體匹配搭配兩張與四張輸入影像的基本總體最佳化演算法，以及搭配四張輸入影像架構的對稱立體演算法於十五個場景當中估算出視差圖的均方誤差所作成的直方圖。比較圖 5.7 與上一小節的圖 5.5，我們發現基於畫素的立體匹配搭配總體最佳化的決策比起搭配 WTA，誤差都有明顯下降。特別是搭配使用四張輸入影像架構修補遮蔽區域的對稱立體演算法。

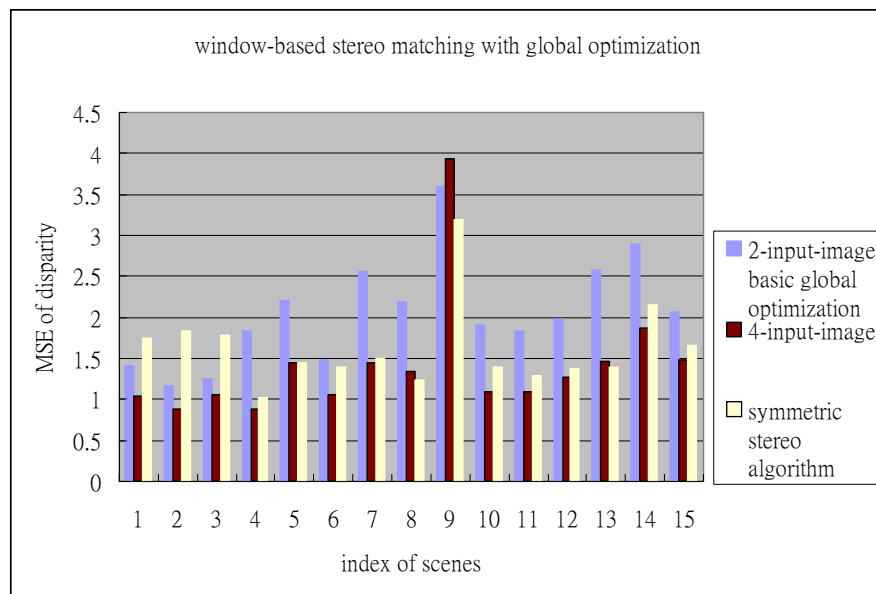


圖 5.9: 基於視窗的立體匹配搭配總體最佳化估算視差圖的均方誤差

圖 5.9 是基於視窗的立體匹配搭配兩張與四張輸入影像的基本總體最佳化演算法，以及搭配四張輸入影像架構的對稱立體演算法於十五個場景當中估算出視差圖的均方誤差所作成的直方圖。比較圖 5.9 與 上一小節的圖 5.6，我們可以發現，基於視窗的立體匹配搭配總體最佳化對誤差的抑制效果不如基於畫素的立體匹配來得明顯。其中最重要的原因是基於視窗的立體匹配在影像的左右兩端因為參考的區域超過了影像的範圍而無法計算 SSD，在程式的實作上我們選擇直接給予這些無法計算 SSD 的情況一個極大的 SSD 值。這導致了基於視窗的立體匹配搭配對稱立體演算法在修補左右兩端的遮蔽區域時無法得到較佳的結果。

