

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

使用二維攝影機陣列訊號
合成自由視點視訊



Free Viewpoint Video Synthesis using
2-D Camera Array Signals

研究生：徐崇毓

指導教授：杭學鳴 博士

中華民國九十八年七月



使用二維攝影機陣列訊號
合成自由視點視訊

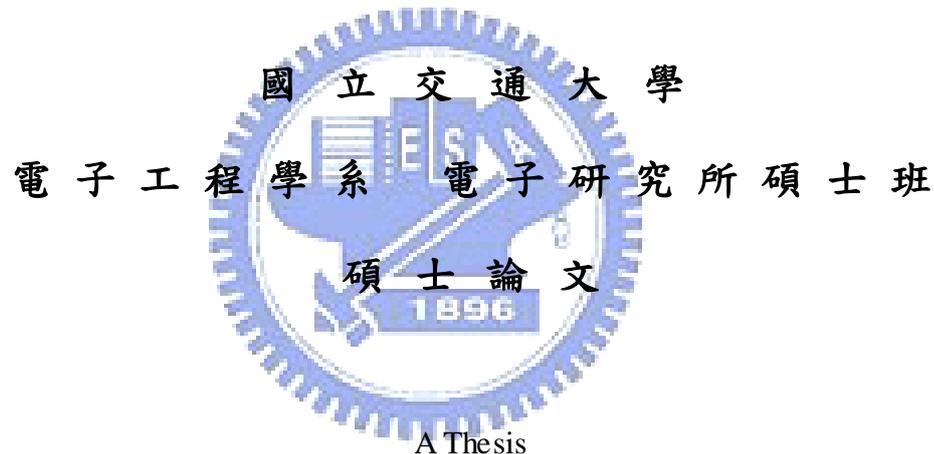
Free Viewpoint Video Synthesis using
2-D Camera Array Signals

研究生：徐崇毓

Student: Chung-Yu Hsu

指導教授：杭學鳴

Advisor: Dr. Hsueh-Ming Hang



Submitted to Department of Electronics Engineering & Institute of Electronics
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
in
Electronics Engineering

July 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年七月



使用二維攝影機陣列訊號 合成自由視點視訊

研究生：徐崇毓

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

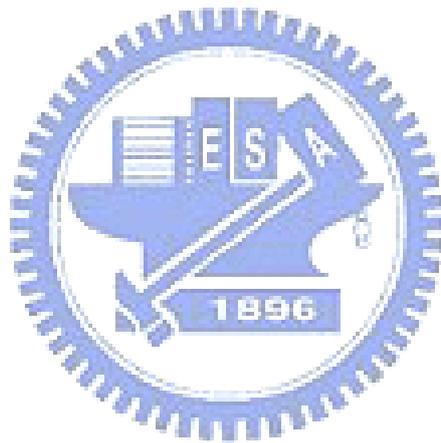


立體影像技術已逐漸成為多媒體領域中的研究主流，而自由視點視訊更是其中的重要議題之一。自由視點的目的是從攝影機陣列所擷取的影像來產生可以任意選擇虛擬視點的影像訊號。在合成自由視點視訊上主要有兩個步驟：**(1)**從攝影機陣列所擷取的影像來重建三維資訊，**(2)**合成使用者視點所看到的虛擬視訊。

現今，大部分的自由視點視訊系統都專注於一維攝影機陣列所記錄的影像資訊，且研究內容也多以影像為主，而非視訊。我們希望將研究成果拓展到二維攝影機陣列與影像序列(視訊)上。由於資料大量增加，除了一般的擷取資訊精確度的問題外，我們也需要同時考慮計算複雜度的問題。

在本論文中，我們提出三個演算法：**(1)**使用二維攝影機陣列進行影像視差估測，**(2)**棋盤式視差估測及影像合成，以及**(3)**使用移動補償技術進行合成自由視點視訊。首先，我們使用四台不同位置攝影機來改善視差估測的精確性。由於計算複雜度的增加，我們只處理在棋盤點上的攝影機所記錄的資訊；換言之，對於棋盤點之間的攝影機視差圖，我們利用視差圖的平移特性以及參考鄰近攝影機影像的視差來進行估測。因此我們省下

了將近一半的計算量，且其結果不亞於每一台攝影機都使用原本估測方法所得到的結果。因爲一連串在時間上接近的影像彼此間具有強大的關聯性，我們引進了 GoP 的概念，對每個 GoP 的第一張畫面優先進行處理。爲了降低計算量，我們接著使用移動向量來估測 GoP 中剩餘畫面的視差圖。進行移動估測時，我們也同時加入了視差資訊來提升移動估測的準確性。經由實驗結果顯示，計算時間可以降爲原本的三分之一，且合成影像的品質僅減少 0.05dB。



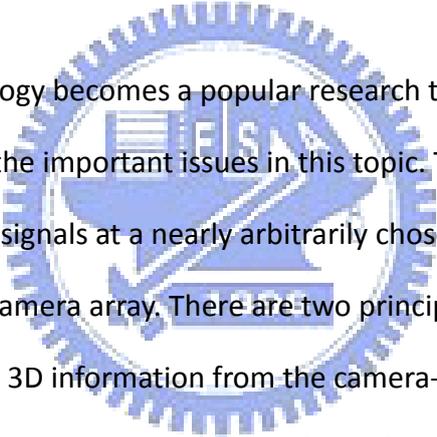
Free Viewpoint Video Synthesis using 2-D Camera Array Signals

Student: Chung-Yu Hsu

Advisor: Dr. Hsueh-Ming Hang

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

Abstract

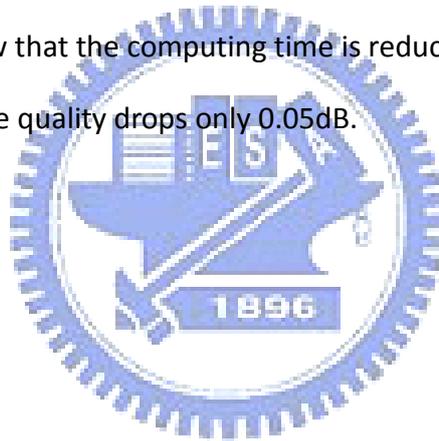


Stereo image technology becomes a popular research topic in multimedia, and the free viewpoint video is one of the important issues in this topic. The purpose of free viewpoint video is to generate video signals at a nearly arbitrarily chosen virtual viewing point based on the captured videos by a camera array. There are two principal steps in free viewpoint video synthesis: (1) estimate the 3D information from the camera-array captured images, and (2) synthesize the virtual video at the user's chosen viewpoint.

Nowadays, most free viewpoint video systems focus on one dimensional camera array in recording the image information. Also, images, rather than videos, are often the subject of study. We like to extend the existing research results to cover the 2D camera array and the image sequences (videos). Because of the large amount of the increased data, in addition to the ordinary accuracy problem in information extraction, we also need to consider the computational complexity issue.

In this thesis, we propose three algorithms: (1) disparity estimation using 2D camera array pictures, (2) checker-board disparity estimation and image synthesis, and (3) free-viewpoint video synthesis using the motion compensation technique. First, we use four

cameras at different locations to improve the accuracy of disparity estimation. Due to the increased computational complexity, we process the data mainly on the cameras at the checker-board locations. That is, we estimate the disparity map of an in-between camera from the maps of its nearby four cameras by using the disparity map shifting property. Thus, we save nearly half of the computations and the results are as good as the original method that processes every camera data. Because the temporally close images in a sequence are strongly correlated, we adopt the group of pictures (GoP) concept that process the first frame of a GOP first. Then, we use the motion vectors to estimate the disparity maps of the remaining frames in a GoP to reduce computation. We improve the motion estimation accuracy by including the disparity information in the motion estimation index. The experimental results show that the computing time is reduced to one third of original time, and the synthesized image quality drops only 0.05dB.



誌謝

在交大這兩年的研究生涯中，首先要感謝我的指導教授杭學鳴老師。杭老師除了在學術研究上給予我指導外，更教導我做學問的方法以及應有的態度，讓我碩士生活中充滿了收穫。在此向老師致上最高的感謝之意。

這篇論文能夠完成，也要感謝蔡家揚學長，除了在實驗的設計以及分析上提供了寶貴的建議，更不厭其煩的與我討論，讓我得到不少研究上寶貴的經驗。此外，也要感謝盧世榮學弟常常幫我買牛奶，並提供了我生活上諸多幫助，讓我可以更致力於研究。

另外，也要感謝 Commlab 提供的豐富資源，讓我得以順利進行研究並完成學業。實驗室的學長及同學們就好像家人一般，大家平時的互動都讓我感到格外親切，讓我這兩年留下難忘的回憶。

最後我要感謝我的家人，由於你們的支持，我曾能無後顧之憂地從事研究。在此，僅將論文獻給曾經幫助過我、支持我的師長、友人、以及家人。





目錄

摘要.....	I
ABSTRACT	III
誌謝.....	V
目錄.....	VII
圖目錄.....	IX
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機及目的.....	2
1.3 論文大綱.....	3
第二章 極平面圖像概述.....	5
2.1 立體影像系統.....	5
2.2 極平面圖像.....	6
2.3 極平面圖像和三維空間的關係.....	11
2.4 不連續極平面圖像.....	15
第三章 使用動態規劃進行視差估測.....	17
3.1 視差定義.....	17
3.2 使用動態規劃進行視差估測.....	19
3.2.1 二維搜尋平面.....	20
3.2.2 樹狀搜尋演算法.....	22
3.2.3 代價函數.....	24
3.2.4 立體樹.....	29
第四章 適用於二維攝影機陣列之影像合成方法.....	33
4.1 傳統視差估測的問題.....	33
4.2 參考鄰近影像進行視差估測.....	35
4.3 棋盤式視差估測.....	44
4.4 影像合成.....	53
4.4.1 一維攝影機陣列影像合成.....	53
4.4.2 二維攝影機陣列影像合成.....	60
4.5 實驗結果.....	62
4.5.1 理想影像.....	64
4.5.2 加入AWGN的影像.....	67

4.5.3 實驗討論.....	71
第五章 適用於二維攝影機陣列之視訊合成方法.....	75
5.1 使用移動向量進行視差估測.....	75
5.2 參考視差資訊進行移動估測.....	79
5.3 實驗結果.....	86
5.3.1 理想連續影像.....	87
5.3.2 加入AWGN的連續影像.....	95
5.3.3 實驗討論.....	102
第六章 結論與未來工作.....	107
6.1 結論.....	107
6.2 未來工作.....	108
參考文獻.....	109



圖目錄

圖 2-1 典型的立體影像系統.....	5
圖 2-2 以基線維軸心的平面束.....	6
圖 2-3 在同一直線上的攝影機成像.....	7
圖 2-4 影像資料立方體.....	7
圖 2-5 鏡頭方向和基線垂直時，極像相當於影像上的一條水平掃描線.....	8
圖 2-6 在圖 2-4 中的資料立方體的水平橫切面.....	8
圖 2-7 (A)相機鏡頭方向和基線垂直的設置結構 (B)此種鏡頭設置下所得到的極平面影像.....	9
圖 2-8 (A)相機鏡頭方向和基線之間角度固定的設置結構 (B)此種鏡頭設置下所得到的極平面圖像.....	10
圖 2-9 (A)相機鏡頭方向和基線之間角度不固定的設置結構 (B)此種鏡頭設置下所得到的極平面圖像.....	10
圖 2-10 攝影機和三張不同距離的平面在空間中的關係.....	12
圖 2-11 在圖 2-10 中所得到的極平面圖像.....	12
圖 2-12 真實影像所得到的極平面圖像和攝影機擷取影像的關係.....	13
圖 2-13 位於 $Z = 0$ 上的攝影機陣列和空間中一點 (X, Y, Z) 的位置關係.....	14
圖 2-14 極平面圖像上各種不同直線的示意圖.....	14
圖 2-15 極平面圖像的不連續情形.....	15
圖 3-1 典型多重視點系統.....	18
圖 3-2 雙重視點系統.....	19
圖 3-3 一對針孔攝影機對同一場景所取得的掃描線.....	20
圖 3-5 連結路徑和節點間的關係.....	21
圖 3-6 一個符合樹狀搜尋演算法條件的匹配序列.....	23
圖 3-7 PIXEL-BASED DISSIMILARITY FUNCTION.....	25
圖 3-8 WINDOW-BASED DISSIMILARITY FUNCTION.....	26
圖 3-9 (A)正確的匹配 (X_L, X_R) 所在位置及其視窗範圍 (B)錯誤的匹配 (X_L', X_R') 所在位置及其視窗範圍.....	27
圖 3-10 SINGLE-SIDE WINDOW-BASED DISSIMILARLY FUNCTION 在匹配 (X_L, X_R) 的示意圖.....	28
圖 3-11 以圖 3.6 為範例的立體樹，其中打叉的節點為超出預設視差值範圍的匹配.....	30
圖 3-12 用來實現樹狀搜演算法的三個矩陣。其中矩陣 Γ 和矩陣 Π 為剛開始處理的狀態.....	31

圖 3-13 其中一種走法在矩陣 Γ 與矩陣 Π 上(A)走一步的結果 (B)走兩步的結果	32
圖 4-1 因遮蔽效應造成的OCCLUSION	34
圖 4-2 因為兩物體彼此位置過於接近，導致兩者之間的區域視差值錯誤	34
圖 4-4 給定的攝影機 O_c 和鄰近攝影機之間的位置關係.....	35
圖 4-5 攝影機 O_c 和鄰近攝影機的擷取影像.....	36
圖 4-6 參考不同位置的攝影機所得到的視差圖	37
圖 4-7(A)參考右邊影像時所選取的掃描線及搜尋方向 (B) 參考下方影像時所選取的掃描線及搜尋方向	38
圖 4-8 視窗和掃描線方向無關的情形，點 P 為圖中紅色的區塊，方形紅框為視窗，紫色虛線標記為掃描線位置：(A)掃描線方向為水平 (B)掃描線方向為垂直	39
圖 4-9 視窗和掃描線方向相同的情形，點 P 為圖中紅色的區塊，矩形紅框為視窗，紫色虛線標記為掃描線位置：(A)掃描線方向為水平 (B)掃描線方向為垂直	40
圖 4-10 視窗和掃描線方向垂直的情形，紅點表示 P 所在的位置，方形紅框為視窗，紫色虛線標記為掃描線位置：(A)掃描線方向為水平 (B)掃描線方向為垂直.....	40
圖 4-11 三種視窗對圖 4-6 進行投票的結果：(A)視窗和掃描線無關 (B)視窗和掃描線方向相同 (C)視窗和掃描線方向垂直.....	41
圖 4-12 圖 4-13 之測試影像	42
圖 4-13 三種視窗對圖 4-12 進行投票的結果：(A)視窗和掃描線無關 (B)視窗和掃描線方向相同 (C)視窗和掃描線方向垂直.....	42
圖 4-14 不同視窗大小對投票結果的影響情形(A)視窗長度為 5 個PIXEL (B)視窗長度為 10 個PIXEL (C)視窗長度為 15 個PIXEL (D)視窗長度為 20 個PIXEL	43
圖 4-15 視差圖之間的平移關係.....	45
圖 4-16 因相鄰相機距離太遠導致同一物體在兩個成像平面上外觀不同	46
圖 4-17 兩種以上的視差值對應到同一點時的處理情形	46
圖 4-18 以左邊影像的視差圖來推測右邊影像的視差圖.....	47
圖 4-19 因不同視差值移動距離不同而導致右邊影像上部份區域無法定義視差	47
圖 4-20 使用主視差圖合成副視差圖的流程圖	48
圖 4-21 鄰攝影機 O_{nc} 和周圍主攝影機的位置關係	49
圖 4-22 攝影機 O_c 和鄰近攝影機的擷取影像.....	49

圖 4-23 各主攝影機擷取影像的視差圖	50
圖 4-24 使用不同位置的主視差圖推測影像 I_c 的視差結果	51
圖 4-25 以圖 4-24 進行投票來判斷影像 I_c 的視差圖	52
圖 4-26 主攝影機和副攝影機在二維攝影機陣列上的位置關係.....	53
圖 4-27 (A)左邊影像視差值大於對應的右邊影像視差值 (B)視差值修正後的結果.....	54
圖 4-28 (A)對應位置的視差值相同的情形 (B)虛擬視點平面的合成	55
圖 4-29 步驟 2 合成後結果	56
圖 4-30 (A)左邊影像視差值小於對應到的右邊影像視差值	57
(B) 實際的對應情形	57
圖 4-31 (A)虛擬視點位置靠近左邊攝影機時的合成情形	58
虛擬視點位置靠近右邊攝影機時的合成情形	58
圖 4-33 右邊影像在左邊影像上被遮蔽的情形.....	59
圖 4-34 步驟 4 完成後的結果.....	59
圖 4-35 以線性內插的方式將虛擬視點成像平面上無值區域填補後的結果.....	60
圖 4-36 在二維攝影機陣列上的虛擬視點影像合成流程圖	60
圖 4-37 兩階段的虛擬影像合成過程	61
圖 4-38 在二維攝影機陣列系統上合成虛擬視點影像之流程	62
圖 4-39 各種不同的物體運動行為連續影像：	63
圖 4-40 兩種視差估測方式在不同測試影像上的平均PSNR	64
圖 4-41 兩種視差估測方式在測試影像MONKEY-TRANSLATION上PSNR.....	65
圖 4-42 兩種視差估測方式在測試影像MONKEY-ROTATION上PSNR.....	65
圖 4-43 兩種視差估測方式在測試影像MONKEY-LEAVING上PSNR	66
圖 4-44 兩種視差估測方式在測試影像MONKEY-APPROACHING上PSNR	66
圖 4-45 兩種視差估測方式在測試影像MONKEY-COVERING上PSNR	67
圖 4-46 兩種視差估測方式在添加AWGN的測試影像上的平均PSNR	68
圖 4-47 兩種視差估測方式在測試影像MONKEY-TRANSLATION添加了AWGN後的.....	68
圖 4-48 兩種視差估測方式在測試影像MONKEY-ROTATION添加了AWGN後的PSNR	69
圖 4-49 兩種視差估測方式在測試影像MONKEY-LEAVING添加了AWGN後的PSNR.....	69

圖 4-50 兩種視差估測方式在測試影像MONKEY-CLOSING添加了AWGN後的PSNR	70
圖 4-51 兩種視差估測方式在測試影像MONKEY-COVERING添加了AWGN後的PSNR.....	70
圖 4-52 (A) D_{map} (B) $D_{MapEmbed}$ (C) $I_{Virtual}$ (D) $I_{VirtualEmbed}$	71
圖 4-53 (A) $I_{Virtual}$ 和實際結果的誤差 (B) $I_{VirtualEmbed}$ 和實際結果的誤差	72
(C) D_{map} 和 $D_{MapEmbed}$ 之間的差異.....	72
圖 4-54 使用完整視差估測相當於參考四張影像資訊.....	73
圖 4-55 使用完整視差估測相當於參考十六張影像資訊.....	74
圖 5-1 MARCOBLOCK和動態向量之間的關係	76
圖 5-2 使用移動向及線性內插的方式合成視差圖	76
圖 5-3 (A)原始影像 (B)使用完整視差估測方式得到視差圖 (C)利用移動向量合成的視差圖 (D)使用周圍視差值填後補結果 (E)最後合成之視差圖 (F)兩種方式所得的視差圖之間的差異.....	78
圖 5-4 (A)參考影像間略過七張FRAME所合成的視差圖 (B)和使用完整視差估測的視差圖之間的差異	78
圖 5-5 (A)視差圖上MARCOBLOCK對應到正確位置的情形 (B)視差圖上MARCOBLOCK對應到不正確位置的情形.....	79
圖 5-6 加入視差資訊後合成的視差圖和使用完整視差估測的視差圖之間的差異	80
圖 5-7 物體平移和係數 α 之間的關係	81
圖 5-8 物體旋轉和係數 α 之間的關係	81
圖 5-9 物體遠離和係數 α 之間的關係	82
圖 5-10 物體靠近和係數 α 之間的關係.....	82
圖 5-11 物體彼此覆蓋和係數 α 之間的關係.....	83
圖 5-12 參考影像之間略過一張FRAME和係數 α 的關係.....	84
圖 5-13 參考影像之間略過三張FRAME和係數 α 的關係.....	84
圖 5-14 參考影像之間略過七張FRAME和係數 α 的關係.....	85
圖 5-15 利用移動向量來加速視差估測的流程	85
圖 5-16 不同的物體運動行為連續影像:	86
(A)EARTH-TRANSLATION(物體平移) (B)EARTH -ROTATION(物體旋轉).....	86
(C)EARTH -LEAVING(物體遠離) (D)EARTH -APPROACHING(物體靠近)	86

(E)EARTH -COVERING(兩物體互相遮蔽)	86
圖 5-19 在圖 4-39 中各種運動行為的合成影像品質和略過的FRAME數量之關係	89
圖 5-20 在圖 5-16 中各種運動行為的合成影像品質和略過的FRAME數量之關係	89
圖 5-21 連續影像MONKEY-TRANSLATION在略過不同數量FRAME的情況下的各張合成影像品質	90
圖 5-22 連續影像MONKEY-ROTATION在略過不同數量FRAME的情況下的各張合成影像品質	91
圖 5-23 連續影像MONKEY-LEAVING在略過不同數量FRAME的情況下的各張合成影像品質	91
圖 5-24 連續影像MONKEY-APPROACHING在略過不同數量FRAME的情況下的各張合成影像品質	92
圖 5-25 連續影像MONKEY-COVERING在略過不同數量FRAME的情況下的各張合成影像品質	92
圖 5-26 連續影像EARTH-TRANSLATION在略過不同數量FRAME的情況下的各張合成影像品質	93
圖 5-27 連續影像EARTH-ROTATION在略過不同數量FRAME的情況下的各張合成影像品質	93
圖 5-28 連續影像EARTH-LEAVING在略過不同數量FRAME的情況下的各張合成影像品質	94
圖 5-29 連續影像EARTH-APPROACHING在略過不同數量FRAME的情況下的各張合成影像品質	94
圖 5-30 在圖 4-39 添加了AWGN後，各種運動行為的合成影像品質和略過的FRAME數量之關係	95
圖 5-31 在圖 5-16 添加了AWGN後，各種運動行為的合成影像品質和略過的FRAME數量之關係	96
圖 5-32 連續影像MONKEY-TRANSLATION加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	97
圖 5-33 連續影像MONKEY-ROTATION加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	97
圖 5-34 連續影像MONKEY-LEAVING加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	98
圖 5-35 連續影像MONKEY-APPROACHING加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	98
圖 5-36 連續影像MONKEY-COVERING加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	99
圖 5-37 連續影像EARTH-TRANSLATION加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	99
圖 5-38 連續影像EARTH-ROTATION加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	100
圖 5-39 連續影像EARTH-LEAVING加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質	100

圖 5-40 連續影像EARTH-APPROACHING加入AWGN後，在略過不同數量FRAME的情況下的各張合成影像品質.....	101
圖 5-41 (A) D_{map0} (B) D_{map3} (C)使用 D_{map0} 合成的影像 (D)使用 D_{map3} 合成的影像	103
(E)使用 D_{map0} 合成的影像和實際影像的誤差 (F)使用 D_{map3} 合成的影像和實際影像的誤差.....	103
圖 5-42 使用完整視差估測方式得到的兩張相鄰影像視差	104
圖 5-43 利用圖 5-42 所合成出來的影像	104
圖 5-44 參考移動向量資訊得到的兩張相鄰影像視差.....	105
圖 5-45 利用圖 5-44 所合成出來的影像	105



第一章 緒論

1.1 研究背景

近年來由於硬體以及相關演算法的進步，單一視點視訊技術已達成熟；而隨著三維顯示技術的發展，多重視點視訊已逐漸成爲多媒體領域下一個階段的研究重心。MPEG 在 2003 年 11 月的會議中，正式制定了 3DAV(3D Audio-Video)文件 [1][2]，其主要目標是擴展現有的視訊標準，其內容包含了三項多重視點視訊的主要應用：自由視點視訊(Free Viewpoint Video)、全景視訊(Omni-directional Video)、與互動式立體視訊(Interactive Stereo Video)，其中的自由視點視訊即爲本論文主要研究主題。

自由視點視訊的目是使用有限的影像擷取設備來精確描述三維空間中的場景資訊，並能有效地壓縮所紀錄的資訊。當呈現時，能夠即時合成出任意視點所見的畫面。自由視點視訊在娛樂、教育、觀光等領域中有著極大的應用空間，而目前自由視訊最主要的應用爲 Free Viewpoint Television(FTV)[3][4]。

一個典型的自由視點視訊系統包含了五種程序：影像擷取、場景成像(rendering)、壓縮、資料傳輸、以及顯示。其中場景成像爲最複雜的程序，亦是整個系統的核心。場景成像主要分爲兩類：

(1) model-based rendering(MBR)

MBR 主要是利用物體模型投影到成像平面上而成。理想的 MBR 可以達到完美的自由視點視訊，但是建構精細的物體模型本身就有其難度，因此成像時容易產生以下問題：

- 模型中各多邊形的紋理不易分析。
- 多邊形數量不足時，成像結果會有瑕疵。
- 精細的模型(即多邊形數量極多)不適合用於即時運算。

(2) image-based rendering(IBR)[5]

IBR 是利用台不同位置或角度的攝影機所擷取的影像，依據光學原理內插出不同視點所見的畫面。相對於分析及計算物體模型，其所需時間大幅降低。相對的，IBR 則面臨了如何內插的問題。

由以上的比較可以得知 IBR 為實現自由視點視訊較為可行的辦法，因此現有的自由視點視訊系統中，大部分採用 IBR 的方式來進行場景成像。

1.2 研究動機及目的

傳統的自由視點視訊系統大多使用一維攝影機陣列來擷取影像[6]，這使得視點位置僅能在一個平面上移動。為了將移動的範圍延伸到立體空間上，我們將一維攝影機陣列拓展為二維攝影機陣列。

自由視點視訊系統在進行影像內插時，為了增加其準確度會搭配視差(disparity)資訊作為參考。但是視差資訊的計算具有高度複雜度，並且結果容易受到遮蔽效應的影響。為了解決以上兩個問題，在本論文中，我們利用二維攝影機陣列中每台攝影機在空間上的相關性，以及視差資料的特性來減少整體系統計算影像視差的次數，並設計一套投票機制來取得更精確的視差資訊。

另外，現有的自由視點視訊系統僅能處理同一時間點上的影像，所以不同時間點的影像都必須重新計算一次視差資訊。這會使得整個系統無法滿足即時運算的要求。基於連續影像在時間上的相關性，我們利用移動向量來減少視差估測的時間。為了取得更為精確的移動向量，在進行移動偵測時，我們不僅比較原始影像之間的差異，同時也會加入已知的視差資訊作為參考。

1.3 論文大綱

在本論文中，第二章將介紹核平面圖像和三維空間中的場景彼此之間的關係，並利用該關係來合成虛擬視點影像。在第三章中，我們將說明本論文所採用的視差估測方法。第四章將介紹如何利用二維攝影機陣列的空間相關性來提昇視差估測結果和降低整體系統在視差估測上所花費的時間，以及如何使用視差資訊來合成不同視點所看見的影像。該章節最後包含為實驗數據與結果比較。在第五章中，我們將介紹如何利用移動向量減少連續影像在計算視差估測上所花得時間，以及如何使用已知的視差資訊來提昇移動向量的精確度。該章節最後為不同參數設定的實驗結果比較。第六章為結論以及未來工作。





第二章 極平面圖像概述

2.1 立體影像系統

圖 2-1 為一個三維空間中典型的立體影像系統配置。我們使用針孔攝影機來擷取影像資料，並將成像平面置於鏡頭前方。兩台攝影機之間的基線(basic line)和成像平面的交錯處稱之為極點(epipole)，而空間中的任一點 P 和兩台攝影機的基線所組成的平面稱為極平面(epipolar plane)。所有的極平面集合相當於一個和兩台攝影機之間的基線連結的平面束(pencil of planes) [7][8]，如圖 2-2。極平面在成像平面上的相交線稱為極線(epipolar line)，在極平面上的每一點都會投影到與其對應的極線上的點。因此，對應於同一張極平面的兩條極線，其線上各點也會存在著對應關係。這項特性在立體影像處理上非常重要。當我們在兩張影像上搜尋互相對應的點時，藉由此特性能將原本的搜尋範圍從在平面上的搜尋轉為在極線上的搜尋，大量減少了搜尋時間。這項幾何特性被稱為 epipolar constraint[7]。

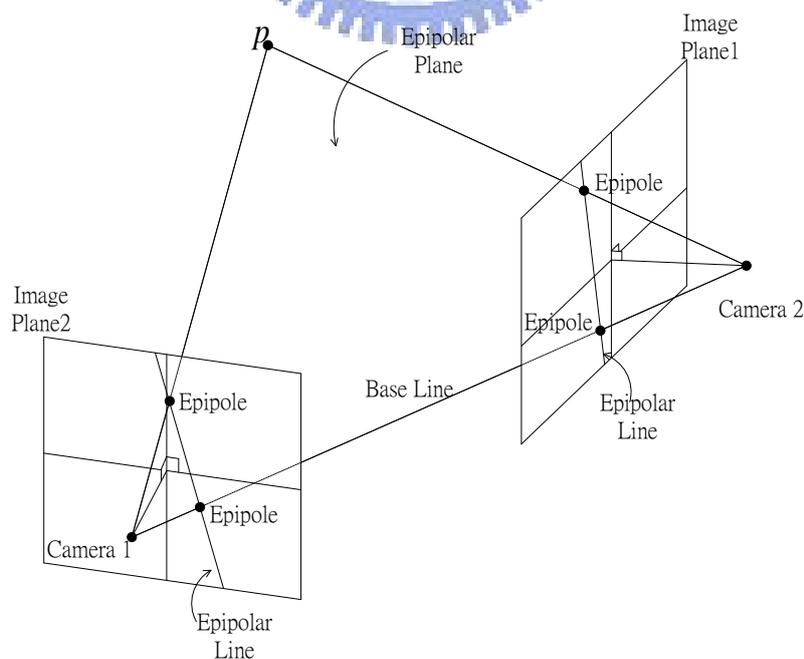


圖 2-1 典型的立體影像系統

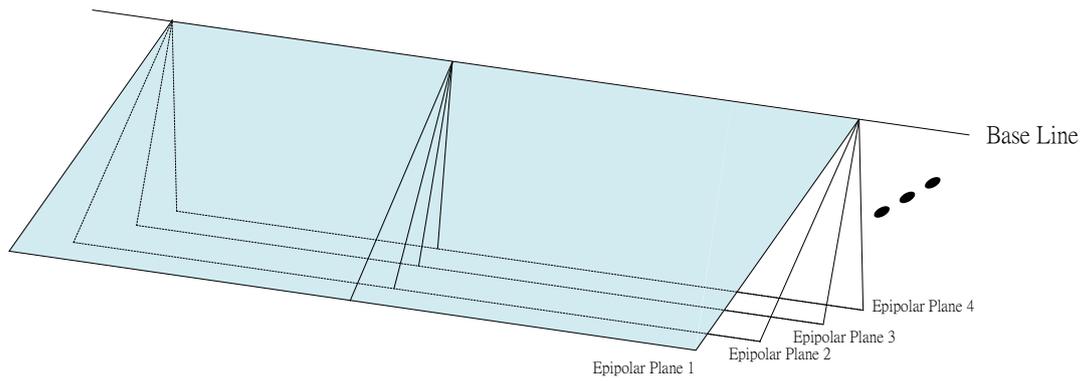


圖 2-2 以基線維軸心的平面束

由於 **epipolar constraint** 能夠降低了在兩台攝影機成像上搜尋對應點的時間，爲了將此概念延伸至攝影機陣列上，我們必須限制攝影機陣列的擺設方式：全部攝影機的位置需在同一條直線上。在此情況下，每一台攝影機之間的基線都會和該直線重合，因此任一對攝影機對於空間中一點所對應的極平面都會相同。所以在兩張以上的影像之間搜尋一給定的對應點時，我們只需要計算一次極平面即可。



2.2 極平面圖像

由於在極平面上的各點都會投影成每張影像上的一條極線，所以一個極平面上的影像資訊都被包含在所對應極線的集合中。爲了方便處理這些資訊，我們將極線集合以圖像方式表現。由於該圖像包含了一個極平面上的所有特徵，所以我們稱之爲極平面圖像(epipolar plane image) [7][8]。

圖 2-3 爲同一直線上的三台攝影機的擷取影像。當攝影機彼此的距離夠小且數量足夠時，每張影像之間會存在空間上的連續性。藉由空間上的連續性，我們可以将攝影機陣列所擷取到的全部影像以一個影像資料所組成的立方體表現，如圖 2-4，其中 v 軸和 u 軸分別爲擷取影像上的座標，而 P 軸表示各台攝影機在直

線上的位置。

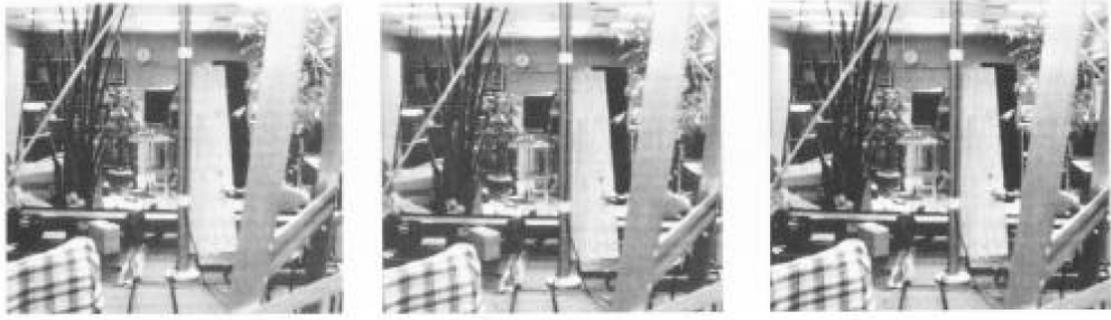


圖 2-3 在同一直線上的攝影機成像 [7]

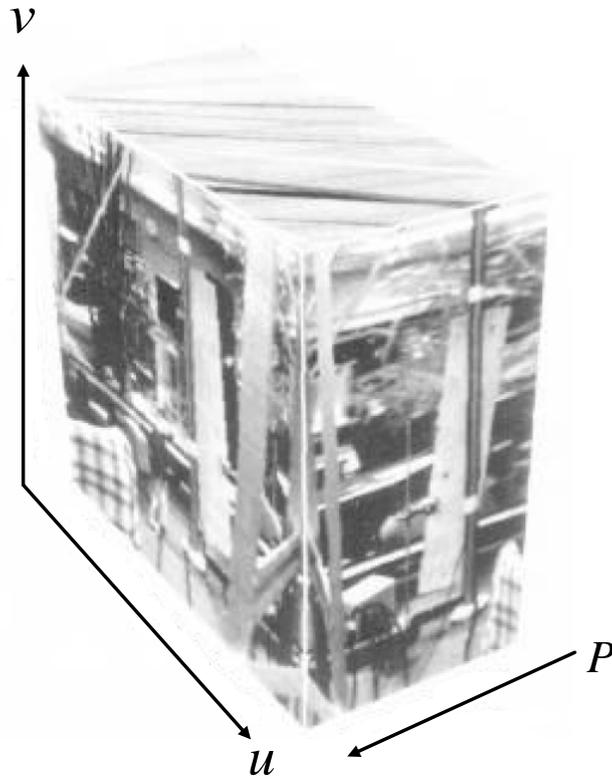


圖 2-4 影像資料立方體 [7]

一張極平面圖像相當於是資料立方體中的一個切面[7][8]，切面的位置和外觀會受到攝影機鏡頭的設置所影響。當全部攝影機鏡頭方向和其基線垂直時，極平面會經過每張攝影機成像上同樣高度的水平掃描線，如圖 2-5。因此，每張影像上的水平掃描線都相當於一條極線。在此種鏡頭設置的情況下，資料立方體的

每一個水平切面都會是一張極平面圖像。圖 2-6 為圖 2-4 的資料立方體中的一個水平橫切面。

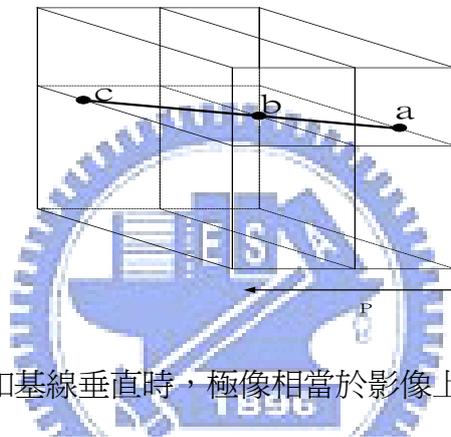
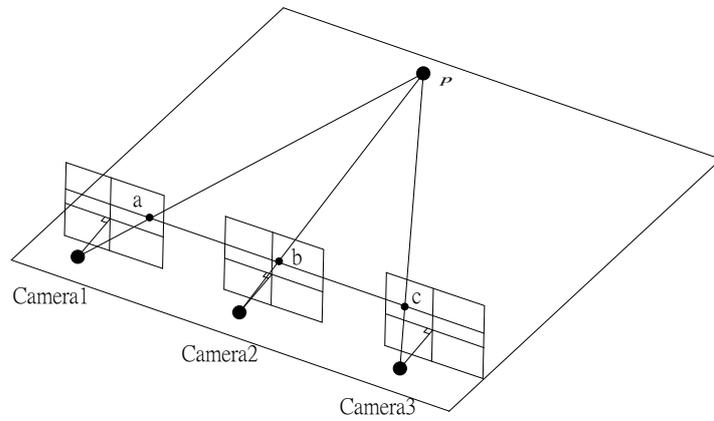


圖 2-5 鏡頭方向和基線垂直時，極像相當於影像上的一條水平掃描線

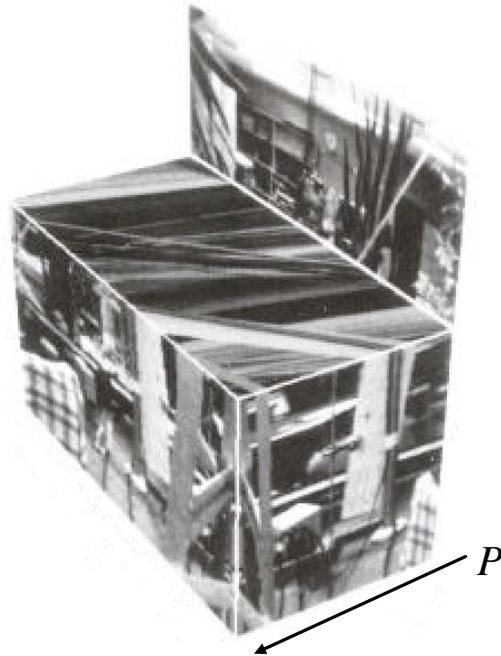


圖 2-6 在圖 2-4 中的資料立方體的水平橫切面 [7]

由以上討論我們可以知道在同一條直線上的攝影機隨著鏡頭方向、聚焦的不同，相同位置的攝影機所產生的極平面圖像也會不同。攝影機鏡頭的設置主要分為三種：(1)全部攝影機鏡頭方向都和基線垂直，如圖 2-7(a)，這也是最簡單的情形。其極平面圖像如圖 2-7(b)。(2)每台攝影機鏡頭方向和基線間的夾角 ϑ 為定值，且 $\vartheta \neq 90^\circ$ ，如圖 2-8(a)，其極平面圖像如圖 2-8(b)。(3)每台攝影機鏡頭方向和基線間的夾角大小皆不固定，如圖 2-9(a)，產生的極平面圖像如圖 2-9(b)。

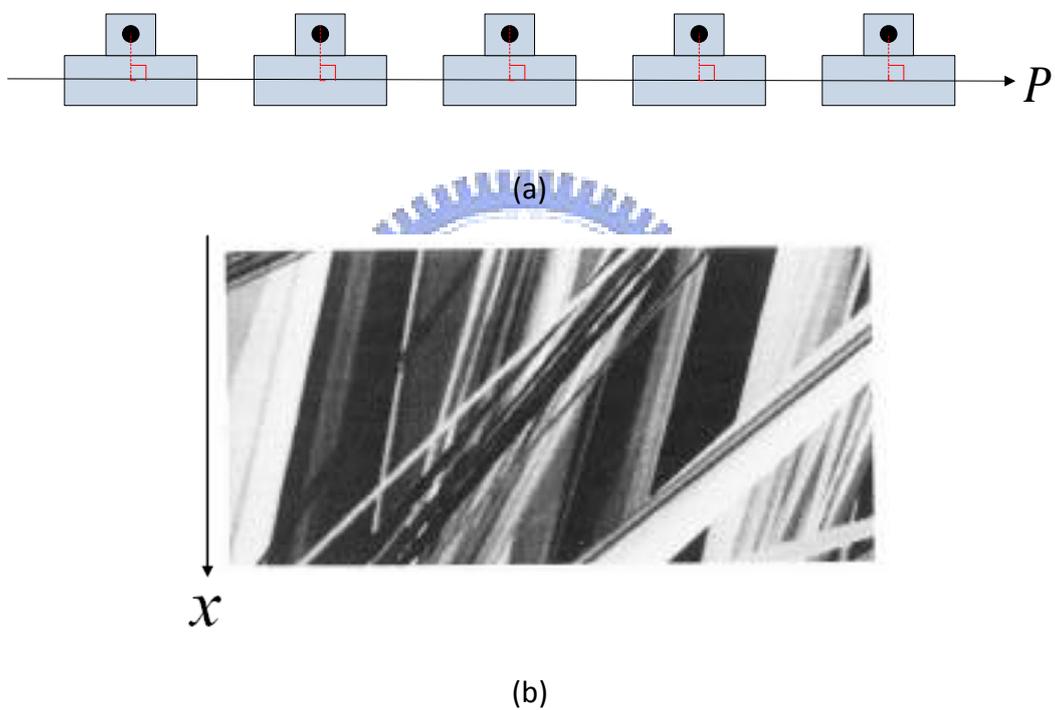
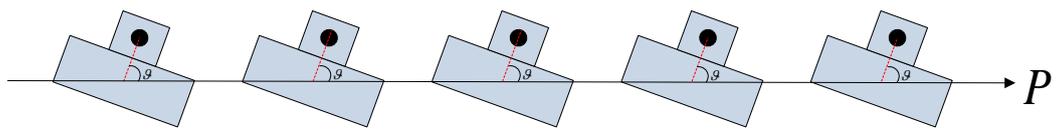


圖 2-7 (a)相機鏡頭方向和基線垂直的設置結構

(b)此種鏡頭設置下所得到的極平面圖像 [7]，其中縱軸為攝影機在基線上的座標

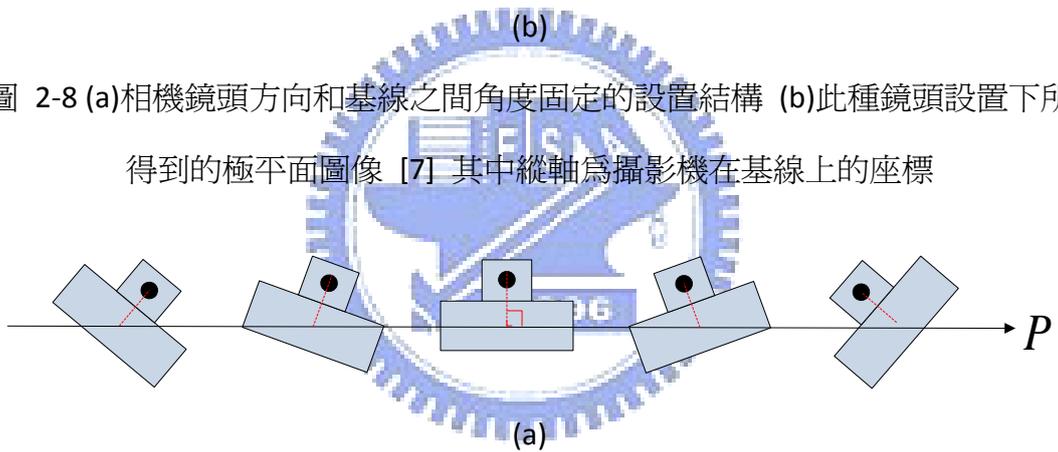


(a)



(b)

圖 2-8 (a)相機鏡頭方向和基線之間角度固定的設置結構 (b)此種鏡頭設置下所得到的極平面圖像 [7] 其中縱軸為攝影機在基線上的座標



(a)



(b)

圖 2-9 (a)相機鏡頭方向和基線之間角度不固定的設置結構 (b)此種鏡頭設置下所得到的極平面圖像 [7]

由於在第一種攝影機鏡頭設置的情況下，製成極平面圖像的過程可以簡化成將每一張攝影機成像上同樣高度的水平掃描線取出並排列。此外，此種鏡頭設置方式所得到的極平面圖像能夠充分地描述影像上的物體在三維空間中的位置關係，所以大多數的攝影機陣列都採用鏡頭和基線方向垂直的設置。

2.3 極平面圖像和三維空間的關係

爲了描述空間中物體位置和極平面圖像紋理之間的關係，我們以圖 2-10 作爲範例。在圖 2-10 中分別爲三個和攝影機陣列距離不同的平面以及一個給定的極平面，而圖 2-11 則是對應該極平面所得到的極平面圖像。可以發現空間中的各平面都會在極平面圖像上形成了一條由右上往左下延伸的帶狀紋理，距離攝影機越近的平面所對應到的帶狀紋理斜率越小，而當平面在無窮遠處時所對應帶紋理形斜率爲 0。這是因爲空間中的物體投影到攝影機成像平面時，同一件物體在不同影像上的位置變化會呈現平移的關係，距離攝影越近的平面，在不同的攝影機成像上的位置移動就會越大，對應到極平面上的圖像紋理斜率會越小。此外可以發現在極平面圖像上，同一影線平面點位置，斜率較小的帶狀紋理會覆蓋斜率較大的帶狀紋理。

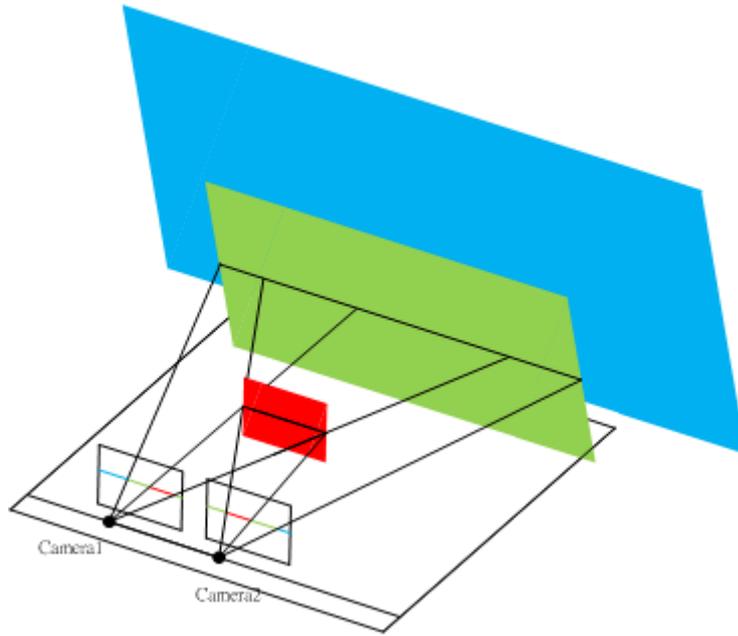


圖 2-10 攝影機和三張不同距離的平面在空間中的關係

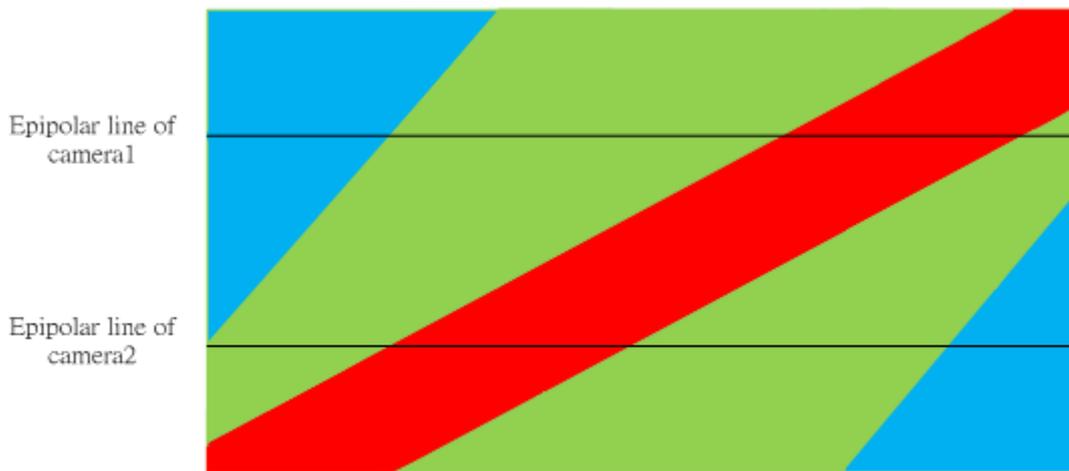


圖 2-11 在圖 2-10 中所得到的極平面圖像

圖 2-12 以實際影像說明空間中的物體位置與極平面圖像紋理之間的關係。圖中在極平面圖像下面的是最右邊攝影機擷取影像的下半部份，而在極平面圖像上面的是最左邊攝影機擷取影像的上半部。可以明顯看出影像中遠景的區域，例如後方的牆壁(紅色標記處)，在極平面圖像形成的帶狀紋理斜率較大；而影像中前景區域，例如影像中的盆栽(綠色標記處)，所對應的條狀紋理斜率較小。