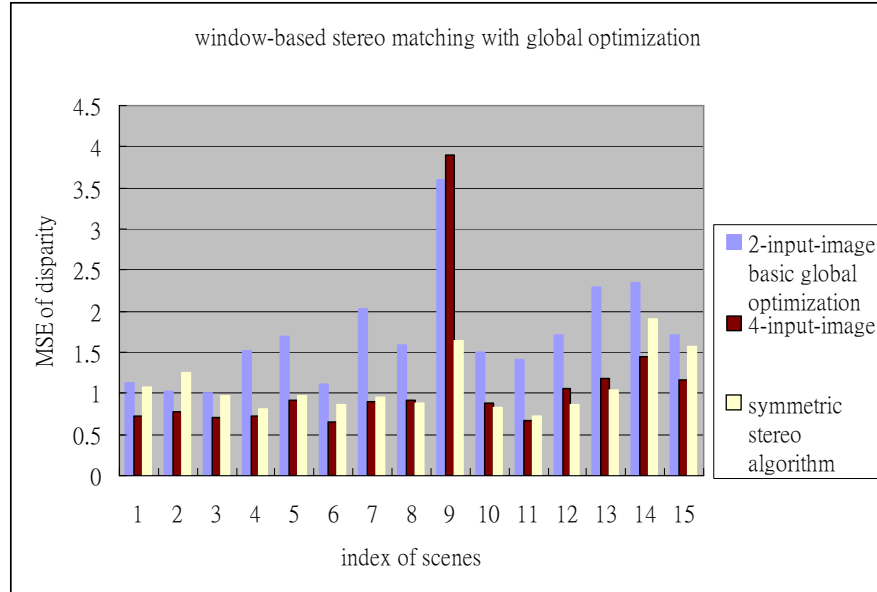


圖 5.10: SWP 配搭總體最佳化估算視差圖的均方誤差

圖 5.10 是 SWP 搭配兩張與四張輸入影像的基本總體最佳化演算法，以及搭配四張輸入影像架構的對稱立體演算法於十五個場景當中估算出視差圖的均方誤差所作成的直方圖。與圖 5.7 比較，我們發現若是改以總體最佳化取代 WTA 做決策搭配 WSP，在視差估測準確度有大幅度的提升。特別是搭配對稱立體演算法。甚至連第九個場景那種擁有大量無紋理區域的輸入影像都能估算出很準確的視差圖。相當的難得，見圖 5.11。

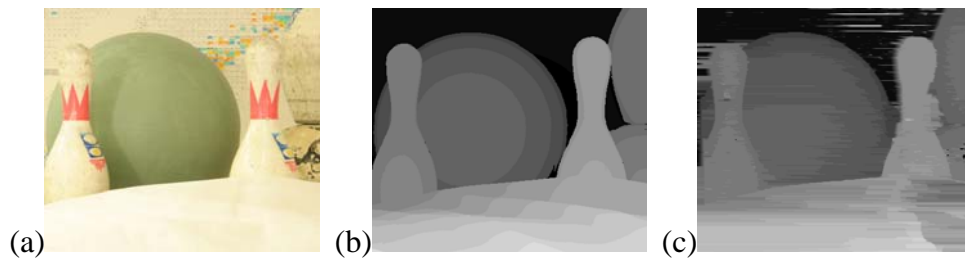


圖 5.11: 場景九: (a) 輸入影像 (b) ground truth (c)以 SWP 求得的視差圖

表 5.2: 三種立體匹配方法搭配總體最佳化在各場景視差估算的均方誤差

	Pixel-based	Window-based	SWP
2-input-image basic global optimization	1.919	2.063	1.710
4-input-image basic global optimization	1.497	1.423	1.107
Symmetric stereo algorithm	1.164	1.631	1.080

由表 5.2 我們可以發現基於視窗的立體匹配是最不適合搭配總體最佳化決策的。同時參考多個畫素的優勢僅止於搭配 WTA 決策的架構之下。在遮蔽區域當中，因為受到半邊干擾資訊的影響而無法以正確的視差值計算出低的 SSD，因而使得動態規劃無法做出正確的決策。反觀基於畫素的立體匹配，雖然很多可能的視差值計算都會得到很低的 SSD，然而動態規劃卻極可能從這些有可能是正確的候選視差值當中挑出正確的那個。因此基於畫素的立體匹配搭配總體最佳化其實是很好的視差估算方法。至於兼具了其它兩者長處的 SWP，在準確度的表現上依然是最好的，而且對於各種不同特性的場景都能維持一定的準確度，不會有偶發的劇烈誤差。

5.3 image-based rendering 的實驗結果

在本節中，我們以三組 Blender 合成的影像 image-based rendering。每一組影像都合成兩張影像，輸入影像與輸出影像的視點位置如圖 5.12 所示，在空間座的 X 軸上 0、1、2、3 的視點合成輸入影像，在 1.4、1.8 的位置合成輸出影像的理論值。最後再將輸出影像與 Blender 合成的理論值進行 PSNR 的計算以評估各種演算法的效能。須注意的是，現行的 FTV 系統都是多台攝影機的架構，因此在本節我們並不以兩張輸入影像的架構進行實驗。