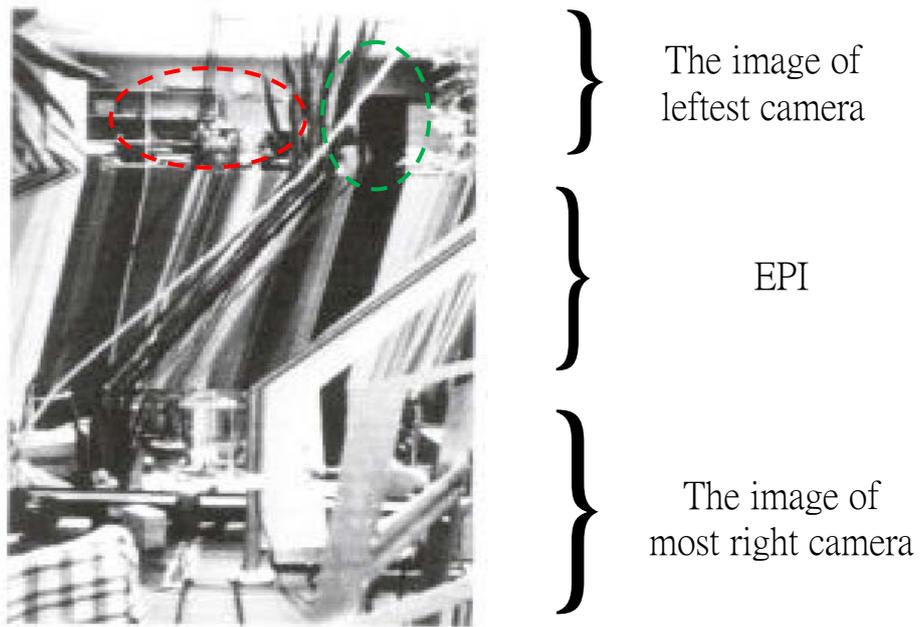


圖 2-12 真實影像所得到的極平面圖像和攝影機擷取影像的關係 [7]

由以上的描述，我們可以知道帶狀紋理的斜率大小和彼此覆蓋的現象包含了三維空間中物體位置資訊。以下我們將討論攝影機陣列、極平面圖像以及三維空間中的物體彼此之間的關係。

爲了方便討論，我們將攝影機陣列置於三維空間中 $Z = 0$ 上。在三維空間中的一點 (X, Y, Z) 和攝影機之間的位置關係如圖 2-13，需要注意的是圖中 Z 軸的正方向是向下，且 P 軸和 X 軸方向一致。 θ 爲攝影機和 $Z = 0$ 之間的夾角，且攝影機擷取影像的水平座標 $u = \tan \theta$ 。因此該點 (X, Y, Z) 和極平面圖像 $P - u$ 平面間的關係表示式如下：

$$P = X + Zu \quad (1)$$

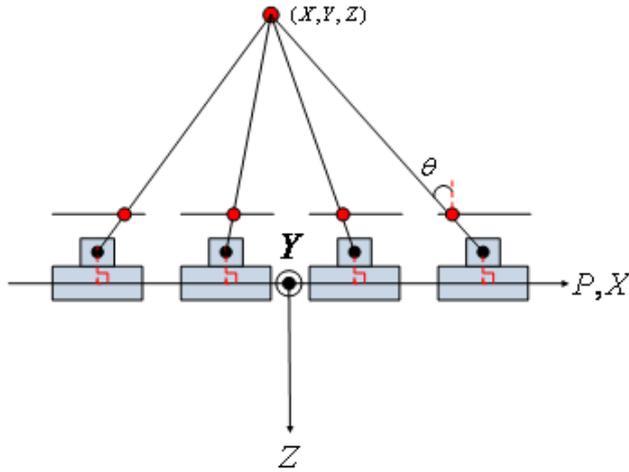


圖 2-13 位於 $Z = 0$ 上的攝影機陣列和空間中一點 (X, Y, Z) 的位置關係

以下我們參考 2-1 式說明在極平面圖像 $P-u$ 上不同的直線方程式對在三維空間的意義[9]：

- (1) 在極平面圖像上一直線 $P = const.$ ，如圖 2-14(a)，相當於在 $Z = 0$ 平面上某一台攝影機擷取影像上的一條水平掃描線。
- (2) 在極平面圖像上一直線 $P = au + b (a < 0)$ ，如圖 2-14(b)，相當於在三維空間中 $Z < 0$ 的一個物件點。
- (3) 在極平面圖像上一直線 $P = au + b (a > 0)$ ，如圖 2-14(c)，相當於一個在 $Z > 0$ 的視點所看到的影像的一條水平掃描線。

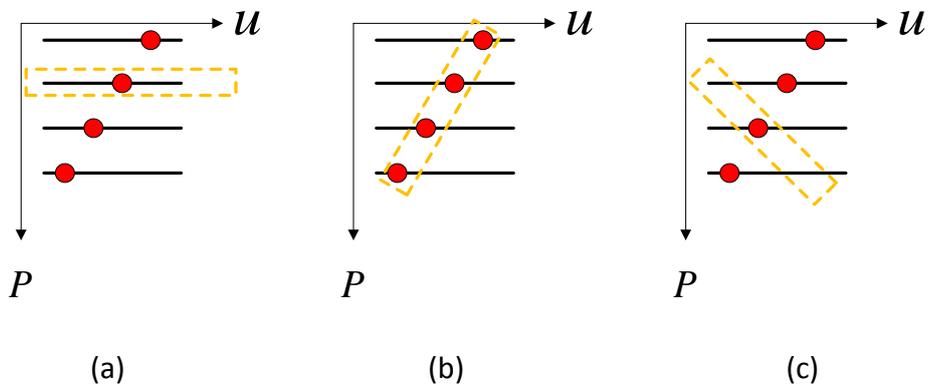


圖 2-14 極平面圖像上各種不同直線的示意圖 (a) $P = const.$ (b) $P = au + b (a < 0)$

(c) $P = au + b (a > 0)$

從上述的說明中可以知道，藉由極平面圖像所提供的資訊以及對應再三維空間中的關係，我們能夠合成出空間中不同視點所對應的虛擬影像。

2.4 不連續極平面圖像

要得到一個理想的極平面圖像，其中最重要的條件就是每張影像之間必須要有高度的空間相關性；換言之，攝影機彼此之間的距離必須很小。若攝影機彼此間的距離過大，則同一個物件在極平面圖像上會發生不連續的現象，如圖 2-15。然而，現實中的攝影機陣列受限於硬體上的因素，往往無法達到此要求。在本論文第四章中，我們提出了參考視差(disparity)資訊來內插兩條極線間的資訊，以求模擬出理想的極平面圖像。

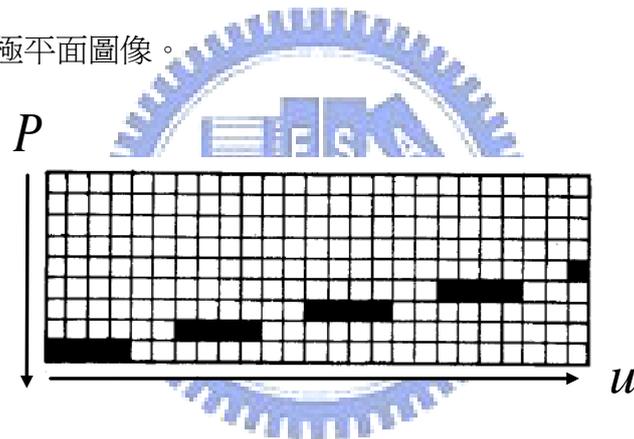


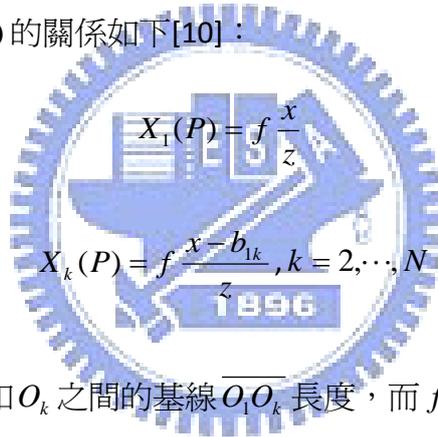
圖 2-15 極平面圖像的不連續情形 [7]



第三章 使用動態規劃進行視差估測

3.1 視差定義

假設在一個多重視點系統中有 N 台針孔攝影機 O_k ($k = 1, \dots, N$)，其中攝影機之間的光軸彼此互相平行，且基線皆對齊 x 軸，如圖 3-1 所示。我們令 I_k 為攝影機 O_k 之擷取影像，並以攝影機 O_1 為中心，令三維空間中的一點 P 的座標為 (x, y, z) 。若 $X_k(p)$ 表示點 P 投影至影像 I_k 上的水平座標位置，則由幾何光學我們得知 $X_k(p)$ 和 (x, y, z) 的關係如下[10]：



$$X_1(P) = f \frac{x}{z} \quad (2)$$

$$X_k(P) = f \frac{x - b_{1k}}{z}, k = 2, \dots, N \quad (3)$$

其中 b_{1k} 表示攝影機 O_1 和 O_k 之間的基線 $\overline{O_1O_k}$ 長度，而 f 表示鏡頭焦距。

假如點 P 皆出現在影像 I_1 以及影像 I_k 中，則我們定義點 P 在影像 I_1 和 I_k 之間的視差(disparity) $d_{1k}(P)$ 為：

$$d_{1k}(P) = X_1(P) - X_k(P). \quad (4)$$

結合(2)、(3)式，我們可以得到視差和攝影機之間的關係如下：

$$d_{1k}(P) = f \frac{b_{1k}}{z}. \quad (5)$$

觀察上式可以發現在固定焦距以及基線長度的情況下，視差大小和 z 值為一個反比關係；換言之，若空間中某一物體和攝影機之間的垂直距離越小，則該物體在不同的影像中的位置變化會越大，若物體在距離攝影機無窮遠處，則物體在不同

的影像中的位置是固定的。

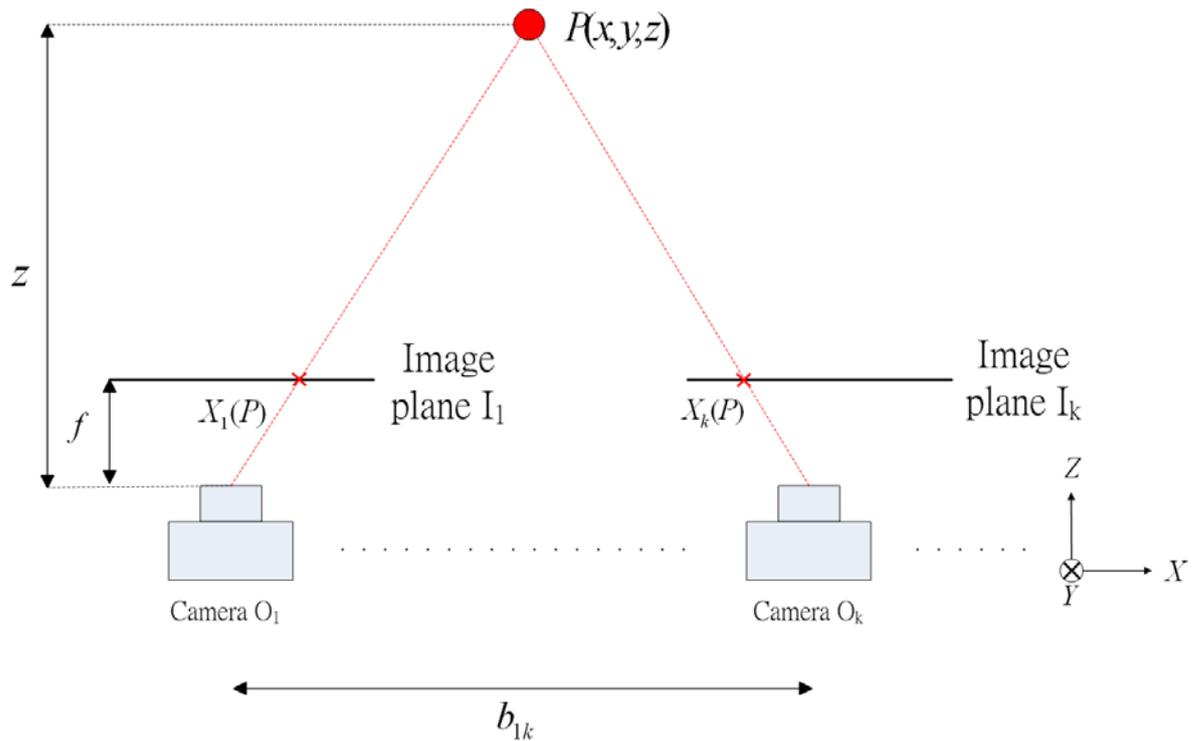


圖 3-1 典型多重視點系統[10]

圖 3-2 為一個雙重視點系統。在這個例子中，區段 \overline{AB} 、 \overline{GH} 以及 \overline{EF} 上的各點皆會出現在影像 I_1 和 I_2 中，所以在這些區段中的各點視差值皆是明確的；但受到前方區段 \overline{DE} 遮蔽的影響，區段 \overline{BC} 僅會顯示在影像 I_1 上，因此在 I_2 中不存在對應於 $\overline{B'G'}$ 的區段，導致我們無法得知區段 $\overline{B'G'}$ 的視差值大小。這類因遮蔽現象造成無法得知視差值的區域被稱為 occlusion。同樣地，影像 I_2 中的區段 $\overline{H'E'}$ 也無法在 I_1 前找到對應區段，所以 $\overline{H'E'}$ 也是一個 occlusion。判別 occlusion 的區域在視差估測中是個很重要的問題，因為在多重視點系統中，Occlusion 必定會伴隨著物體前後位置差異發生。

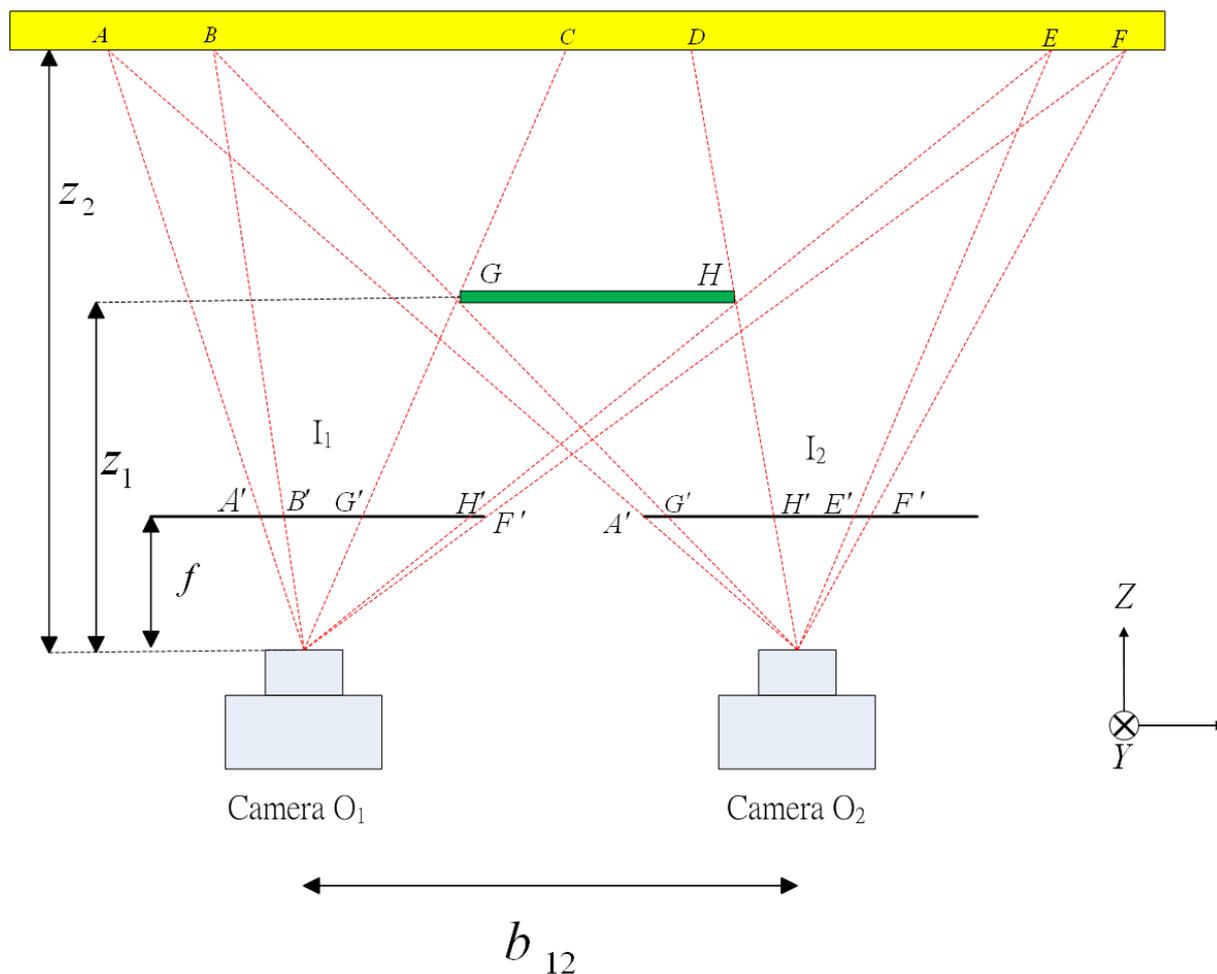


圖 3-2 雙重視點系統[10]

3.2 使用動態規劃進行視差估測

動態規劃(dynamic programming)是在處理整體最佳化演算法中常使用的方法。動態規劃最大特色是可以將 N 階決策程序轉換成 N 個子決策程序，並藉由最佳化每個子程序來求得整體程序的最佳解。這個特色使得演算法的時間複雜度可以從指數級降為多項式時間。然而，動態規劃並非沒有使用上的限制。一個決策程序必須符合以下兩個條件才能使用動態規劃：(1)該程序必須具有最優子結構性質，這表示對於一個最優化策略，其子策略必皆為最優的。一個具有最優子結構特性的程序又被稱為滿足最優化原理。(2)該程序必須具有無後向性，這表

示在一個已排序的程序中，對某個給定的決策階段，其結果不受之前各階段的決策結果影響，而僅由當前階段決定。換句話說，每一個決策階段彼此間都是互相獨立的。一個具有無後向性的程序又被稱為為一個馬可夫程序(Markovian Process)。

3.2.1 二維搜尋平面

視差估測可以視為在一對經過攝影機校正的立體圖片上，對兩邊相同高度的掃描線(scanline)循序進行對應點(corresponding point)的搜尋。對於兩個互相對應的 pixel 組合，我們稱唯一組匹配(match)。圖 3-3 是一個在左右兩台針孔攝影機成像上的掃描線。整個對應點搜尋過程可以使用一個二維搜尋平面(2-D search plane)來描述[11]，如圖 3-4 所示，其中水平軸和垂直軸分別表示左側掃描線以及右側掃描線上的 pixel 位置。因此，兩條掃描線之間最理想的所有匹配集合就相當於一條以搜尋平面第一行及第一列的節點作為起始，和以最後一行及最後一列的節點最為結束的最佳路徑(path)。

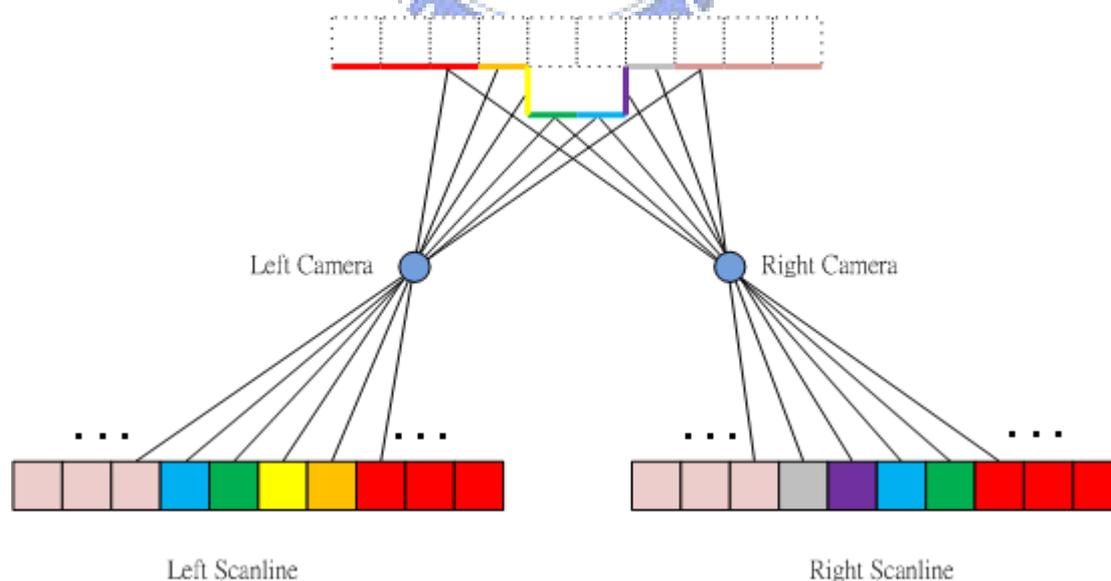


圖 3-3 一對針孔攝影機對同一場景所取得的掃描線

在二維搜尋平面中，每一個節點(node)皆代表一組對匹配。如圖 3-5 所示，對於某個給定的節點，前一個可能和它連接的節點只有三種情形：(1)前一組匹配在左側掃描線上發生 occlusion，兩個節點間的連接路徑為水平。(2)前一組匹配在左側及右側掃描線上皆緊鄰該組匹配，兩個節點間的連接路徑為斜線。(3)前一組匹配在左側掃描線上發生 occlusion，兩個節點間的連接路徑為垂直。

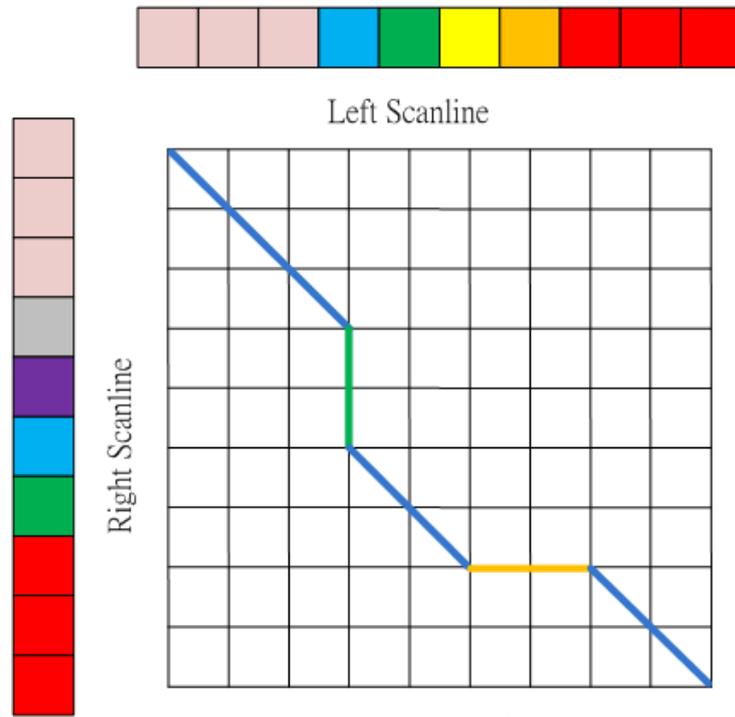


圖 3-4 二維搜尋平面

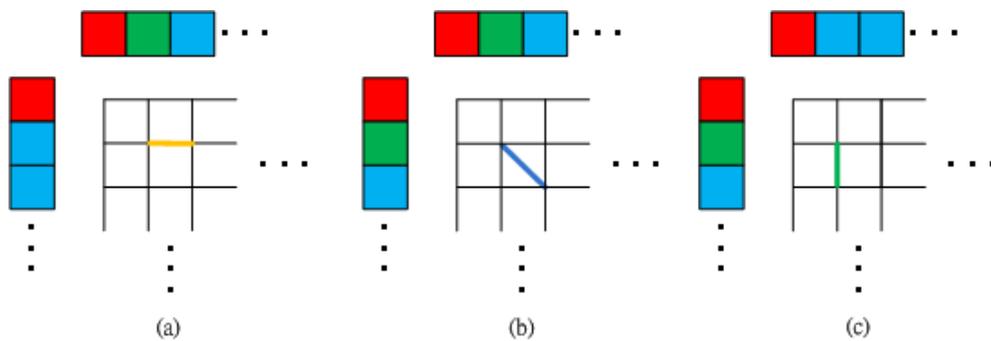


圖 3-5 連結路徑和節點間的關係：(a)相鄰的匹配在右側掃描線上出現 occlusion，節點間路徑為水平 (b)相鄰的匹配掃描線上無 occlusion，節點間連線為斜線 (c)相鄰的匹配在左側掃描線上出現 occlusion，節點間路徑為垂直

由以上敘述，我們可以發現對應點搜尋程序符合使用動態規劃的基本限制：
(1)若某個給定的節點為一組最佳匹配，則和它相連的前一個節點必定也是最佳匹配(最優子結構性質)。(2)一個節點在選擇連接前一個節點的路徑時，僅受到前一個節點的狀態影響，而和之前的路徑的選擇結果無關(無後向性)。因此我們能利用動態規劃的方式來實現對應點搜尋程序。

使用動態規劃時，搜索平面上的每一個節點都會紀錄它連結至起始節點(start node)所需要的最低 cost。而對於一個給定的節點，它在選擇連結路徑時，只需要考慮可能連結到的節點所紀錄的累加 cost 還有與其相連所花費的 cost 即可。我們保留兩種 cost 相加結果最小的路徑來決定何者為前一個與它連結的節點，並將 cost 相加結果作為該節點連結至起始節點所需要的最小 cost。當所有節點都處理完後，我們在搜尋平面最後一行及最後一列上找出累加 cost 最小的結束節點(end node)，並追溯它和起始節點間的連接路徑。由於最佳子系統的特性，該條連結路徑必為一個整體最佳解，而連結路徑上的各節點也會是兩條掃描線中最佳的一組匹配集合。



3.2.2 樹狀搜尋演算法

在上一節中，我們雖然將搜尋對應點程序轉換成在一個二維搜尋平面上尋找整體最佳路徑，但是這在實際上仍然不是一個容易且直觀的方法。樹狀搜尋演算法(tree search algorithm)提供了一個稱為立體樹(stereo tree)的樹狀結構來描述二維搜尋平面[13]。立體樹可以幫助我們以矩陣的形式來解決尋找最佳路徑的問題。

在樹狀搜尋演算法中將一個匹配標記為 $M = (M_{xL}, M_{xR})$ ，其中 M_{xL} 和 M_{xR} 分別表示 pixel 在左側掃描線與右側掃描線上的位置。當一個有序的匹配集合 $\{M_i\}_{i=1, \dots, N}$ 符合以下四個條件時，則我們稱呼該集合為一組匹配序列(match

sequence) :

$$C1: d_{\min} \leq d \leq d_{\max}, \text{ 其中 } d \text{ 表示視差值} \quad (6)$$

$$C2: M_{i,xL} < M_{j,xL} \ \& \ M_{i,xR} < M_{j,xR}, \text{ 其中 } 0 \leq i < j < N_m - 1 \quad (7)$$

$$C3: (M_{i+1,xL} = M_{i,xL} + 1) \cup (M_{i+1,xR} < M_{i,xR} + 1), \quad (8)$$

其中 $i = 0 \dots N_m - 2$

$$C4: (M_{0,xL} = 0) \cup (M_{0,xR} = 0) \quad (9)$$

第一個條件是對於任一組有效的匹配，其對應的視差值都必須落在預設的範圍內。第二個條件是規定在一個匹配序列中，任何一個 pixel 僅能出現一組匹配中，且該序列中的每一組匹配彼此之間要符合排序上的關係。第三個條件是表示在一組匹配序列中不允許出現缺口；換言之，在任何一組匹配中，無論是在左側掃描線上或右側掃描線上，至少要有一個 pixel 跟下一組匹配中的 pixel 相鄰。第四個條件則是匹配序列的第一組匹配一定要包含左側掃描線或右側掃描線的第一個 pixel；若是該匹配序列包含左側掃描線或右側掃描線的最後一個 pixel，則稱該匹配續列為完整的(complete)。

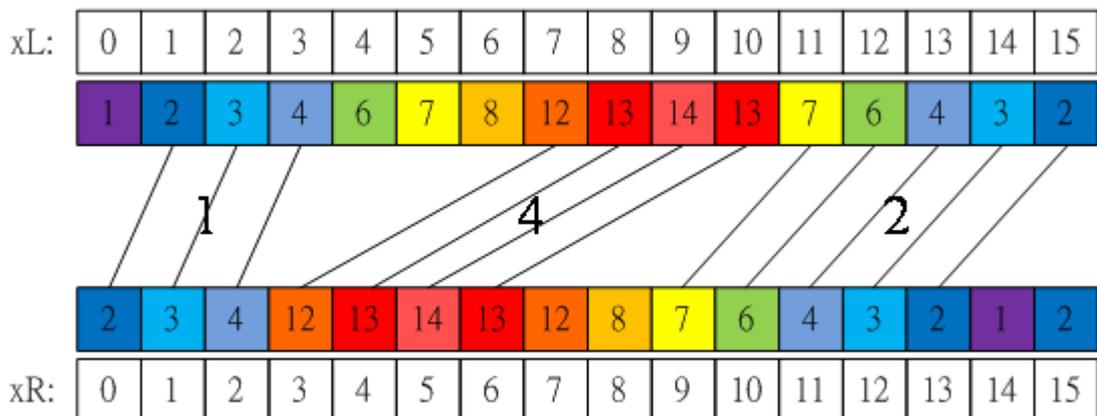


圖 3-6 一個符合樹狀搜尋演算法條件的匹配序列

圖 3-6 為一個符合匹配序列條件的範例。在這個例子中可以明顯看出最理想的匹配續列為：

$$MS = \{(1,0), (2,1), (3,2), (7,3), (8,4), (9,5), (10,6), (11,9), (12,10), (13,11), (14,12), (15,13)\} \quad (10)$$

為了表達上的方便及編碼上的效益，通常會用視差序列(disparity sequence)來取代匹配序列。上述範例中的視差序列表達方式如下：

$$DS = \{Undef, 1, 1, 1, Undef, Undef, Undef, 4, 4, 4, 4, 2, 2, 2, 2\} \quad (11)$$

其中 *Undef* 表示該位置的 pixel 為 occlusion。

3.2.3 代價函數

在定義了匹配序列後，我們將在本節討論如何計算一個匹配序列(或視差序列)所對應的 cost。進行對應點搜尋時，一般所採用的代價函數主要包含兩部份：第一部份是針對未被遮蔽的 pixel。在視差序列中，一組未被遮蔽的 pixel 的 cost 為兩個 pixel 之間的差異值，因此又被稱為差異函數(dissimilarity function)。我們會在下一段中做較詳細的討論。第二部份是針對受到遮蔽效應影響的 pixel。若一個 pixel 在視差序列中被判斷為 occlusion，則其 cost 設為一個定值 β 。當發生 occlusion 的區域越大，這類 cost 在視差序列中的比例也會呈現線性增加。需要注意的是，不論是左側掃描線或右側掃描線的 occlusion 區域都需要列入考量。綜合以上敘述，典型的視差序列代價函數可以用以下方程式來表示：

$$C(DS) = \sum_{xL=0}^{W-1} (D(xL, xR) |_{match}) + \sum_{xL=0}^{W-1} (\beta |_{occlusion}) + \sum_{xR=0}^{W-1} (\beta |_{occlusion}) \quad (12)$$

差異函數主要用來描述左側掃描線上 xL 位置的 pixel 和右側掃描線上 xR 位置的 pixel 兩者間的相似程度。它同時也是影響整體 cost 的關鍵，因此設計一個

良好差異函數就顯得非常重要。以下我們介紹兩個常見的差異函數以及一個經過改良的差異函數

1. Pixel-Based Dissimilarity Function

Pixel-Based Dissimilarity Function 為一個最基本差異函數[12][14]。當我們欲判斷在左側掃描線及右側掃描線上兩個 pixel 的相似程度時，它僅僅計算兩個 pixel 之間差值的絕對值(absolute)或平方值(square)，如圖 3-7。其表示式如下：

$$D(xL, xR) = |I(xL) - I(xR)|, \text{ (Absolute)} \quad (13)$$

$$D(xL, xR) = (I(xL) - I(xR))^2, \text{ (Square)} \quad (14)$$

其中 I 表示 pixel 的亮度大小。Pixel-Based Dissimilarity Function 的優點為簡單和計算速度快，不過它並不是一個可靠方法。除了容易受到雜訊影響外，它處理平緩變化或是週期性紋理的區域時很容易得到錯誤的結果。

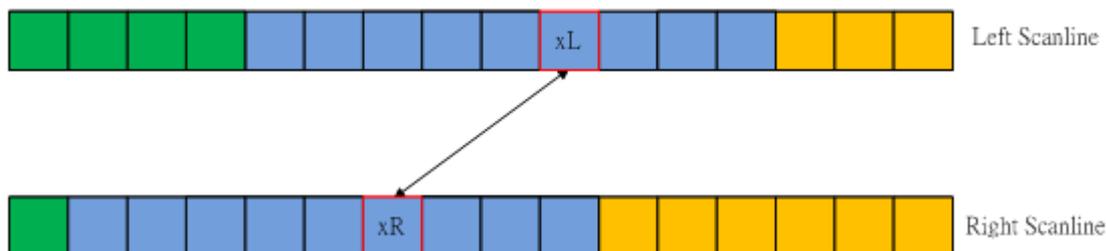


圖 3-7 Pixel-Based Dissimilarity Function

2. Window-Based Dissimilarity Function

Window-Based Dissimilarity Function[12][14][16]在判斷兩個 pixel 的相似程度時，會以待判斷 pixel 的位置為中心，將該點周圍一定範圍的 pixel 一併列入考量，如圖 3-8。其表示式如下[12][14]：

$$D(xL, xR) = \sum_{k=-R}^R |I(xL+k) - I(xR+k)|, \text{ (Absolute)} \quad (15)$$

$$D(xL, xR) = \sum_{k=-R}^R (I(xL+k) - I(xR+k))^2, \text{ (Square)} \quad (16)$$

爲了增加差異函數的鑑別度，通常會給予視窗中個 pixel 不同的權重 (weight)，因此(14)、(15)式可改寫爲：

$$D(xL, xR) = \sum_{k=-R}^R w(k) \times |I(xL+k) - I(xR+k)|, \text{ (Absolute)} \quad (17)$$

$$D(xL, xR) = \sum_{k=-R}^R w(k) \times (I(xL+k) - I(xR+k))^2, \text{ (Square)} \quad (18)$$

一般來說，權重函數 $w(k)$ 爲一個在 $k = 0$ 時具有極大值並以其爲中心的偶函數。

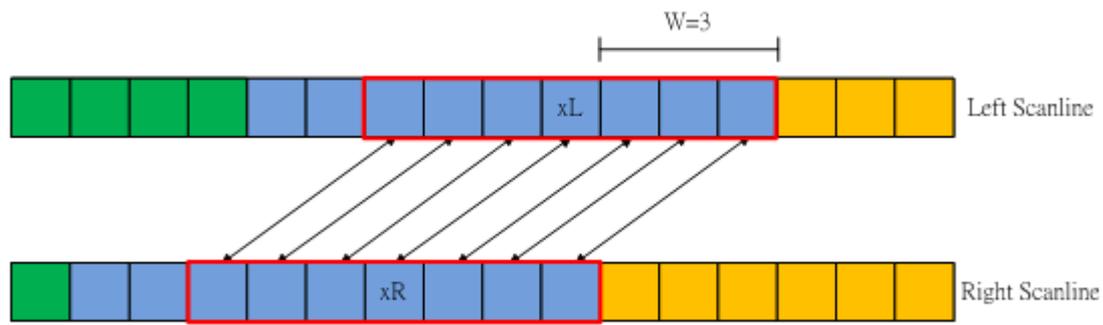
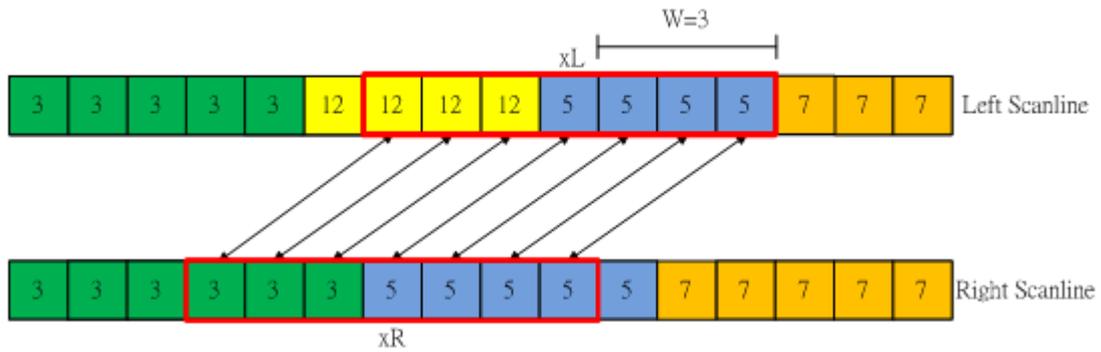


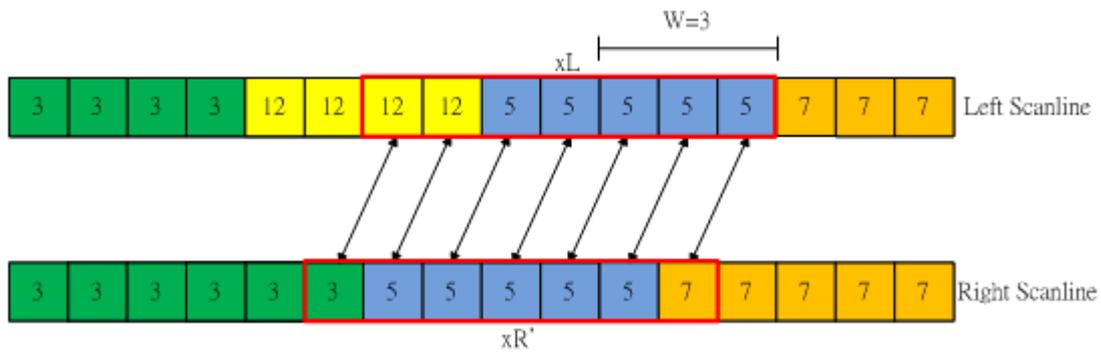
圖 3-8 Window-Based Dissimilarity Function

由於 Window-Based Dissimilarity Function 參考較多的 pixel 作爲對應點間的相似程度判斷，它在抵抗雜訊上有良好的表現，對於平緩變化或是週期性紋理的區域判斷結果也遠比 Pixel-Based Dissimilarity Function 準確。然而，Window-Based Dissimilarity Function 很容易對掃描線中 occlusion 區域周遭的 pixel 做出錯誤判斷，例如圖 3-9，其中掃描線上數字代表各 pixel 的亮度大小。在圖 3-9 (a)是以 (xL, xR) 爲掃描線中的一個理想匹配，但是在左側掃描線上的黃色區塊在右側掃描線上被綠色區塊覆蓋，導致視窗左半部份的 cost 大量增加；而在圖 3-9 (b)中， (xL, xR') 雖然不是最理想的匹配，但是橘色區塊亮度和藍色區塊接近，造成(b)的總 cost

比(a)還小，因此造成判斷上的錯誤。



(a)

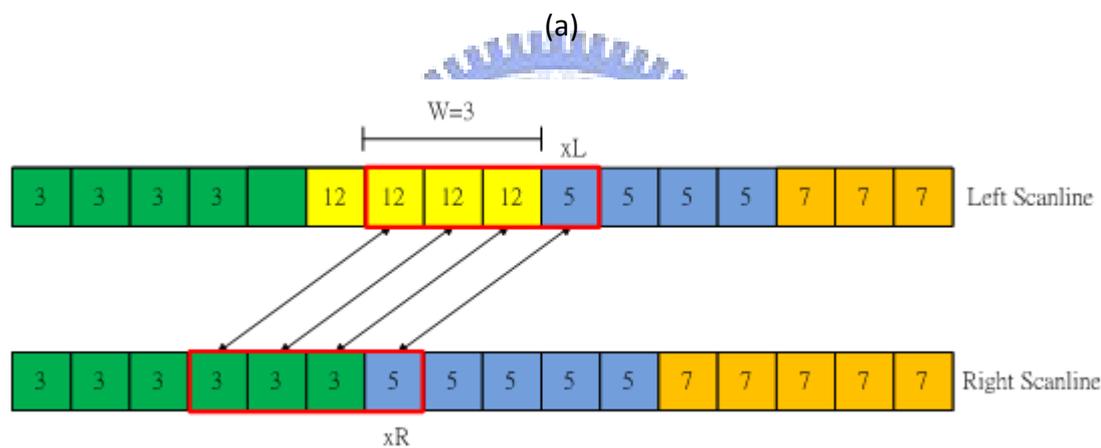
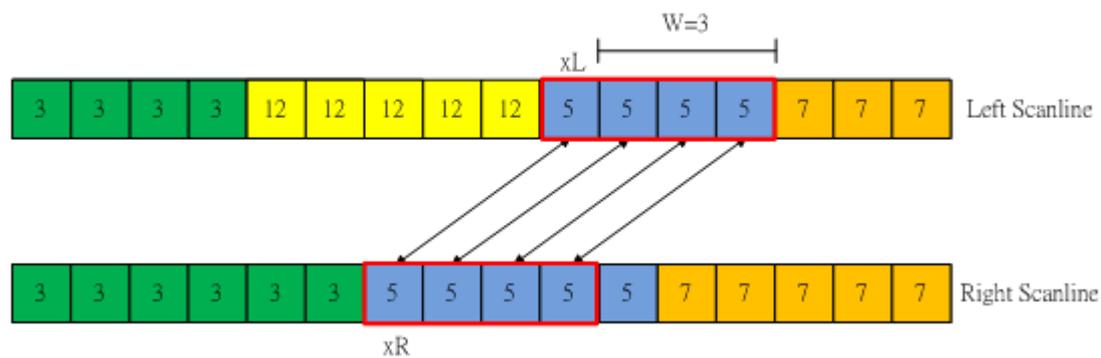


(b)

圖 3-9 (a)正確的匹配(x_L, x_R)所在位置及其視窗範圍 (b)錯誤的匹配(x_L', x_R')所在位置及其視窗範圍

3. Single-Side Window-Based Dissimilarly Function

Single-Side Window-Based Dissimilarly Function 是由 Window-Based Dissimilarity Function 改良而來的[15]。它除了保有 Window-Based Dissimilarity Function 在抗雜訊的能力以及在處理平緩變化和週期性紋理上的優異性，也能夠有效地避免因為遮蔽效應造成 occlusion 周遭區域的判斷錯誤。



(b)

圖 3-10 Single-Side Window-Based Dissimilarly Function 在匹配(xL,xR)的示意圖：(a)

右側視窗範圍 (b)左側視窗範圍

觀察圖 3-9，我們可以發現因為被物體被遮蔽造成 cost 增加的情形僅出現在視窗的左半邊或右半邊。因此，若僅採用正確部份的區域，就能夠有效避免遮蔽效應所帶來的錯誤。基於以上想法，將原本以待判斷 pixel 為中心的視窗改為兩個單邊視窗，如圖 3-10。這兩個單邊視窗會各自得到一個匹配所需的 cost，而我們會判定 cost 較大的單邊視窗是受到遮蔽效應的干擾，而採用較小的 cost 作為判斷結果。因此，Single-Side Window-Based Dissimilarity Function 可以寫成以下表示式：

$$D(xL, xR) = \min\left(\sum_{k=-R}^0 |I(xL+k) - I(xR+k)|, \sum_{k=0}^R |I(xL+k) - I(xR+k)|\right), \quad (19)$$

(Absolute)

$$D(xL, xR) = \min\left(\sum_{k=-R}^0 (I(xL+k) - I(xR+k))^2, \sum_{k=0}^R (I(xL+k) - I(xR+k))^2\right), \quad (20)$$

(Square)



3.2.4 立體樹

和二維搜尋平面基本結構類似，立體樹也是以節點和連結路徑來描述。每個節點都必須符合條件 C1。若兩個節點間的關係符合條件 C2 和 C3，則它們之間存在一條連結路徑。圖 3-11 為一個以圖 3.6 為範本的立體樹，其視差值範圍設定在 0 到 4 之間，且 cost 計算方式是取兩點之間亮度差異的絕度值。圖中每個圓形內的數對分別是對應到左側掃描線上位置 xL 位置與右側掃描線上位置 xR (即 $xL - d$) 的亮度大小，並展示了一個以匹配(0,0)和匹配(1,0)為起始的可能路徑。立體樹中任何一條完整的路徑必須包含起始節點和結束節點，且路徑方向由立體樹的第一行以及第一列向最後一行及最後一列延伸。基於以上兩種特性，我們可以利用矩陣來實現樹狀搜尋演算法[13]。

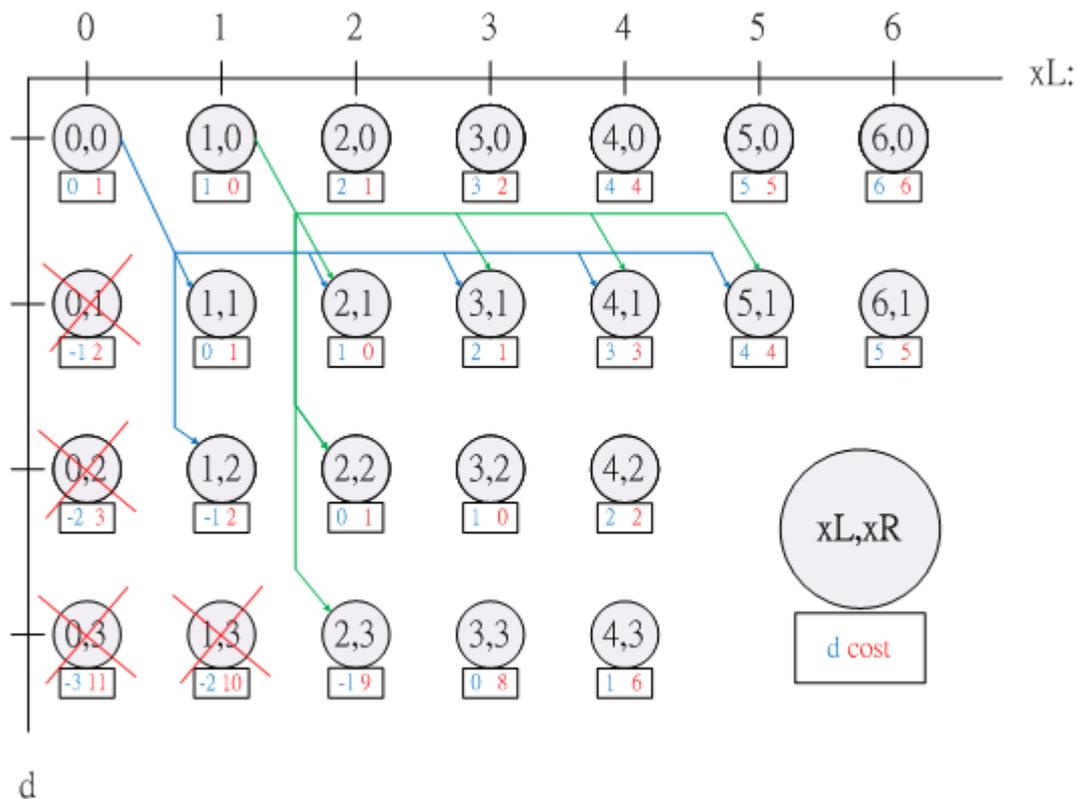


圖 3-11 以圖 3.6 為範例的立體樹，其中打叉的節點為超出預設視差值範圍的匹配

以圖 3-11 為例，我們用三個矩陣來實現該演算法，如圖 3-12，其中矩陣橫軸 xL 代表 pixel 在左側掃描線上的座標，縱軸 d 代表各種可能的視差值。由於每一組匹配間的 dissimilarity cost 是彼此獨立的，且可以快速取得，所以我們會先計算各匹配的 dissimilarity cost 並將結果紀錄在矩陣 γ 。根據立體樹的定義，每一個節點只會有一條和前一個節點相連的路徑。為了能夠明確地知道和前一個節點相連的路徑，我們會在矩陣 Π 中紀錄前一個節點的位置以方便查詢，並設其初始值為 0，若該節點為起始節點，其值則設為 F。最後一個矩陣 Γ 紀錄了每一個節點到其為止所累加的 cost。圖 3-13 是其中一種走法的下一步以及下兩步在矩陣 Γ 與矩陣 Π 上的結果。當所有矩陣都填滿後，我們只需要從矩陣 Γ 上找出記錄著最小 cost 的結束節點，並參考矩陣 Π 所紀錄的節點位置回溯，便可以找出在該立體樹中的最佳路徑。