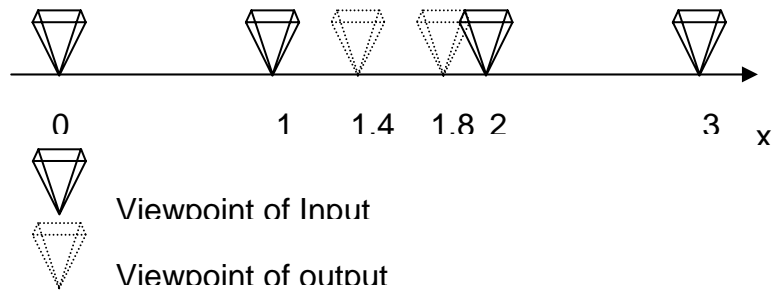
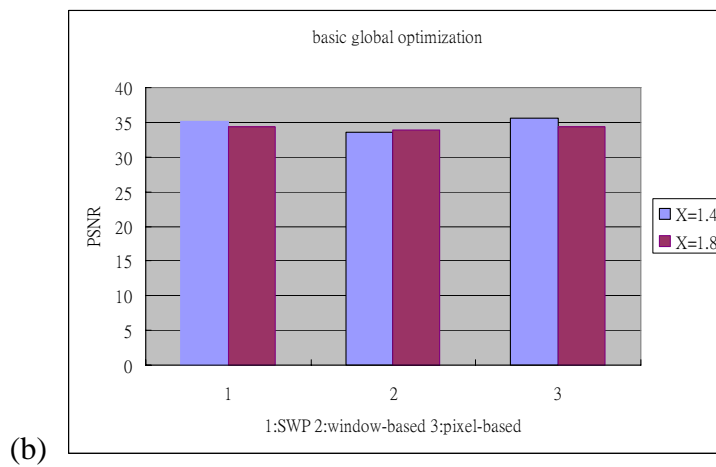
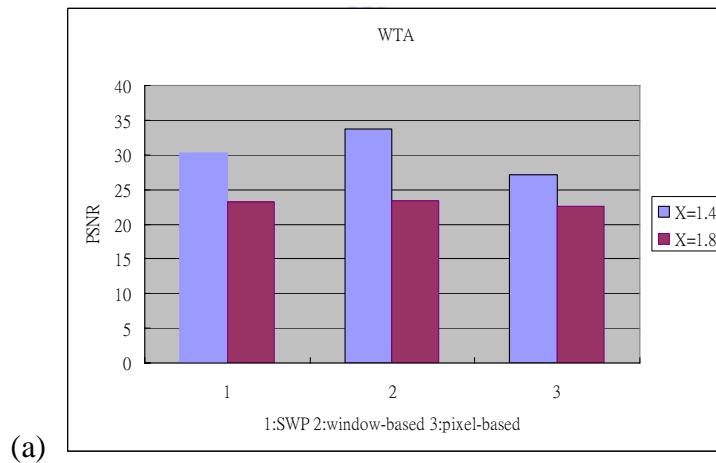


圖 5.12: 輸入與輸出影像的視點

5.3.1 場景一的實驗結果

第一個場景是會議室的場景。和一般簡單的虛擬影像不同，光線、材質等都和真實影像差不多。空間複雜度稍高。而且地板上有兩處鏡面反光的現象容易造成視差判斷的錯誤。以下是我們的實驗數據。



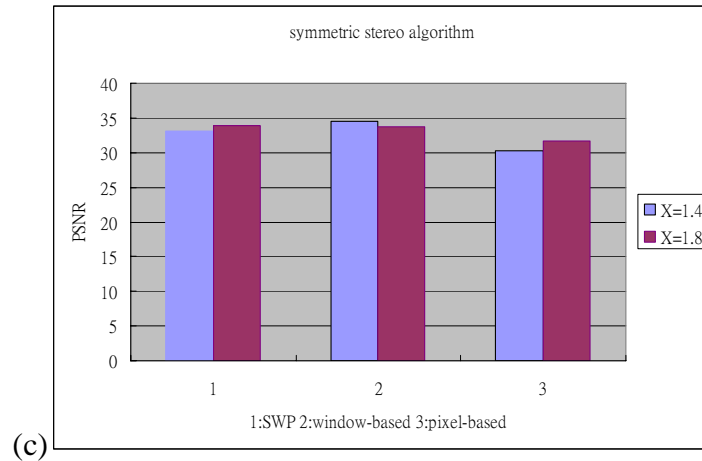


圖 5.13: 場景一，各種決策方式的 PSNR: (a) WTA (b) 基本總體最佳化
(c) 對稱立體演算法

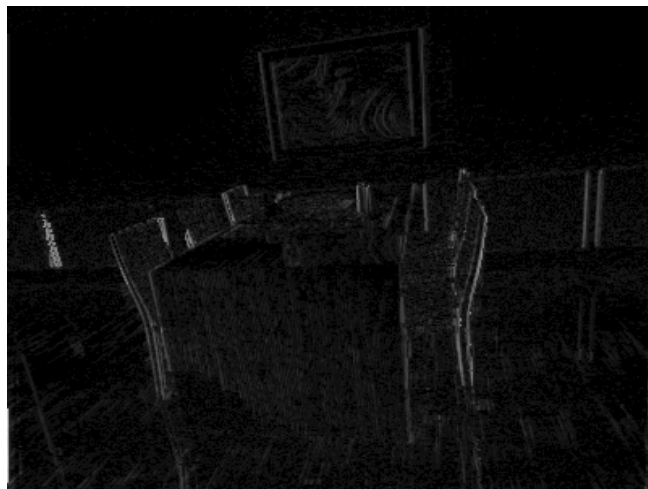
由圖 5.13 可以發現除了以 WTA 進行的決策產生的輸出影像 PSNR 較低之外，不論那種立體匹配方法搭配了兩種以動態規劃加速的總體最佳化決策方式都得到了不錯的影像品質，PSNR 在 33dB 之上。原因是視差值的錯誤只影響輸入影像的畫素最終落在輸出影像上的位置，然而在這種紋理反差不大的場景中並不足以造成輸出影像嚴重的誤差。圖 5.14 是 SWP 搭配動態立體演算法合成的 $X=1.4$ 的輸出影像以及 Blender 合成的理論影像的以及兩者之間的誤差圖。可以發現誤差主要是發生在輪廓的位置。



(a)



(b)



(c)

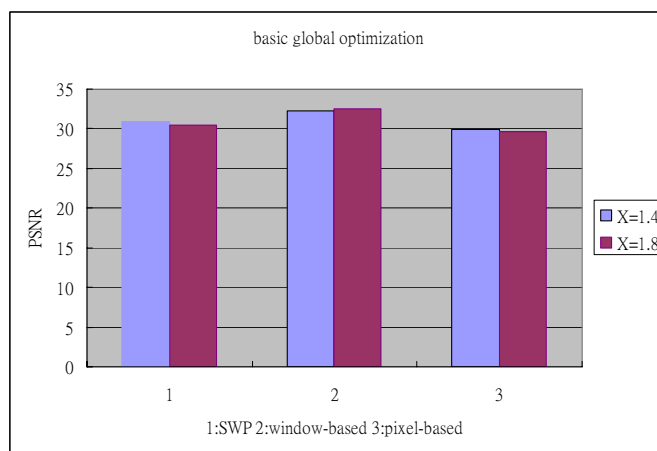
圖 5.14: 場景一的影像合成實例: (a) 理論影像 (b) 合成結果(c) 誤差

5.3.2 場景二的實驗結果

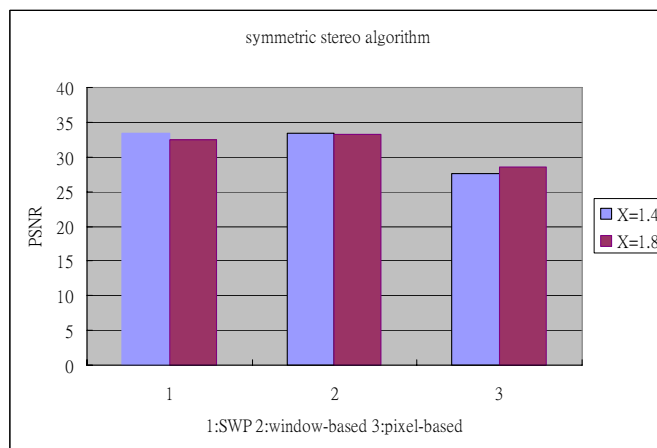
有別於場景一，這個馬雅祭壇的場景立體架構較為簡單也沒有鏡面的行情，然而和一般簡單的虛擬影像類似，場景二產生的影像的紋理反差很大。可以想見視差的誤判將會導致嚴重的錯誤。圖 5.15 是場景二的 PSNR 圖表。



(a)



(b)



(c)