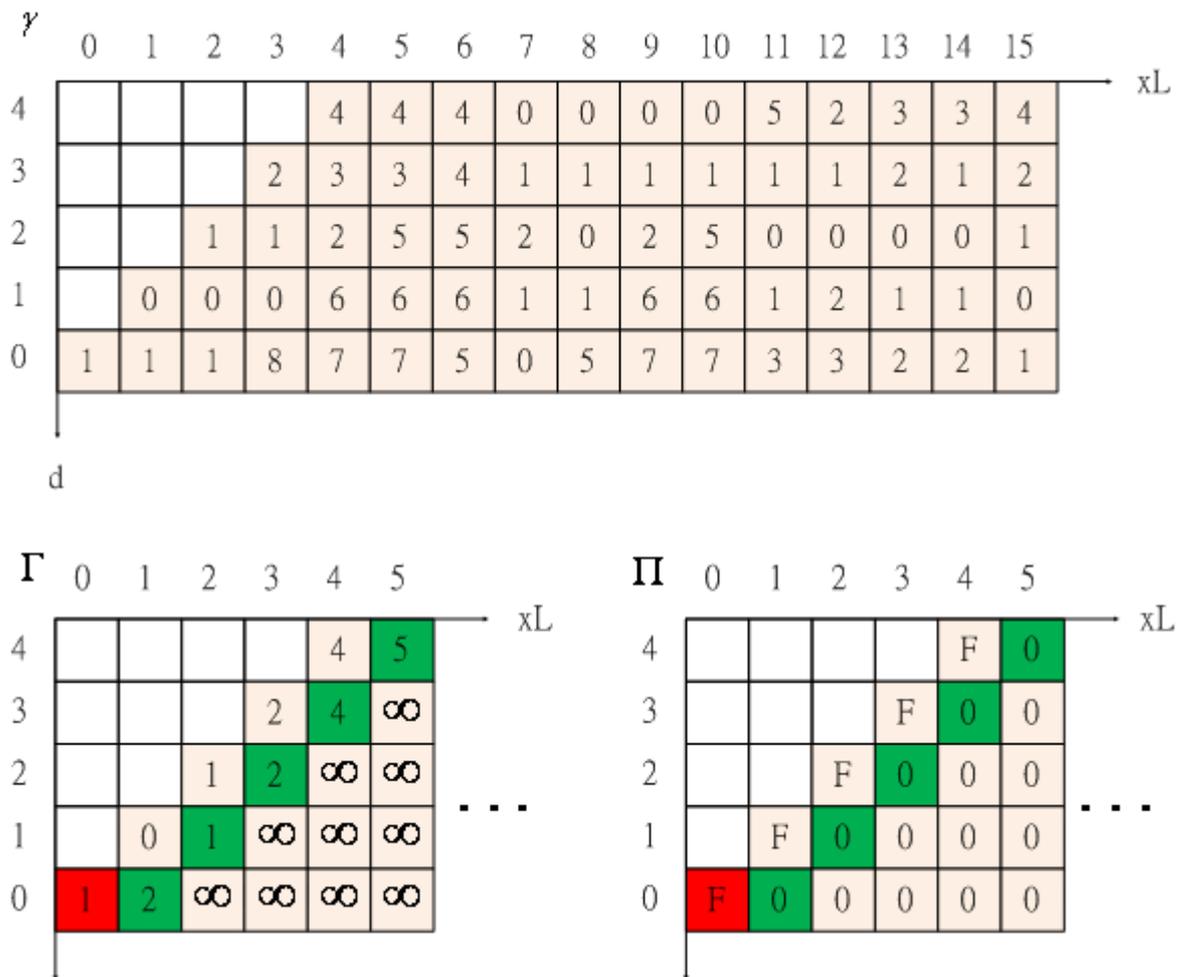
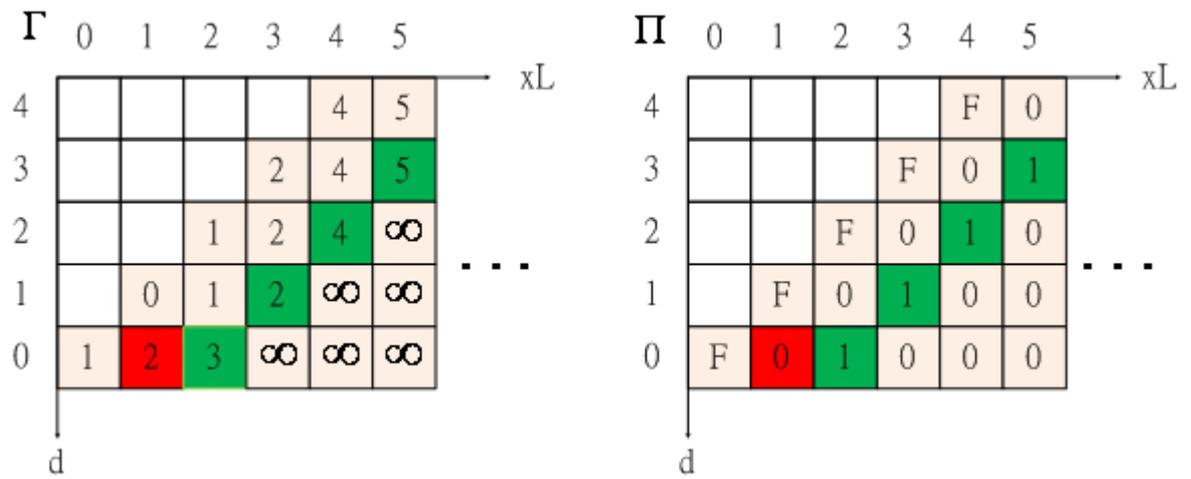
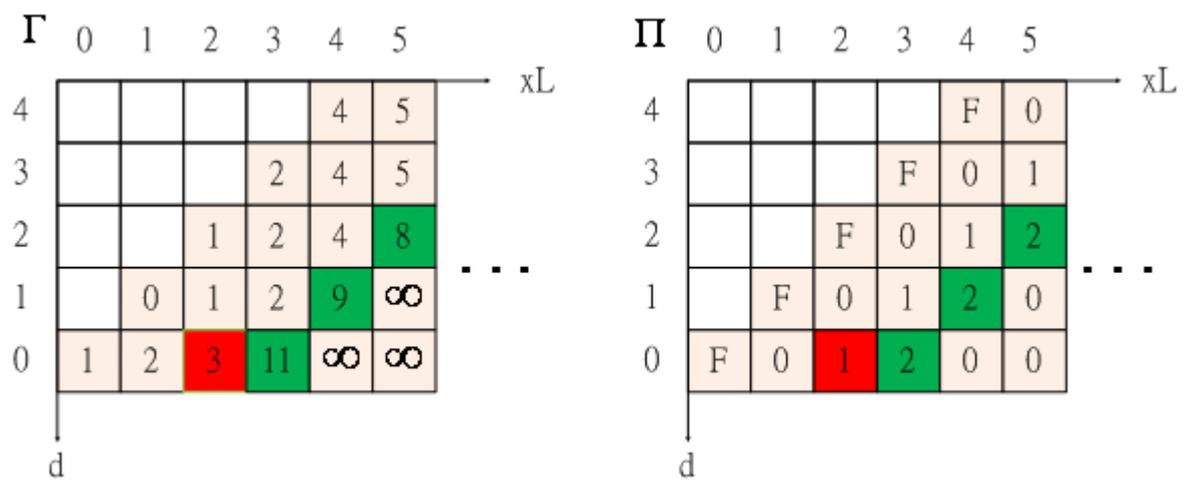


圖 3-12 用來實現樹狀搜演算法的三個矩陣。其中矩陣 Γ 和矩陣 Π 為剛開始處理的狀態。其中紅色的節點為綠色節點前一個可能的連結。



(a)



(b)

圖 3-13 其中一種走法在矩陣 Γ 與矩陣 Π 上(a)走一步的結果 (b)走兩步的結果

第四章 適用於二維攝影機陣列之影像合成方法

在現有自由視點視訊系統中，大部分仍然使用一維攝影機陣列來進行影像擷取，這使得使用者在視點位置的選擇上被限制在一個二維平面上。在本論文中，我們將攝影機陣列從一維擴展為二維，讓使用者的視點可以在三圍空間中自由移動。此外，一般的自由視點視訊系統僅使用攝影機所擷取的影像資料進行虛擬視點影像的合成，因此很容易受到雜訊或物體材質等非人為因素的影響而導致合成影像品質不穩定。為了降低這些外在因素的影響，在本論文中我們使用影像的視差圖作為合成虛擬視點影像時的主要參考資訊。

在本章中，我們將介紹如何使用二維攝影機陣列在空間上的相關性來改善傳統視差估測所會遇到的問題，並利用不同視差圖之間的平移特性來降低整體自由視點視訊系統的計算量。

4.1 傳統視差估測的問題

在傳統的立體影像系統上，通常僅使用兩台水平方向的攝影機來進行視差估測。雖然藉由動態規劃可以得到一張最佳的視差圖(disparity map)，但是仍然存在兩個問題：(1)由於影像中各物體的景深(depth)不同而產生遮蔽效應，造成視差圖上會出現無法定義視差的區域，亦即該區域為 occlusion，如圖 4-1 中藍色區域。(2)在影像中兩個未互相遮蔽的物體，若是兩者位置過於接近，則進行對應點搜尋時會導致它們之間的區域視差值錯誤，例如在圖 4-2 中所標記的部份。當物體和影像邊緣很靠近時，物體和影像邊緣之間的區域也會出現類似的情形，如圖 4-3 標記的區域。其原因是因為該區域涵蓋範圍太小，導致視差值所帶來的 cost 無法累積到讓整體搜尋程序做出正確判斷。

在本論文中，我們以樹狀搜尋演算法和 single-side window-based dissimilarly function 為基礎來加以延伸，並採用下式作為權重函數：

$$w(k) = (R - |k| + 1)^2 \quad (21)$$

其中 R 為 single-side window-based dissimilarity function 中是參考 pixel 的範圍，在實驗中我們設定其大小為 10，而 k 則表示 pixel 在視窗中的位置。在決定一張影像的視差圖時，除了參考水平方向的影像來進行視差估測外，也同時參考垂直方向的影像，並將參考的攝影機數量從兩台增加至五台。在下一節中，我們將介紹如和使用不同位影像的視差資訊來修正傳統方法所遇到的問題。



圖 4-1 因遮蔽效應造成的 occlusion：(a)原始影像 (b)原始影像的視差圖，其中藍色區域為 occlusion 區域

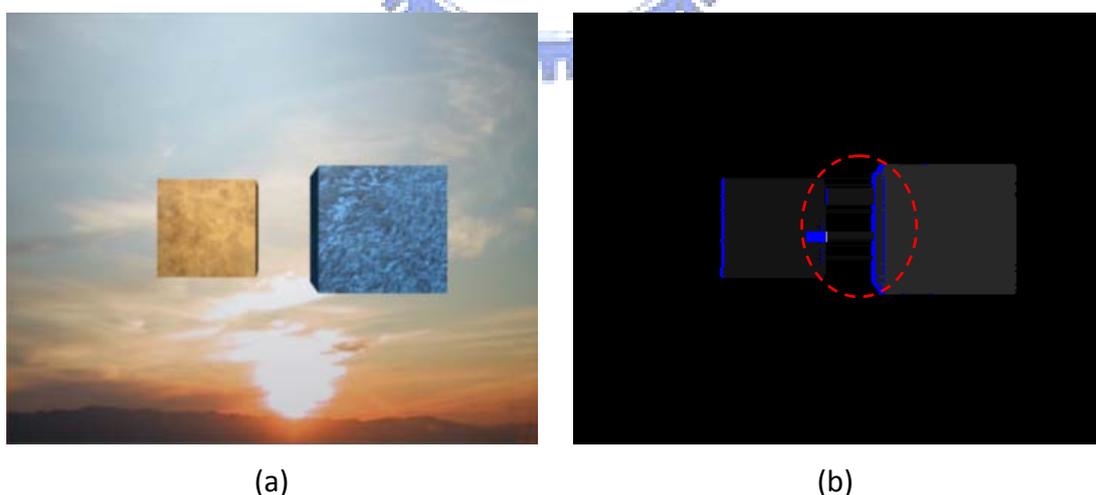


圖 4-2 因為兩物體彼此位置過於接近，導致兩者之間的區域視差值錯誤：(a)原始影像 (b)原始影像的視差圖，其中標記區域為視差值錯誤區域

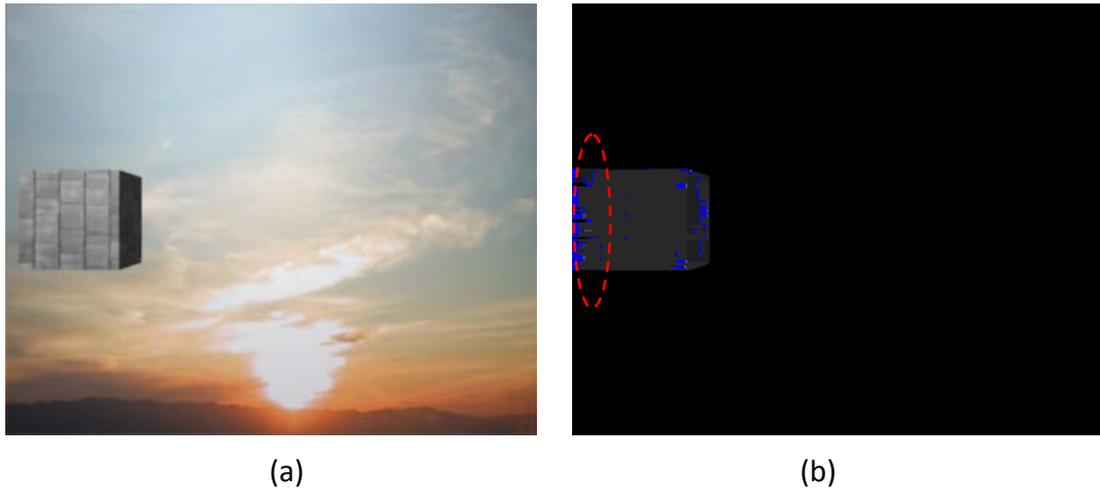


圖 4-3 因為物體位置太接近影像邊緣，導致兩者之間的區域視差值錯誤：(a)原始影像 (b)原始影像的視差圖，其中標記區域為視差值錯誤區域

4.2 參考鄰近影像進行視差估測

在二維攝影機陣列上，對於一台給定的攝影機 O_c 和其鄰近的攝影機位置關係如圖 4-4 所示，其中 O_r 、 O_l 、 O_d 以及 O_u 各表示 O_c 位置的右方、左方、下方、和上方的攝影機。假設圖 4-5 為這五台攝影機所擷取到的影像，並依照所對應的攝影機分別標記為 I_c 、 I_r 、 I_l 、 I_d 以及 I_u 。

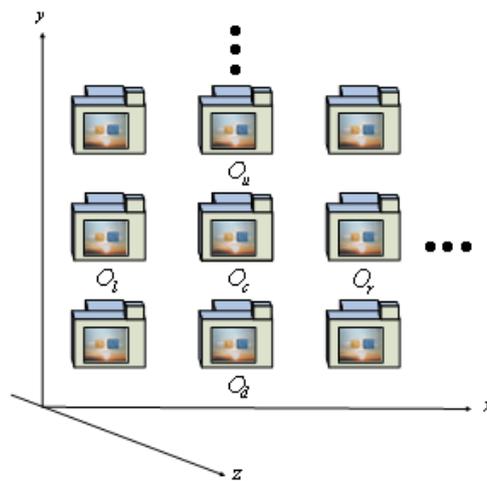
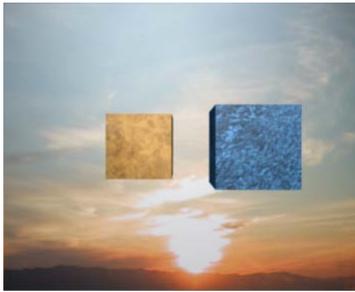


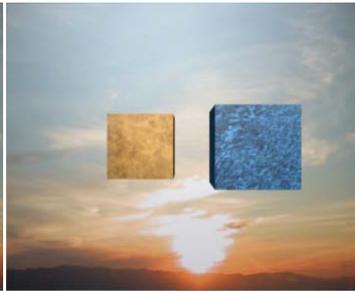
圖 4-4 給定的攝影機 O_c 和鄰近攝影機之間的位置關係



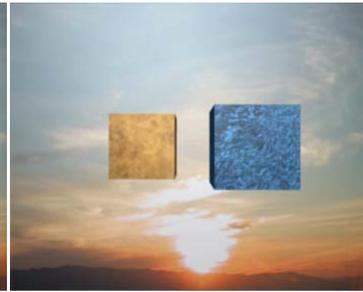
(a)



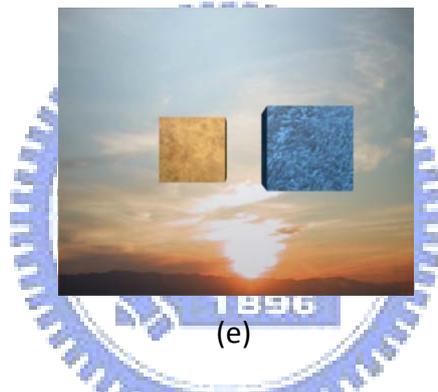
(b)



(c)



(d)



(e)

圖 4-5 攝影機 O_c 和鄰近攝影機的擷取影像：

(a)攝影機 O_u 的擷取影像 (b)攝影機 O_l 的擷取影像

(c)攝影機 O_c 的擷取影像 (d)攝影機 O_r 的擷取影像

(e)攝影機 O_d 的擷取影像

若要計算影像 I_c 的視差圖，傳統作法只會參考右邊的影像 I_r 。我們將此方法延伸，以 I_c 為中心，分別參考周遭四張影像進行視差估測，其結果如圖 4-6。需要注意的是，參考影像的位置不同，進行對應點搜尋時的掃描線選取和搜尋方向

也會跟著改變。例如，以右邊攝影機的擷取影像 I_r 作參考，搜尋時是以 I_c 和 I_r 上同樣高度的水平掃描作為比較，並以掃描線最左邊的 pixel 作為起始點向右搜尋，如圖 4-7(a)；若參考下方攝影機的擷取影像 I_d ，則是比較 I_c 和 I_d 上同樣位置的垂直掃描線，並以掃描線最上方的 pixel 作為搜尋的起始點，如圖 4-7(b)。

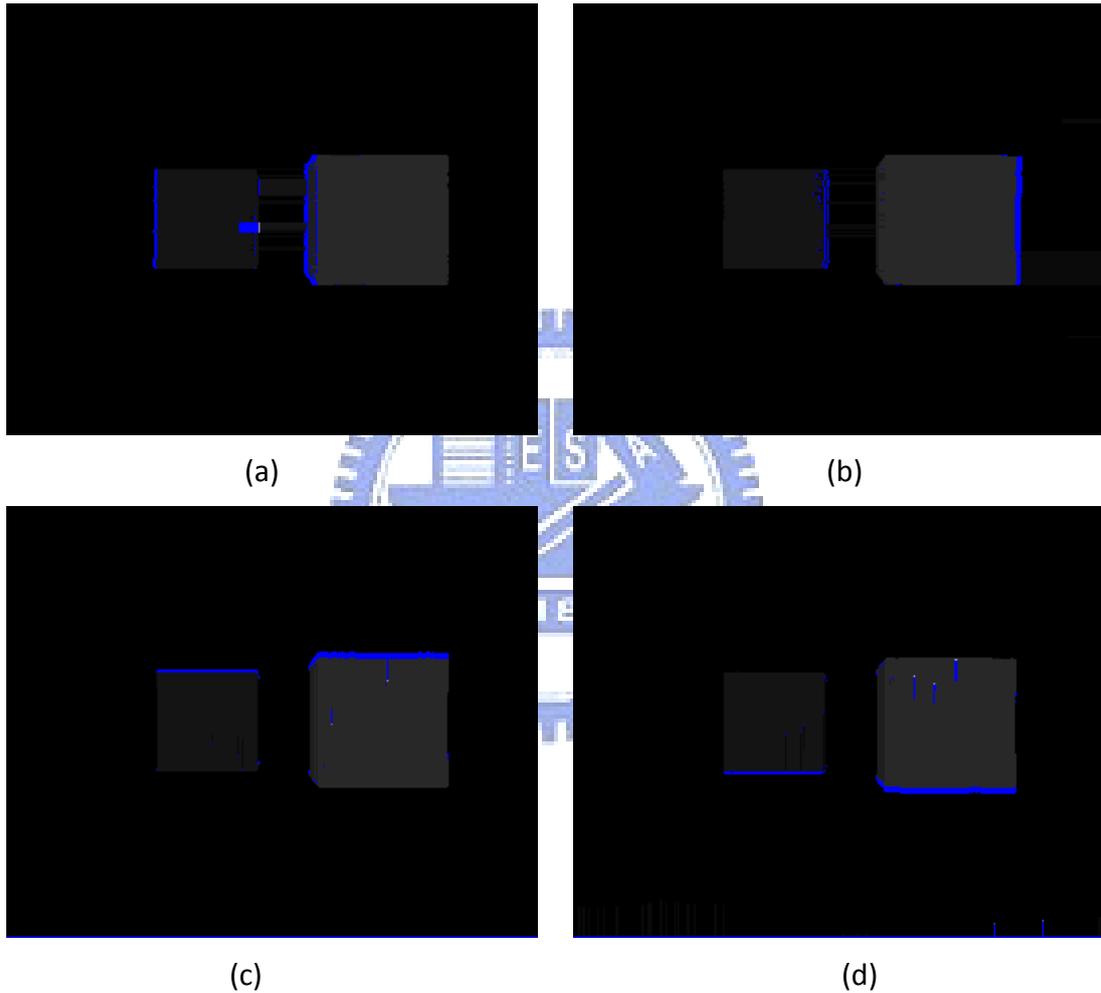
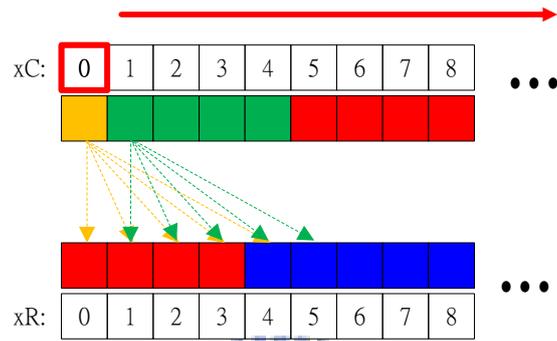


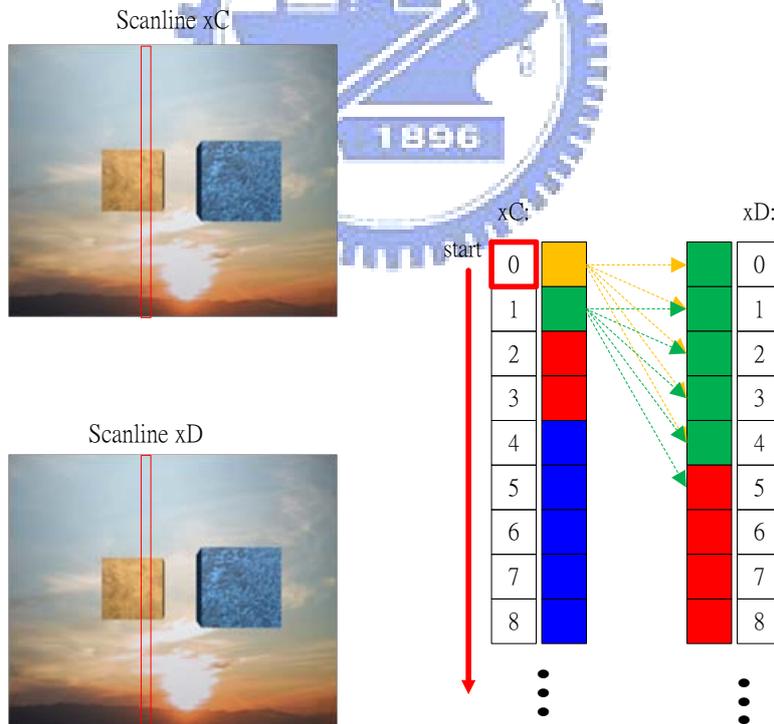
圖 4-6 參考不同位置的攝影機所得到的視差圖：

(a)參考攝影機 O_r (b)參考攝影機 O_l

(c)參考攝影機 O_d (d)參考攝影機 O_u



(a)



(b)

圖 4-7(a)參考右邊影像時所選取的掃描線及搜尋方向 (b) 參考下方影像時所選取的掃描線及搜尋方向

爲了從不同搜尋方向的視差圖中得到一個較理想的結果，我們採用投票的方式來決定影像 I_c 中每個 pixel 視差值大小。對於影像 I_c 中一個給定位置的 pixel P ，最直接的投票方式就是選擇在不同視差圖上，同樣位置出現次數最多的視差值作爲該 pixel 應有的視差值。但這顯然不是一個很客觀的方法，因此我們進行投票時會以 P 爲中心設置一個視窗，並將視窗內視差資料一併列入考慮。以下我們介紹三種不同視窗：