圖 5.15: 場景二中各種決策方式的 PSNR: (a) WTA (b) 基本總體最佳化

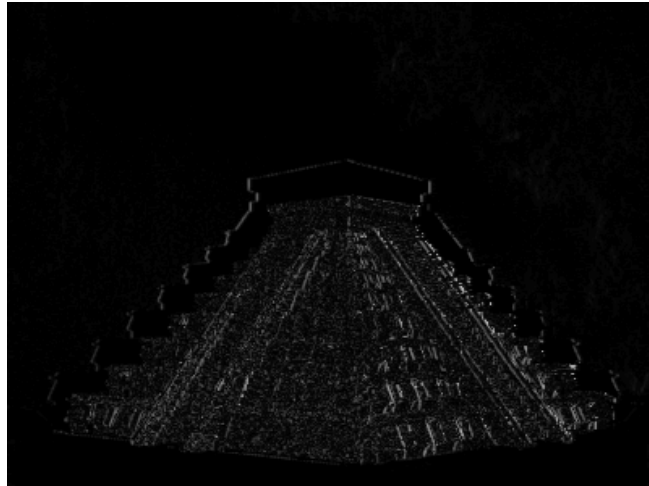
(c)對稱立體演算法

大致而言，場景二當中各種方法的表現與場景一相若，唯有對稱立體演算法

搭配基於畫素的立體匹配所得到的 PSNR 明顯的較場景一為低。圖 5.16 是場景二當中基於畫素的立體匹配搭配對稱立體演算法的輸出景像、輸出影像理論值，誤差圖以及視差圖。



(b)



(c)

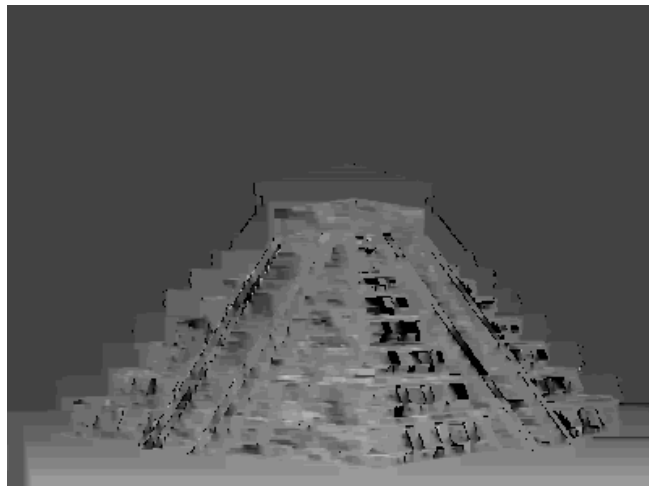


圖 5.16: 場景二中的: (a) 理論輸出影像 (b) 實際輸出影像 (c)誤差 (d)視差圖

圖 5.16 視差圖中的錯誤是導致 PSNR 偏低的主要原因。但是紋理明顯的非遮蔽區域理論上是最容易計算視差的，不應該發生錯誤。後來經過觀察，發現在輸入影像當中，有很多不被遮蔽的畫素並沒有在其它輸入影像當中找到很匹配的畫素。圖 5.19 的白色點是 $X=1$ 的輸入影像當中，最低的 SD 值(以 2.12 式計算)大於一千的畫素，這些白點的分布位置與圖 5.17 中產生明顯誤差的畫素有很高的相關性。因此我們推測輸出影像中祭壇上嚴重的誤差是因為輸入影像在紋理過度複雜而且在具有強烈反差的區域中產了取樣失真，導致視差圖以及輸出影像錯誤。

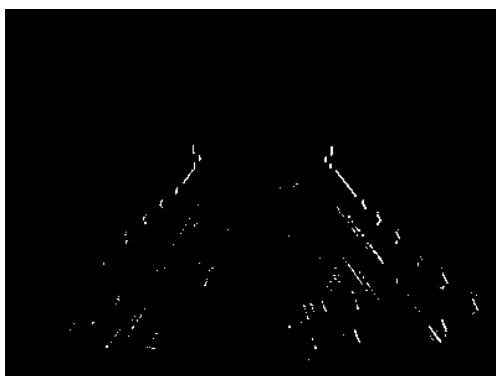
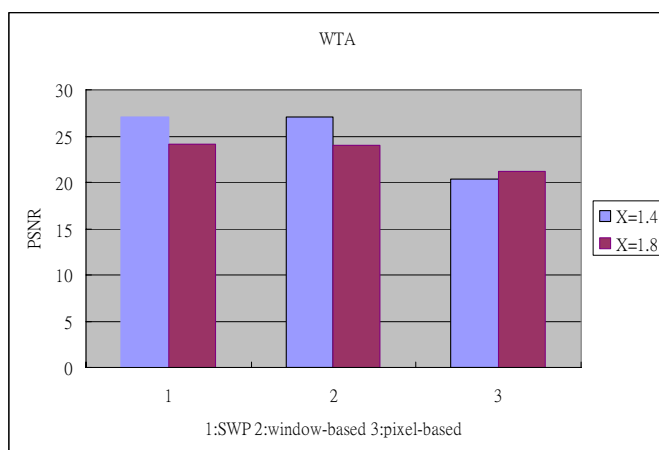