(1) 視窗和掃描線方向無關

若不考慮掃描線的方向，則視窗是一個以點 P 爲中心的方形，如圖 4-8，其中 X 表示該點爲 occlusion。我們統計四張視差圖中在視窗內各視差值出現的次數(occlusion 區域不列入統計)，並選擇出現次數最多的結果作爲點 P 的視差值。

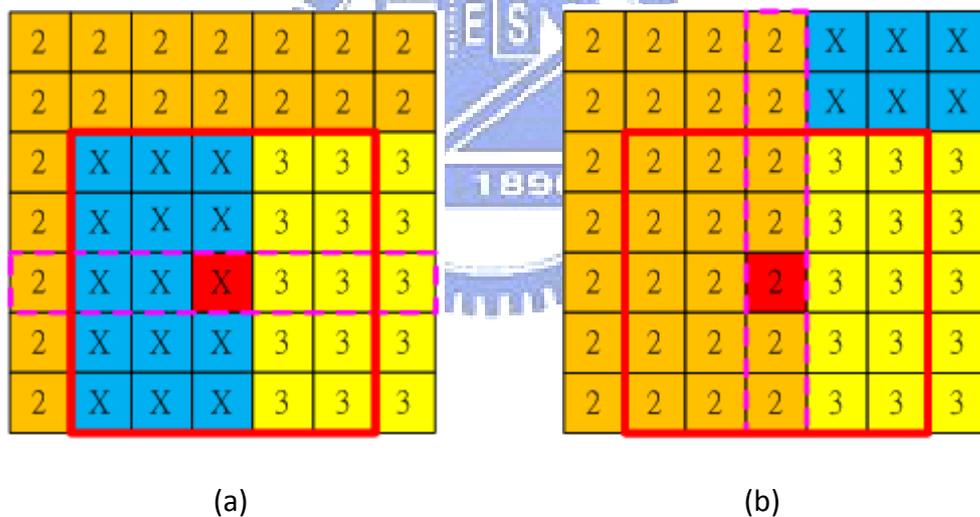


圖 4-8 視窗和掃描線方向無關的情形，點 P 爲圖中紅色的區塊，方形紅框爲視窗，紫色虛線標記爲掃描線位置：(a)掃描線方向爲水平 (b)掃描線方向爲垂直

(2) 視窗和掃描線方向相同

若視窗方向和掃描線方向相同，則視窗視一個以點 P 爲中心且和掃描線方向的矩形，如圖 4-9。

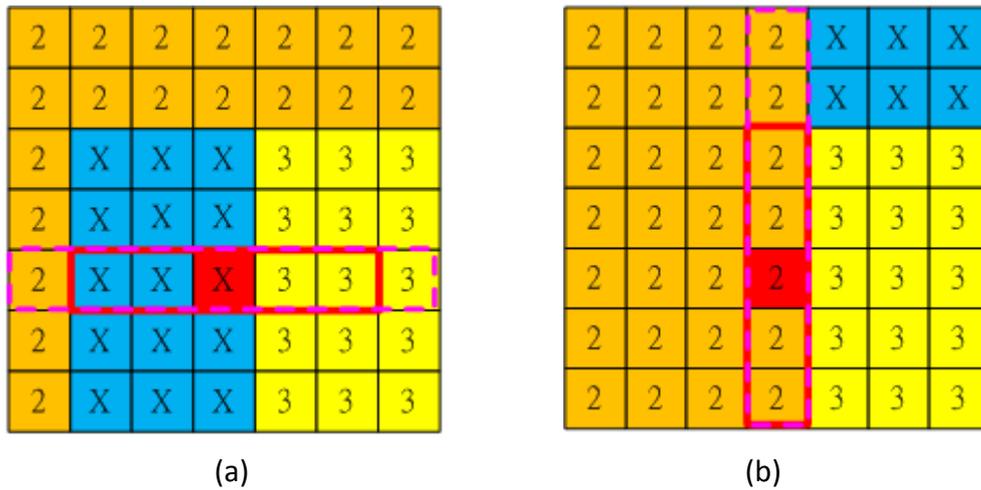


圖 4-9 視窗和掃描線方向相同的情形，點 P 為圖中紅色的區塊，矩形紅框為視窗，紫色虛線標記為掃描線位置：(a)掃描線方向為水平 (b)掃描線方向為垂直

(3) 視窗和掃描線垂直

若視窗方向和掃描線垂直，則視窗視一個以點 P 為中心和掃描線方向垂直的矩形，如圖 4-10。

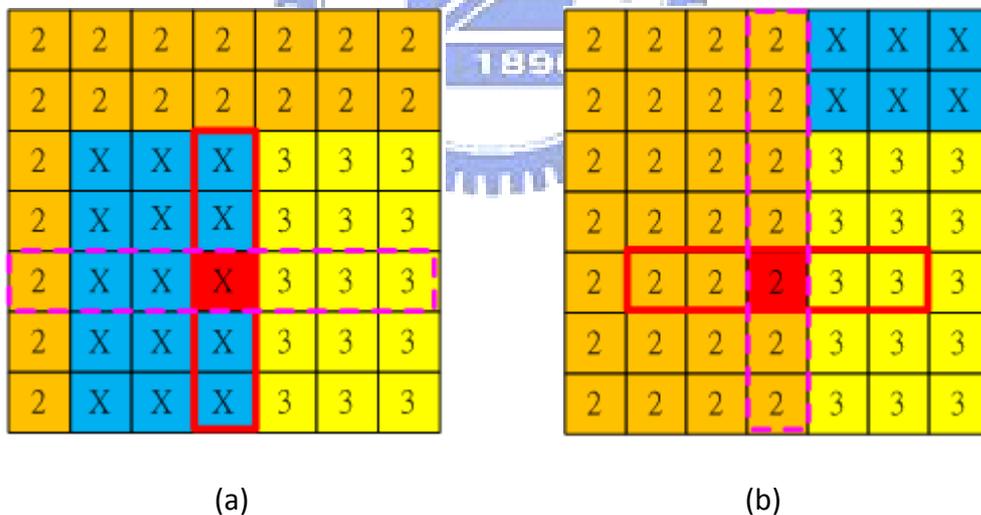
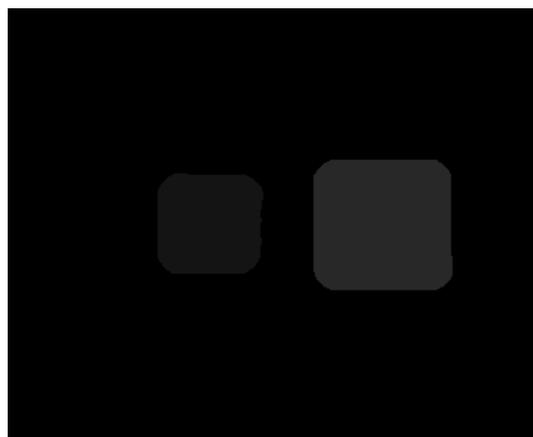


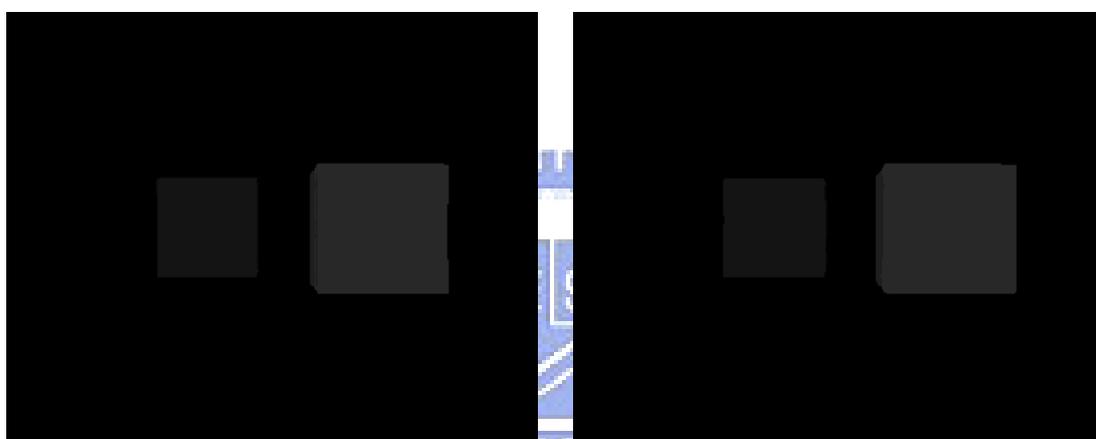
圖 4-10 視窗和掃描線方向垂直的情形，紅點表示 P 所在的位置，方形紅框為視窗，紫色虛線標記為掃描線位置：(a)掃描線方向為水平 (b)掃描線方向為垂直

圖 4-11 是分別以三種視窗對圖 4-6 進行投票的結果。可以明顯看出第一種視窗會造成方形角落的部份變成圓弧狀，而第二種和第三種方式在這個範例中僅

在物體邊緣上出現細微的差異。



(a)



(b)

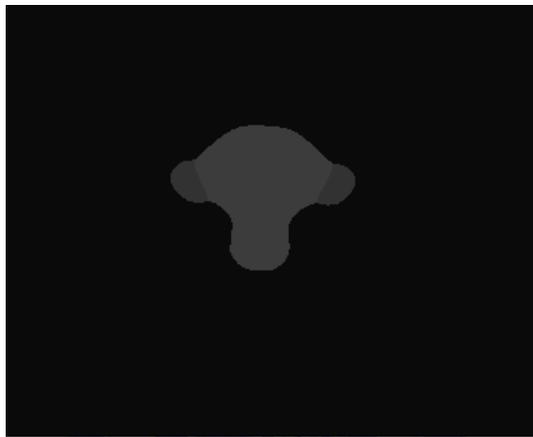
(c)

圖 4-11 三種視窗對圖 4-6 進行投票的結果：(a)視窗和掃描線無關 (b)視窗和掃描線方向相同 (c)視窗和掃描線方向垂直

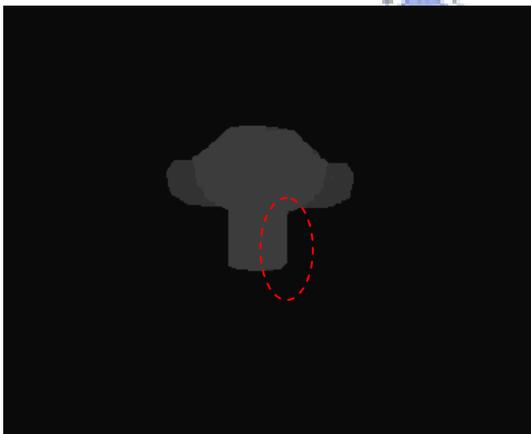
爲了更明顯區別第二種和第三種視窗的差異，我們以圖 4-12 來進行測試，其結果如圖 4-13 所示。可以看出第一種視窗仍然會造成邊緣變得圓滑而外觀嚴重變形。而第二種視窗則會造成物體在視差圖中的面積比原本影像中大，導致部份背景的視差值被物體的視差值覆蓋，例如圖 4-13(b)中所標記的區域。使用第三種視窗的結果在物體邊緣上雖然也有許些變形，但和第二種視窗相比明顯變形較接近原本影像。因此，我們選擇視窗方向和掃描線方向垂直作爲我們的投票機制。



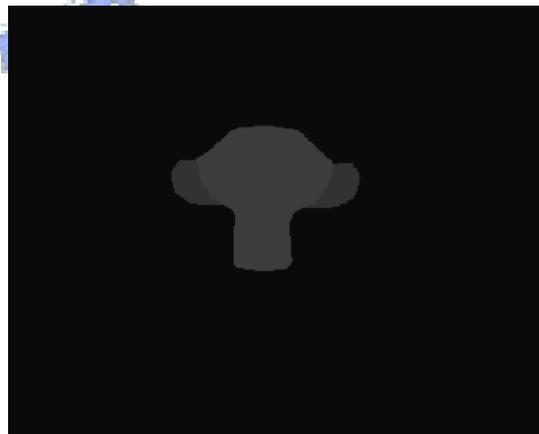
圖 4-12



(a)



(b)



(c)

圖 4-13 三種視窗對圖 4-12 進行投票的結果：(a)視窗和掃描線無關 (b)視窗和掃描線方向相同 (c)視窗和掃描線方向垂直

圖 4-14 是採用掃描線方向垂直的投票機制下比較視窗大小對視差圖的影響，其視窗長度分別為 5、10、15、以及 20 個 pixel。觀察圖中可以發現，當視窗太小時，參考的資料不足，容易造成物體邊緣區域視差值不一致，如圖 4-14(a)中標記部份；而當視窗太大時，容易將其他物體的視差值一起列入參考，導致物體邊緣變形或兩物體間區域視差值錯誤，如圖 4-14(d)。經實驗測試後，視窗長度以 10 個 pixel 最佳。

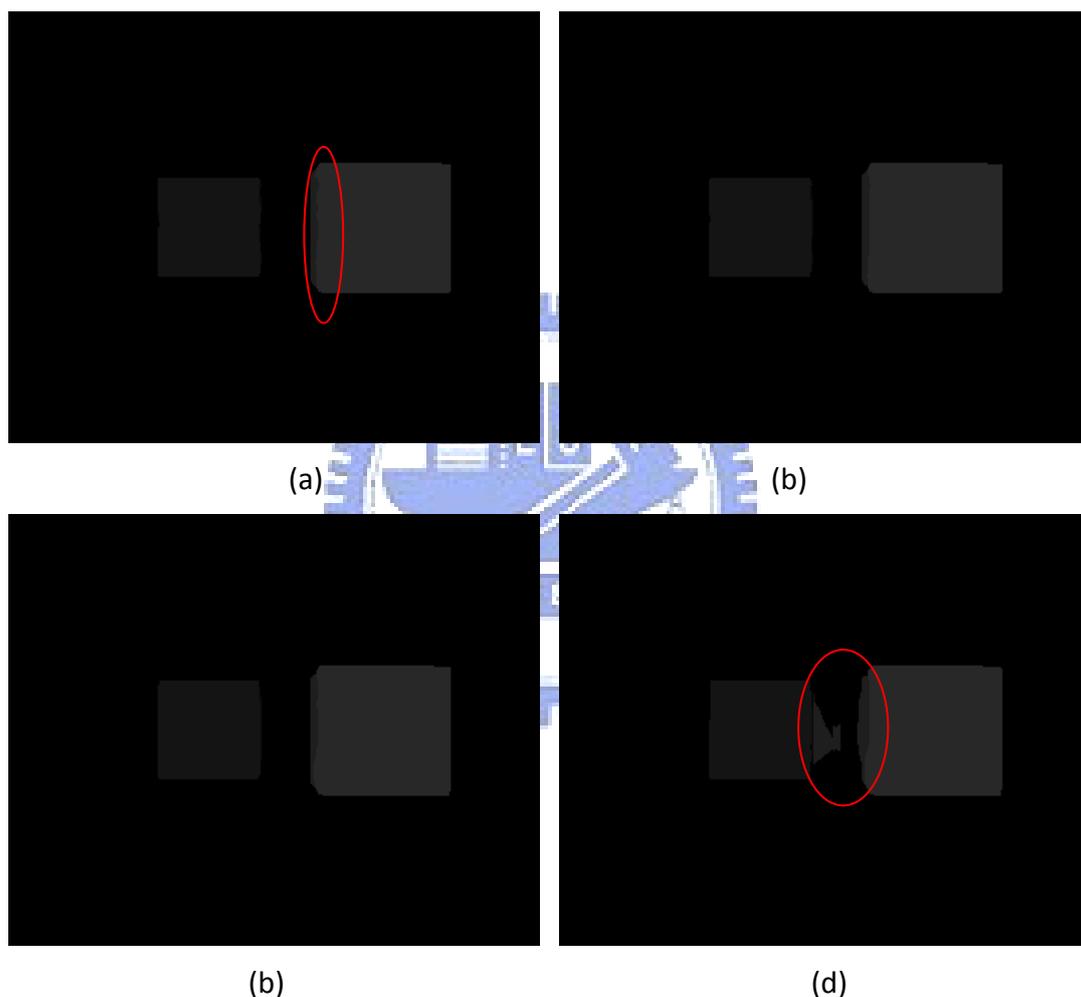


圖 4-14 不同視窗大小對投票結果的影響情形(a)視窗長度為 5 個 pixel (b)視窗長度為 10 個 pixel (c)視窗長度為 15 個 pixel (d)視窗長度為 20 個 pixel

此外，若出現票數相同的情形時，我們選擇較大視差值的作為最後結果。這是因為視差值越大，代表物體越接近攝影機，其資訊相對較為重要。所以在無法

判定視差值時，選擇較大的視差值較能減少遺失重要資訊的風險。

4.3 棋盤式視差估測

在上一節中，我們參考鄰近四個方向的影像來得到更精確的視差圖，但該方法的時間複雜度為傳統視差估測方法的四倍，再加上一個二維攝影機陣列中，攝影機數量動輒超過數十台。這會使得計算整個系統的影像視差所需要的時間會非常驚人。因此，在本節中我們根據二維攝影機陣列特有的排列結構提出一個降低整體系統計算量的視差估測方法。

根據視差的定義，兩張相鄰影像的視差圖間會存在著一個平移的關係。例如圖 4-15 分別表示兩張水平方向上相鄰影像之視差圖， D_{mapL} 代表左邊影像的視差圖， D_{mapR} 為右邊影像的視差圖。其中圓形區域的視差值為 8，背景的視差值為 2。

比較兩張視差圖，可以發現 D_{mapR} 相當於將 D_{mapL} 中的圓形區域向左平移 8 個 pixel 的位置；換言之，若我們已知 D_{mapL} ，則只要將 D_{mapL} 中各 pixel 向左平移和其視差大小相同的距離就可以得到 D_{mapR} 。

對於降低計算二維攝影機陣列系統整體視差值所需的時間，上述的特性提供了一個可行的方法：我們只需要計算系統中部份攝影機擷取影像的視差值，其餘的攝影機擷取影像則利用已計算出來的視差圖來推測。在本論文中，我們稱呼以完整方式計算得到的視差圖為主視差圖(primary disparity map)，與其對應的攝影機稱為主攝影機(primary camera)；而利用主視差圖推測得到的視差圖稱為副視差圖(secondary disparity map)，與其對應的攝影機稱為副攝影機(secondary camera)。



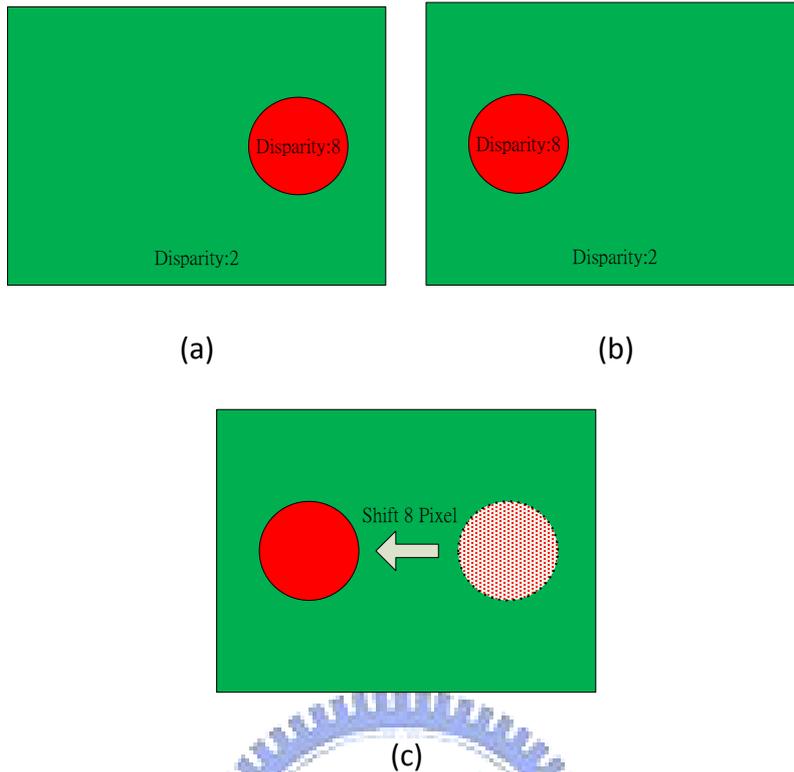


圖 4-15 視差圖之間的平移關係：(a)左邊影像的視差圖 D_{mapL} (b)右邊影像的視差圖 D_{mapR} (c)圓形區域在兩者間的平移關係

使用主視差圖來推測鄰近影像的視差時，需要遵守以下兩個原則：(1)在攝影機陣列中，攝影機之間的距離不可以太大。當攝影機彼此相距太遠，會造成同一件物體在不同攝影機成像中外觀不同，這會導致推測出來的鄰視差圖結果和實際視差圖不合。例如圖 4-16，因為兩台攝影機距離太遠，我們可以看出在左邊攝影機的成像平面上的綠色的區域明顯比在右邊攝影機成像平面上的綠色區域來的大。(2)若發生有兩個以上的視差值在平移後對應到同一位置，則我們保留最大的值作為該點的視差值。因為視差值越大代表物體越靠近攝影機，所以會遮蔽視差值較小(亦即距離相機較遠)的後景，如圖 4-17。

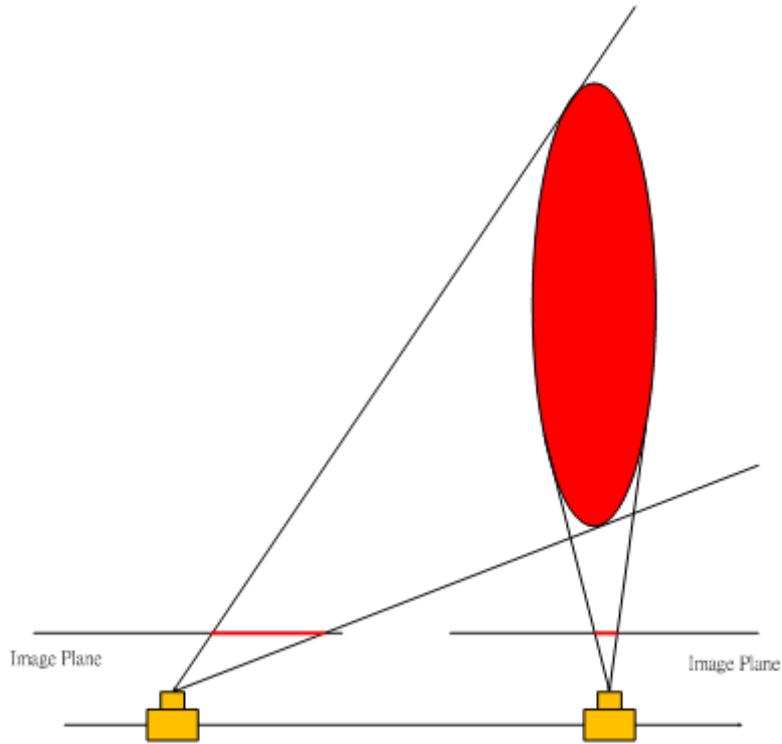


圖 4-16 因相鄰相機距離太遠導致同一物體在兩個成像平面上外觀不同

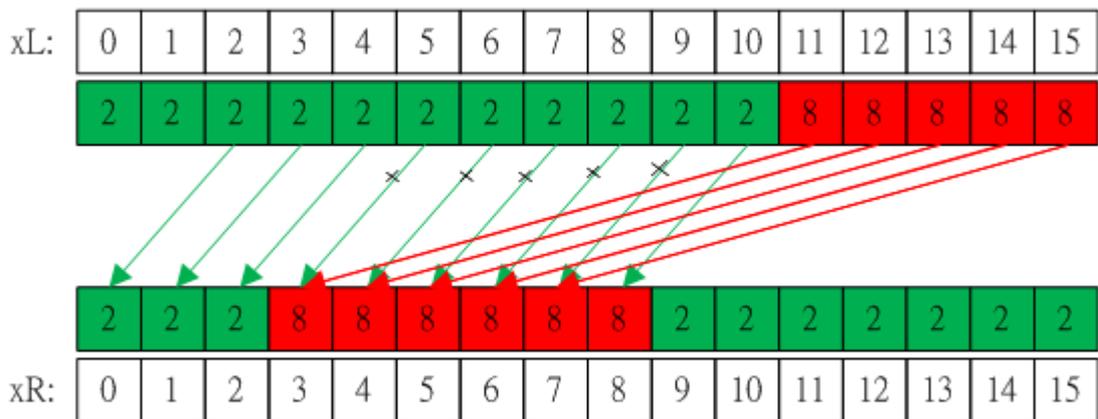


圖 4-17 兩種以上的視差值對應到同一點時的處理情形

圖 4-18 是以左邊影像作為主視差圖 D_{mapL} 來推測右邊視差圖 D_{mapR} 的例子。

我們只要將 D_{mapL} 上的每一點，向左平移和它的視差值大小相同的 pixel 數，即可

得到 D_{mapR} 。觀察 D_{mapR} ，可以發現在視差值高的區域右邊會緊連著一塊無法定義

視差值的區域(藍色部份)。從平移的角度來看，在將 D_{mapL} 上各點向左平移至 D_{mapR} 上的對應位置時，視差值大的區域移動較多，視差值小的區域移動較少，導致原本相鄰的兩個不同視差值區域在平移後被分開，如圖 4-19。而從遮蔽效應的角度來看，相當於該區域所需資訊在 D_{mapL} 中被前景所覆蓋而無法取得，而出現類似 occlusion 的情形。