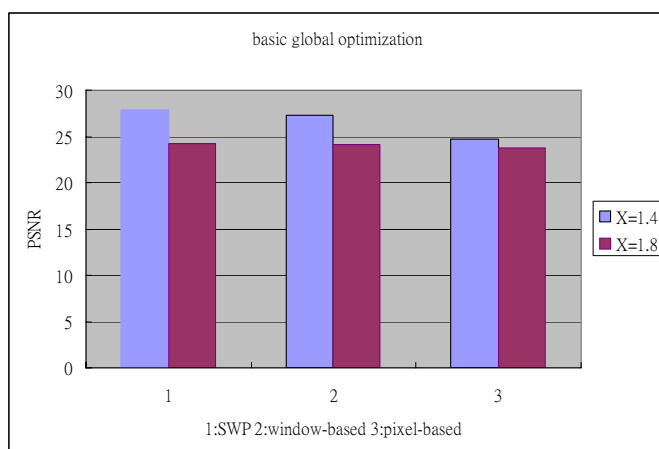
圖 5.17: 場景二當中最低 SD 值大於一千的畫素分布圖

5.3.3 場景三的實驗結果

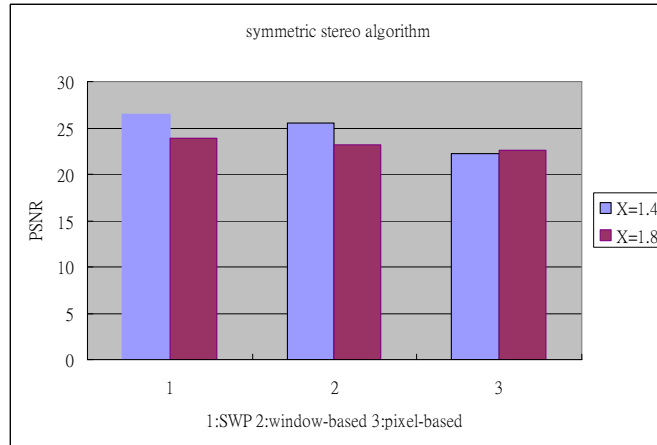
場景三與場景二同樣擁有可能發生取樣失真的區域，而且面積更大反差更強烈，可以預其輸出影像的 PSNR 比前兩者更小的結果。



(a)



(b)



(c)

圖 5.18: 場景三中各種決策方式的 PSNR (a) WTA (b) 基本總體最佳化

(c)對稱立體演算法

由圖 5.18 可以看得出來場景三的輸出影像 PSNR 明顯低於前兩者。而且兩種動態規劃並未能有效的降低誤差，表現只與 WTA 決策相若。圖 5.19 是場景三的實驗當中合成效果最好的 SWP 搭配基本總體最佳化所合成的輸出影像、理論輸出影像、誤差圖以及視差圖。可以發現影像中木紋變化劇烈的區域合成影像以及視差圖的誤差都很大。推測這種發生在空間結構簡單、紋理明顯的區域的視差估算錯誤應該是輸入影像取樣失真所造成。



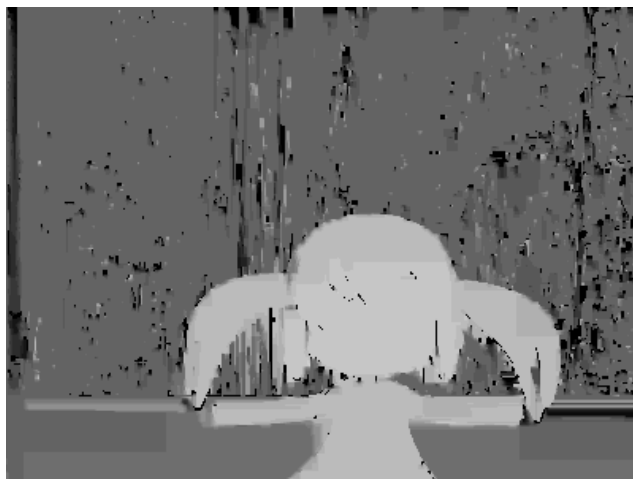
(a)



(b)



(c)



(d)

圖 5.19 場景二中的 (a) 理論輸出影像 (b) 實際輸出影像 (c)誤差 (d)視差圖

由於各場景輸出影像的 PSNR 差異極大，不適合以平均值來總結並衡量各種方法的優劣。因此我們將各場景中不同的立體匹配方法與視差決策方式的 PSNR

都列在表 5.3 中以加以比較。

表 5.3: 三種立體匹配方法搭配三種視差決策方式的 PSNR 列表

		WTA	Basic global optimization	Symmetric stereo algorithm
Scene 1	pixel-based	24.82	34.97	31.02
	window-based	28.53	33.72	33.72
	SWP	26.78	34.71	34.70
Scene 2	Pixel-based	25.56	29.74	28.08
	window-based	31.56	32.55	33.32
	SWP	28.24	30.68	32.95
Scene 3	Pixel-based	20.77	20.77	22.40
	window-based	25.50	25.50	25.73
	SWP	25.61	25.61	26.07

在表 5.3 中我們可以馬上確定的是，三種視差值決策方式當中，WTA 的效果是最差的，三種立體匹配方法中，基於畫素的立體匹配是效果最差的。在三種場景下，搭配三種視差決策方式的九種情況之下 SWP 勝過基於視窗的立體匹配的情形有五種，但差距都在 2dB 內。在三種場景下，搭配三種立體匹配方法的九種情況之下，對稱立體演算法也有五種情況勝過基本的總體最佳化演算法，最大差距是 2.27dB。因此在有限的實驗數據內，我們可以說，以 image-based rendering 的影像品質而言，SWP 比起基於視窗的立體匹配是較好的匹配代價計算方法。而對稱立體演算法比起基本的總體最佳化而言是較好的視差值決策方法。

第六章

結論與未來工作

在這篇論文中我們總共提出了三個方法來改善傳統的立體匹配方法的效能。一是以四張輸入影像架構取代兩張輸入影像架構，二是以單邊視窗對立體匹配方取(SWP)代基於畫素的立體匹配以及基於視窗的立體匹配。三是以可搭配四張輸入影像架構修補遮蔽區域視差值的對稱立體演算法取代 winner-take-all (wta) 視差決策方法。經由實驗的結果我們發現以這三個方法取代傳統方法在視差估算以及 image-based rendering 方面有著下列的改善：

- 比起兩張輸入影像的架構，四張輸入影像的架構搭配基於畫素的立體匹配、基於視窗的立體匹配以及我們提出的 SWP，都夠降低視差估算的誤差。然而搭配基於視窗的立體匹配所得到的改善幅度較小。
- 只有從兩張輸入影像的架構推廣至四張輸入影像的架構，才能支援對稱立體演算法的遮蔽區域視差修補機制。而對稱立體演算法在各種情況、各種目的之下效能都勝過 WTA。
- 比起 WTA 視差決策方法，對稱立體演算法搭配基於畫素的立體匹配、基於視窗的立體匹配以及我們提出的 SWP，在視差估算方面都有獲得改善。平均來說以搭配 SWP 的情形準確度最高，而且對各種場景有穩定的表現。搭配基於畫素的立體匹配則改善幅度很大，準確度稍次於 SWP，穩定性較低。而搭配視窗的立體匹配改善程度很低，準確度最差，遠遠落後前二者。
- 比起 WTA 視差決策方法，對稱立體演算法搭配基於畫素的立體匹配、基於視窗的立體匹配以及我們提出的 SWP，在 image-based rendering 方