圖 4-18 以左邊影像的視差圖來推測右邊影像的視差圖：(a)左邊的視差圖 D_{mapL}
 (b)參考 D_{mapL} 而推測出來的右邊視差圖 D_{mapR}

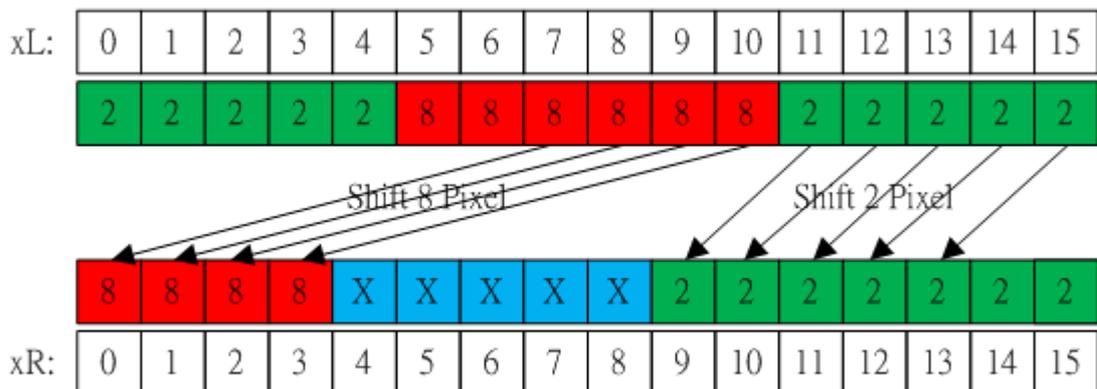


圖 4-19 因不同視差值移動距離不同而導致右邊影像上部份區域無法定義視差

爲了填補無視差值定義的區域以及增加結果的準確度，我們使用四台主攝影機擷取影像的視差圖來合成一張副視差圖，其基本流程如圖 4-20。圖 4-21 爲攝影機彼此的位置關，其中 O_{scC} 是副攝影機， O_{pcR} 、 O_{pcL} 、 O_{pcD} 以及 O_{pcU} 分別爲副攝影機周圍的四台主攝影機，且假設擷取影像分別爲圖 4-22 中的 I_c 、 I_r 、 I_l 、 I_d 以及 I_u ，並以完整估測方式得到相對應的四張主視差圖，如圖 4-23。將主視差圖上的各點平移和其視差值相同的 pixel 數後，我們可以得到四張由不同方向的主視差圖對 I_c 所推測的視差結果，如圖 4-24。我們可以明顯看出任一張視差圖上無法定義視差值的區域(圖中標記處)皆可使用另外三張視差圖資訊來填補。然而，在這四張視差圖上仍然有細部的差異，尤其在物體邊緣的區域。所以對於 I_c 上各點，我們也是以投票的方式，選擇相同位置上出現次數最多視差值作爲該點視差，其結果如圖 4-25(b)。

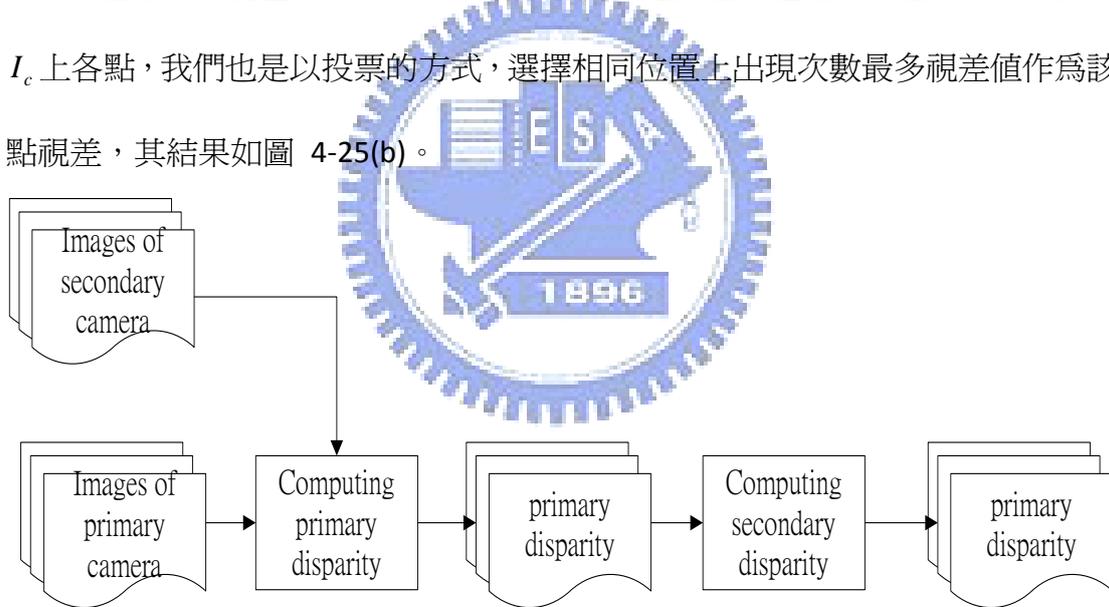


圖 4-20 使用主視差圖合成副視差圖的流程圖

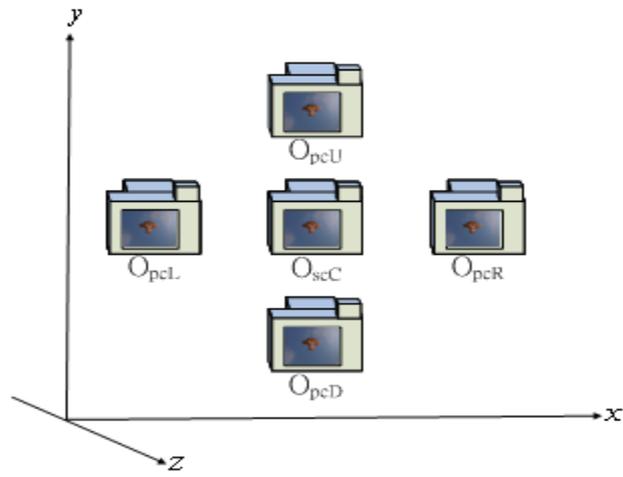


圖 4-21 鄰攝影機 O_{nc} 和周圍主攝影機的位置關係

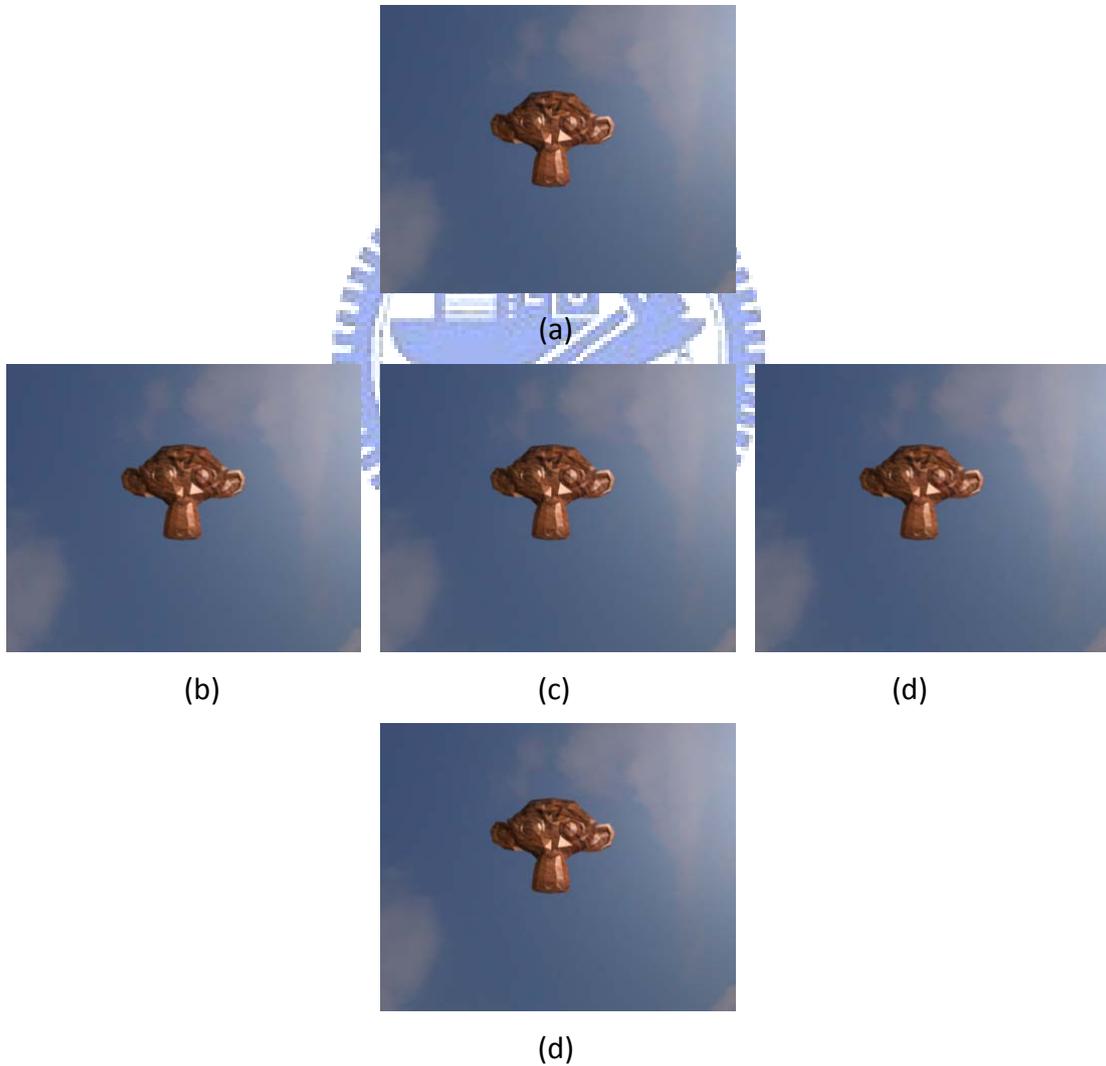


圖 4-22 攝影機 O_c 和鄰近攝影機的擷取影像：(a) O_{mcU} 的擷取影像 (b) O_{mcL} 的擷取影像 (c) O_{nc} 的擷取影像 (d) O_{mcR} 的擷取影像 (e) O_{mcD} 的擷取影像

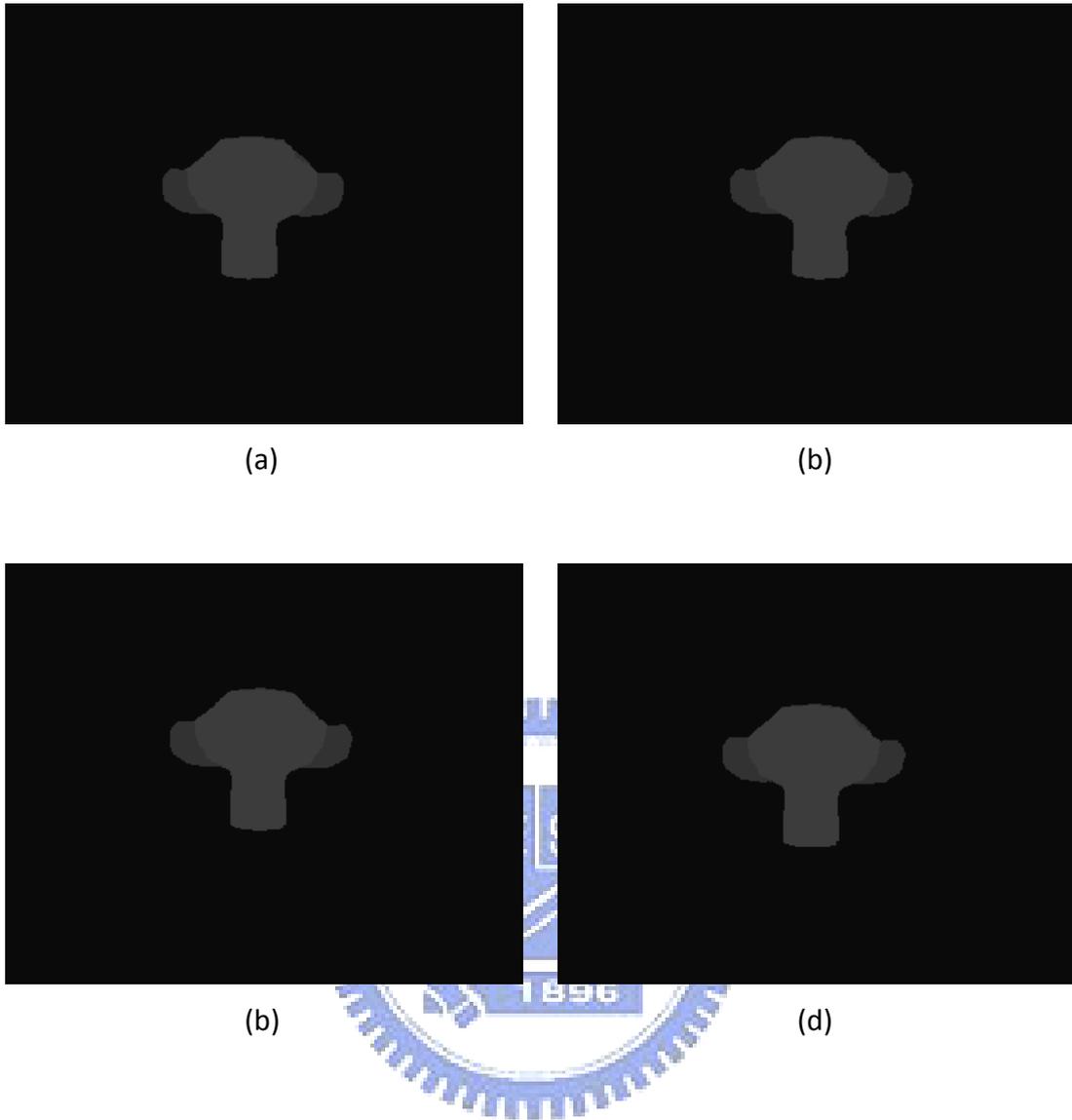


圖 4-23 各主攝影機擷取影像的視差圖：

(a) O_{mcR} 擷取影像之視差圖 (b) O_{mcL} 擷取影像之視差圖

(c) O_{mcD} 擷取影像之視差圖 (d) O_{mcU} 擷取影像之視差圖

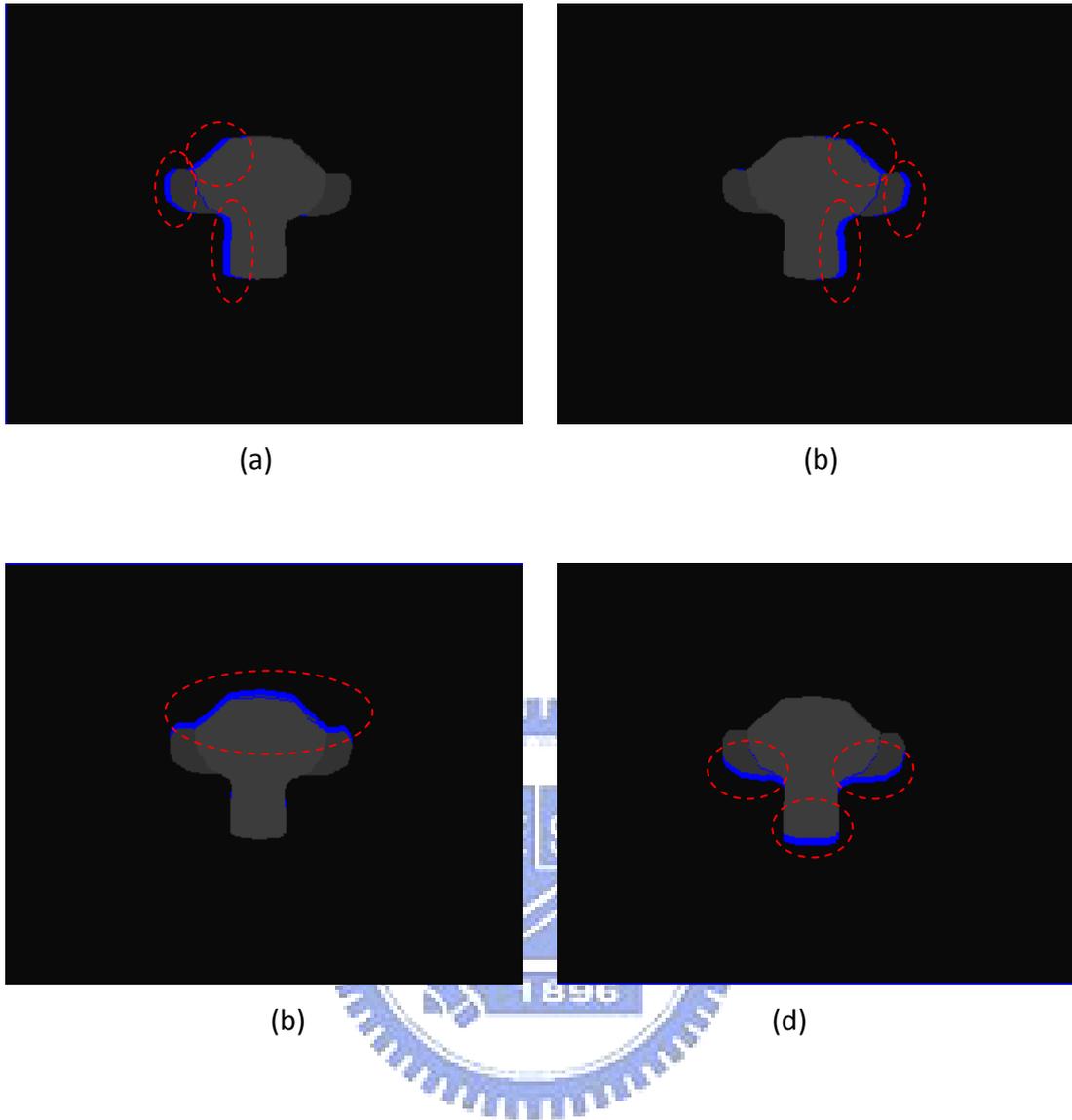


圖 4-24 使用不同位置的主視差圖推測影像 I_c 的視差結果：

(a)使用左邊主視差圖推測的結果 (b)使用右邊主視差圖推測的結果

(c)使用下方主視差圖推測的結果 (d)使用上方主視差圖推測的結果

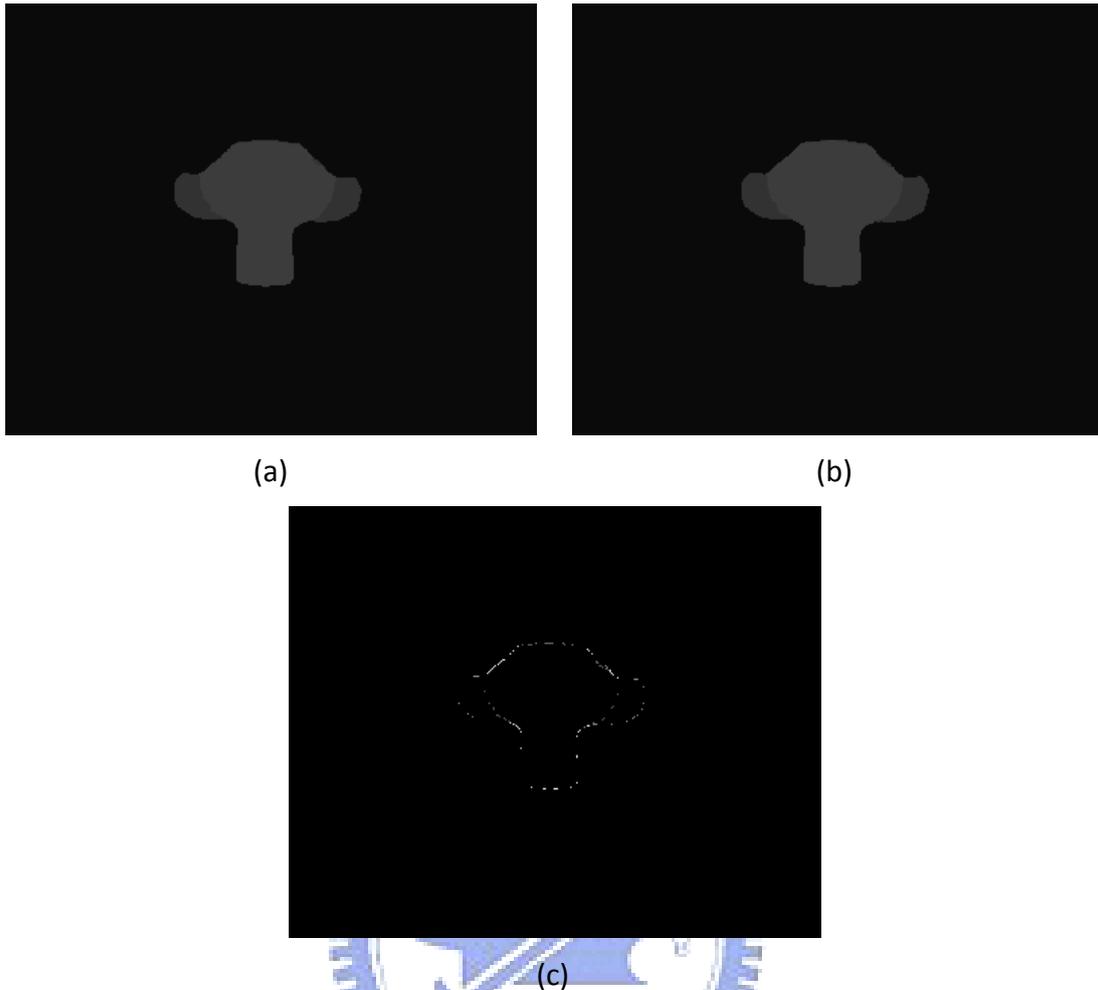


圖 4-25 以圖 4-24 進行投票來判斷影像 I_c 的視差圖：(a)使用完整估測方式取得的 I_c 之視差圖 (b)經投票後得到的 I_c 之視差圖 (c)兩種方法在結果上的差異

圖 4-26 是將圖 4-21 的攝影機關係延伸至整個二維攝影機陣列，其中紅色攝影機為主攝影機，藍色攝影機為副攝影機。我們可以發現在攝影機陣列中主攝影機和副攝影機彼此互相包圍交錯，有如一張棋盤圖結構。所以在本論文中，我們稱此種估測整體二維攝影機陣列的影像視差值方式為棋盤式視差估測(check-board disparity estimation)。在棋盤式視差估測中，僅需要完整計算系統中一半的攝影機擷取影像(主攝影機擷取影像)的視差，而另一半的攝影機擷取影像(副攝影機擷取影像)的視差則使用周遭的主視差圖來合成，因此我們在估測整個二維相機陣列的視差值上可以降低一半的計算量。由於副攝影機視差圖是由四張主攝影機視