面都能夠獲得改善。輸出影像的品質以搭配 SWP 的情況最好，搭配基於視窗的立體匹配稍微次之，差異不明顯，而搭配基於畫素的立體匹配影像品質最差。

- 若單就 image-baser rendering PSNR 為考量，基於視窗的立體匹配搭配對稱立體演算法稍遜於 SWP 搭配對稱立體演算法，差距不明顯，可做為替代方案。但視差估算的均方誤差大約是 SWP 搭配對稱立體演算法的 1.5 倍。
- 若單就視差估算的均方誤差為考量，基於畫素的立體匹配搭配對稱立體演算法稍遜於 SWP 搭配對稱立體演算法，差距不明顯，可做為替代方案。但在 image-based rendering 的 PSNR 方面，SWP 搭配對稱立體演算法在各個場景中都有 3 到 4dB 的優勢。
- 綜合以上兩點，不論是追求 image-based rendering 的高畫質或是視差估算的高準確度高穩定度，對稱立體演算法搭配 SWP 都是最好的選擇。雖然單就 image-based rendering 或是視差估算而言，都有著效果稍差計算複雜度稍低的替代方案，但是都不能同時兼顧 image-based rendering 以及視差估算兩者的效能。

雖然我們提出的演算法比起傳統的方法有所改善，但仍然有努力的空間。以下我們條列了幾項未來工作的重點：

- 在 image-based rendering 的實驗當中我們遇到了因為輸入影像取樣失真造成場景三的輸出影像 PSNR 遠低於其它場景。關於這個問題 G. Van Meerbergen 所提出的，使用動態規劃的階層式對稱立體演算法當中的階層式概念給了我們一些啟發，或許經由各階層模糊化並降低取樣率的影像當中漸近式的估算視差值能做為輸入影像取像失真問題的解決方案。
- 在實驗流程的最後的 render 步驟中，我們只採用了一種 rendering 方式，未來或許可以比較不同的 render 方法，找出效果較佳者。
- 雖然對稱立體演算法在各種情況下的效能都明顯的勝過 WTA，然而在

branch 指令的使用上也比 WTA 多了許多。G. Van Meerbergen 宣稱使用動態規劃的階層式對稱立體演算法當中的階層概念，在處理資料量大的情況下能夠有效的降低時間複雜度。未來我們可以朝這方面研究或是找尋其它時間複雜度與效能都介於對稱立體演算法與 WTA 之間的替代方案。



參考文獻

[1] T. Fujii and M. Tanimoto, Furo-Cho and Chikusa-ku, "Free-Viewpoint TV System Based on Ray-Space Representation", Proceedings of SPIE Vol. 4864(2002) , pp 175-189

[2] Liang-Zhong Fan, Mei Yu, Gang-yi Jiang, Rang-ding Wang and Yong-deak Kim, "New Ray-Space Interpolation Method For Free Viewpoint Video System", Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05) 2005, pp 684-688

[3] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment", IEEE Trans. PAMI, vol.16, no. 9(1989), pp 920-932



[4] G. Van Meerbergen, M. Vergauwen, M. Pollefeys and L. Van Gool, "A Hierarchical Symmetric Stereo Algorithm Using Dynamic Programming", International Journal of Computer Vision 47(2002), pp 275–285

[5] N. Anantrasirichai, C. Nishan Canagarajah , David W. Redmill and David R. Bull, "Dynamic Programming for Multi-View Disparity/Depth Estimation", ICASSP 2006 pp II-269-272

[6] Ruigang Y. and Marc P., "Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware", Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), pp I-211-217

[7] Daniel S. and Richard S., “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms”, International Journal of Computer Vision (2001), pp 7-42

[8] F. Shao., G. Jiang, K. Chen, M. Yu and T. Choi, “Ray-Space Data Compression Based on Prediction Technique”, International Conference on Computer Graphics, Imaging and Visualization (2005), pp 347-350

[9] T. Kobayashi, T. Fujii, T. Kimoto and M. Tanimoto, “Interpolation of Ray-Space Data by Adaptive Filtering” In Three-Dimensional Image Capture and Applications III, Brian D. Corner, Joseph H. Nurre, Editors, Proceedings of SPIE Vol. 3958 (2000) pp 252-259

[10] O. Faugeras, “Three-Dimensional Computer Vision a geometric viewpoint”, 1993

[11] D. Scharstein and C. Pal. “Learning conditional random fields for stereo”, In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007), pp 1-8

[12] D. Scharstein and R. Szeliski. “High-accuracy stereo depth maps using structured light.” In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2003), volume 1, pp 195-202

自傳

許紹唐，男，民國七十二年一月二十九日生於台灣省澎湖縣馬公市。民國九十五年六月畢業於交通大學電子工程學系，並於同年直升交通大學電子研究所碩士班，從事立體影像相關研究，指導教授為杭學鳴博士。民國九十七年七月取得碩士學位，畢業論文題目為『使用多重攝影機影像的立體視差估算』。

e