差圖經投票處理而合成的，其合成效果通常不亞於直接估測所有攝影機視差圖。

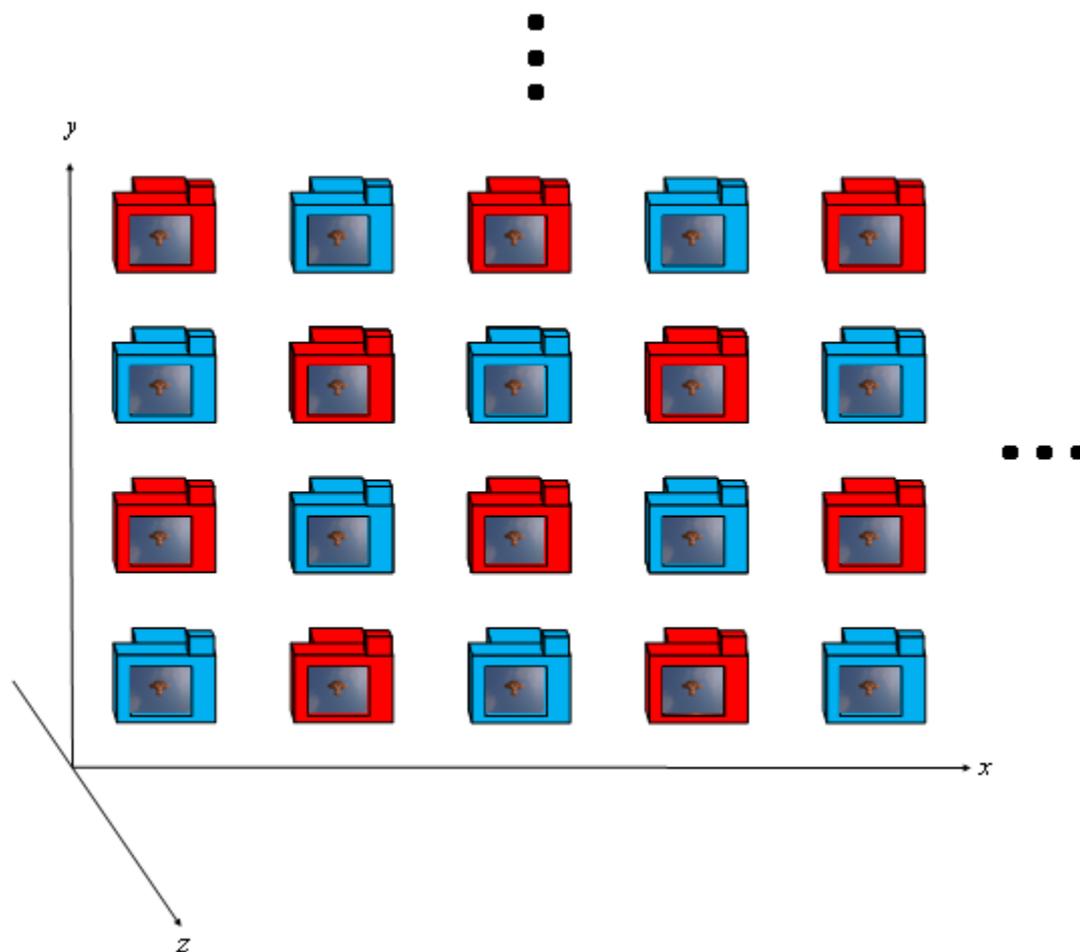


圖 4-26 主攝影機和副攝影機在二維攝影機陣列上的位置關係

4.4 影像合成

4.4.1 一維攝影機陣列影像合成

在合成虛擬視點影像時，我們使用視差資訊來判斷參考資訊所在位置。雖然我們使用整體最佳化以及投票機制來得到較精確的視差圖，但是和實際情形仍然會有誤差，尤其是在物體邊緣的部份。這可能導致合成出來的結果錯誤。此外，因遮蔽效應而在兩張影像之間無法對應的區域也會導致合成效果不佳。以下我們

按處理步驟來討論在合成水平方向上兩台攝影機間的虛擬視點影像時會遇到的情形。

1. 左邊影像視差值對應到右邊影像視差值較小的位置，如圖 4-27(a)：

這種情況為低視差值覆蓋了高視差值，相當於離相機較遠的物體擋住了離相機較近的物體，這顯然是不合常理的。因此我們認為這是視差值錯誤所造成的，並將右邊影像較小的視差值以相對應的左邊影像視差值取代，如圖 4-27(b)。

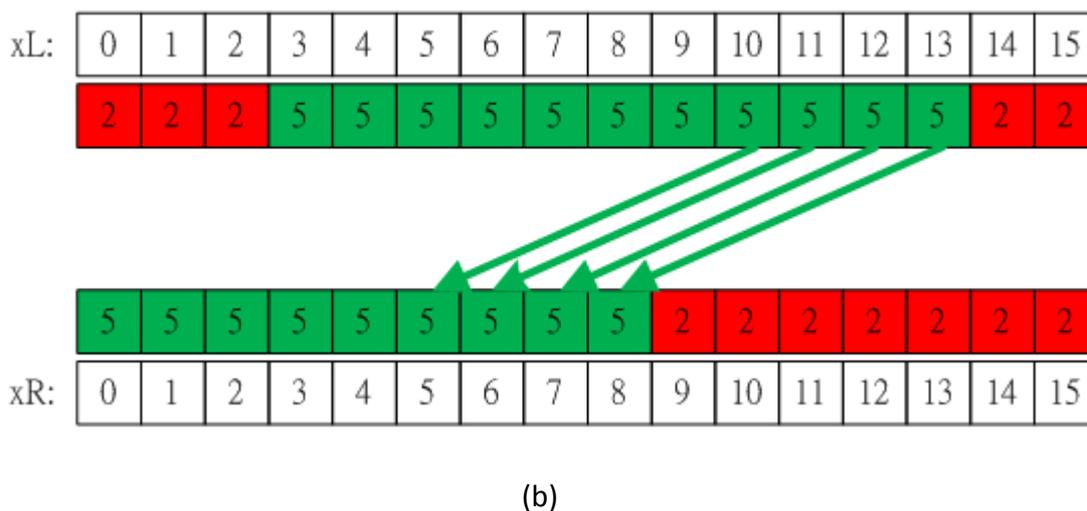
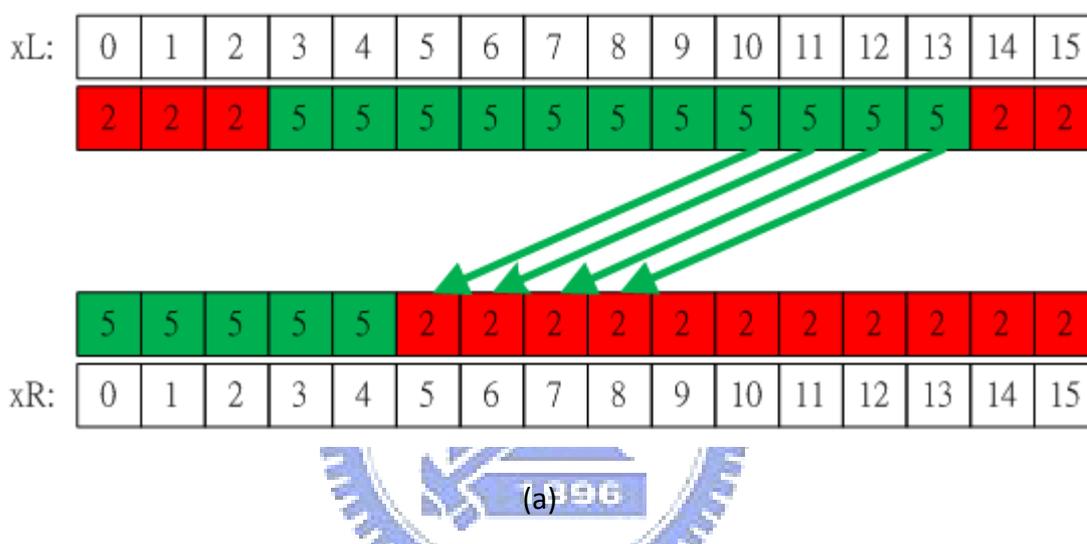


圖 4-27 (a)左邊影像視差值大於對應的右邊影像視差值 (b)視差值修正後的結果

2. 左邊影像視差值對應到右邊影像視差值相同的位置，如圖 4-28(a)：

此為合成影像時最常出現的情形，表示該組對應點在左右兩邊的影像中皆未受到遮蔽效應的影響。若虛擬視點成像平面上的 pixel 正好處在一組視差值相同的對應連線和成像平面的交錯處，則該 pixel 可直接參考該組對應點以線性內插的方式來合成影像，如圖 4-28(b)。此步驟完成後結果如圖 4-29，可以看出合成影像上大部分的區域都已經成型了。

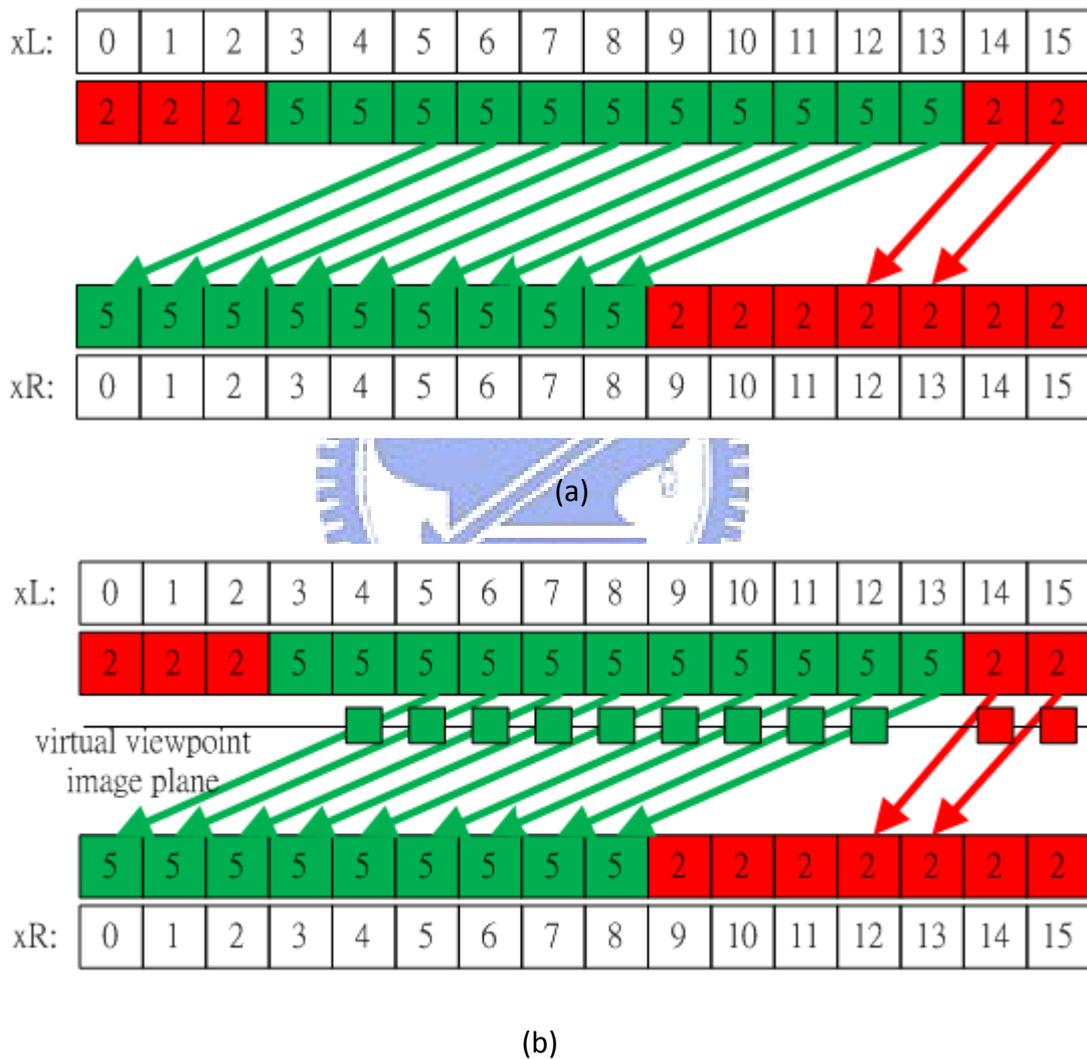


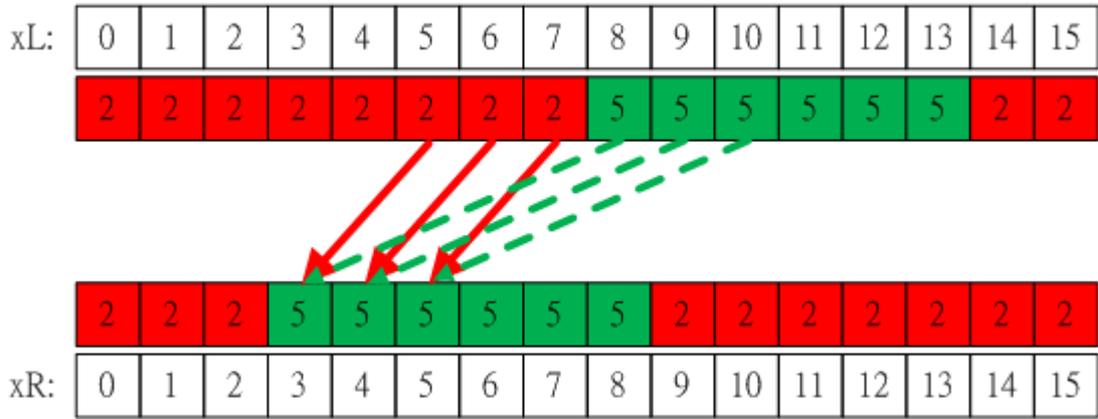
圖 4-28 (a)對應位置的視差值相同的情形 (b)虛擬視點平面的合成



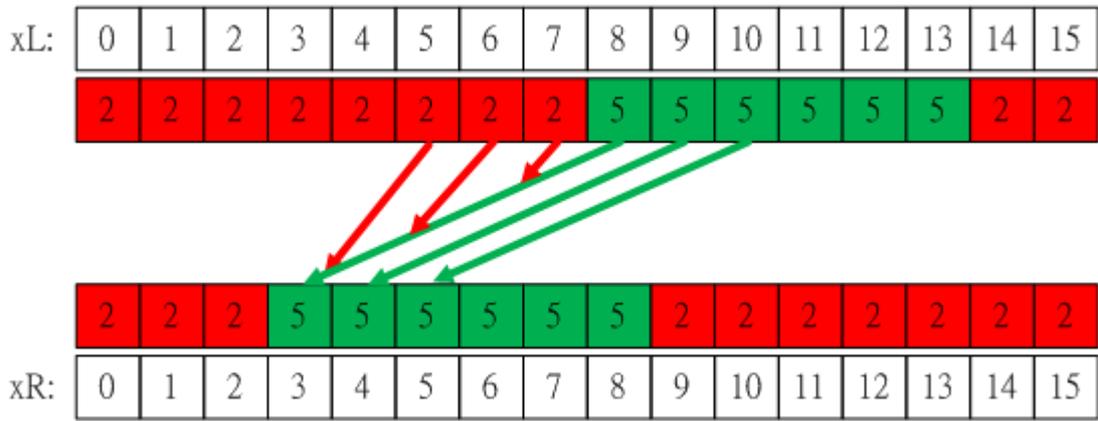
圖 4-29 步驟 2 合成後結果

3. 左邊影像視差值對應到右邊影像視差值較大的位置，如圖 4-30(a)：

此情況表示該點在右邊影像上被遮蔽。由圖 4-30(b)可以看出實際對應到該位置的是左邊影像上視差較高的區域。遇到這種情形時，合成時所參考的對象會依虛擬視點所在位置而改變。以圖 4-31(a)為例，當虛擬視點位置靠近左邊攝影機時，在虛擬視點成像平面上第 6 個 pixel 是以左邊影像第 6 個 pixel 作為參考，因為虛擬視點所在位置上該點尚未被遮蔽。；而當虛擬視點較靠近右邊攝影機時，如圖 4-31(b)，則是以左邊影像第 10 個 pixel 及右邊影像第 5 個 pixel 為參考。此步驟完成後的結果如圖 4-32 中標記區域。



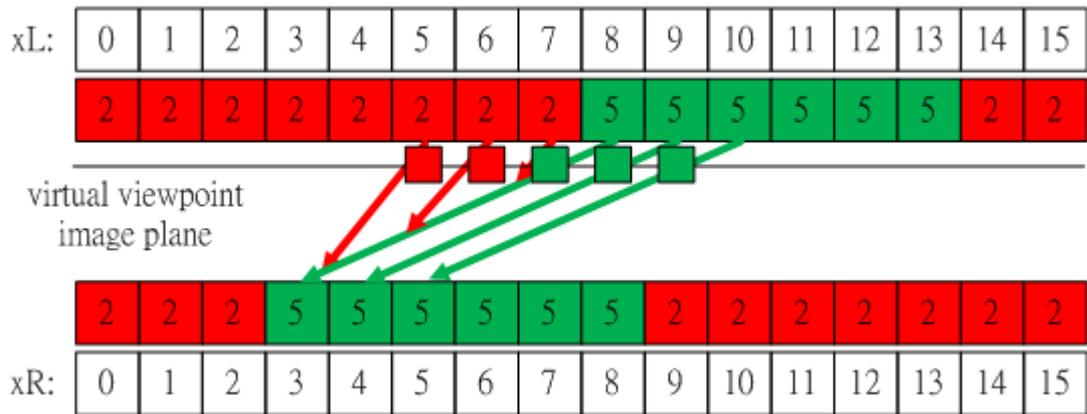
(a)



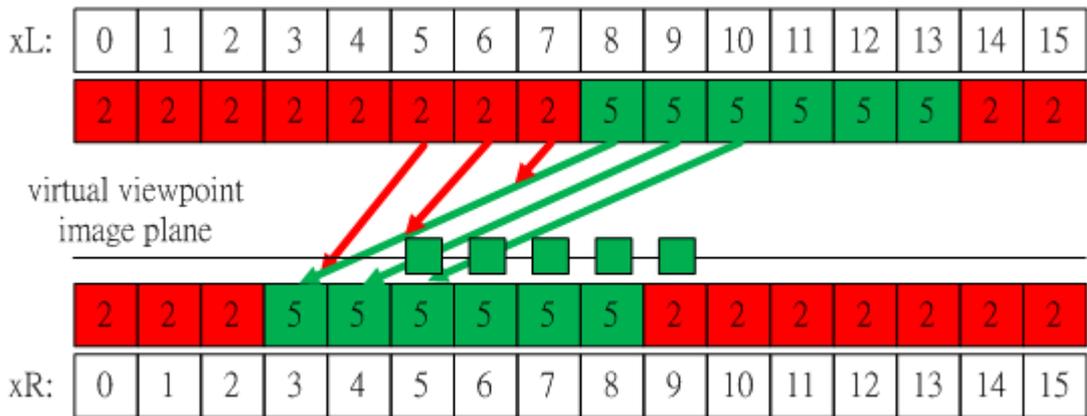
(b)

圖 4-30 (a)左邊影像視差值小於對應到的右邊影像視差值

(b) 實際的對應情形



(a)



(b)

圖 4-31 (a) 虛擬視點位置靠近左邊攝影機時的合成情形
 虛擬視點位置靠近右邊攝影機時的合成情形

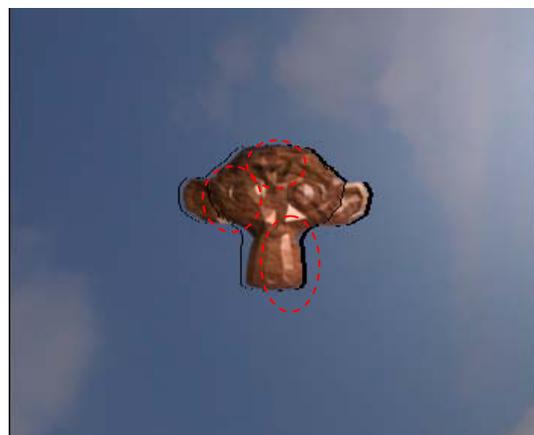


圖 4-32 步驟 3 完成後的結果

- 當以上三種情況都處理完後，我們需要檢查是否有右邊影像在左邊影像上被遮蔽的情形，如圖 4-33。其處理方式和步驟 3 一樣。此步驟完成後結果如圖 4-34 中的標記區域。

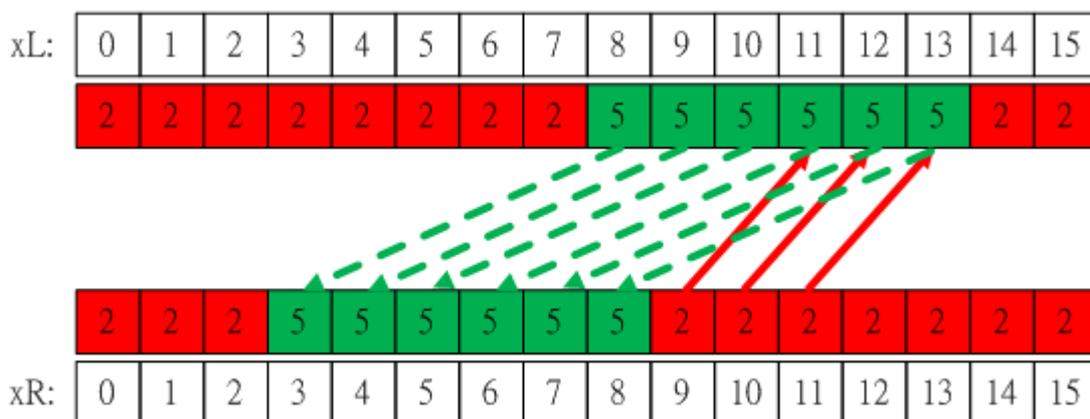


圖 4-33 右邊影像在左邊影像上被遮蔽的情形



圖 4-34 步驟 4 完成後的結果

- 當以上四個步驟處理完後，我們會發現還有部份區域因為沒有對應點連線經過而無法合成。對於這類區域，我們參考在同一條掃描線上最靠近的已知結果以線性內插的方式來合成。此步驟完成後變得到在一維攝影機陣列上的虛擬視點合成，其結果如圖 4-35。



圖 4-35 以線性內插的方式將虛擬視點成像平面上無值區域填補後的結果

4.4.2 二維攝影機陣列影像合成

對於二維攝影機陣列中的虛擬視點影像合成，我們使用兩階段的一維攝影機陣列合成方法來實現。在合成虛擬視點影像時，第一階段我們會先進行水平方向上的影像合成，再接著利用第一階段得到的影像作為參考，進行垂直方向的影像合成，流程如圖 4-36。

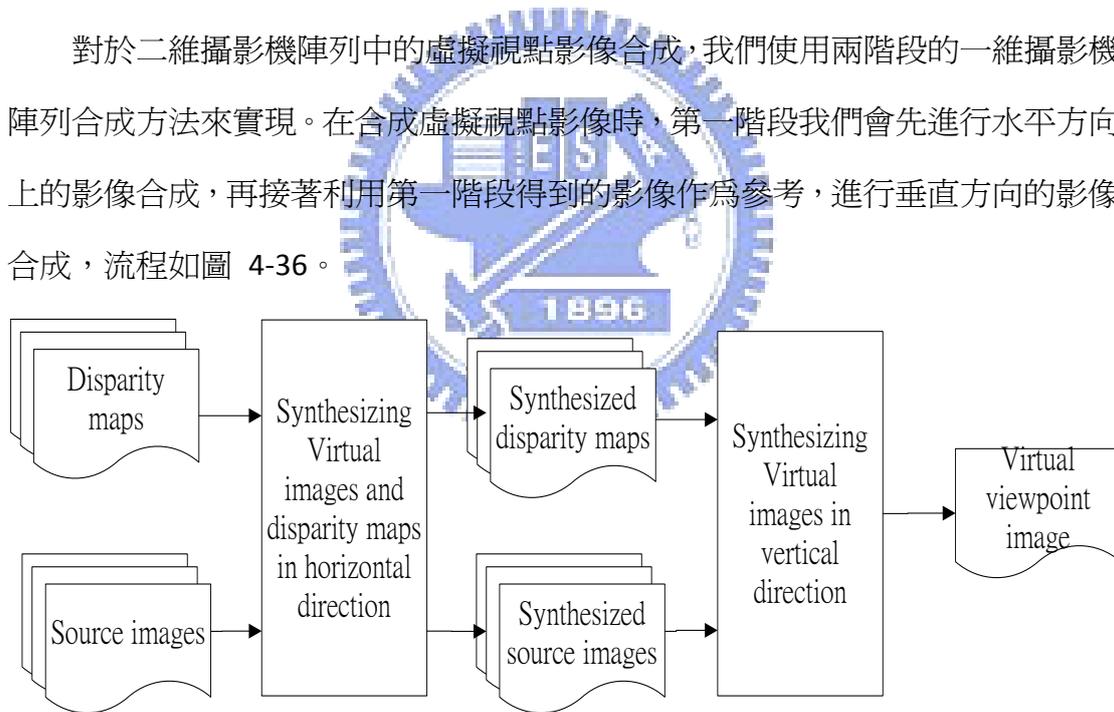


圖 4-36 在二維攝影機陣列上的虛擬視點影像合成流程圖

以圖 4-37 為例，其中 (0,0)、(0,1)、(1,0) 以及 (1,1) 為實際存在的攝影機位置，且假設攝影機擷取影像的視差圖皆為已知。若要合成出在座標 (0.3,0.4) 上所看到

的影像，我們會先合成出位於座標 $(0.3,0)$ 以及 $(0.3,1)$ 的虛擬視點影像，以及對應於該位置的視差圖。在視差圖的合成部份，我們將每張視差圖視為一張影像資訊，並帶入一維攝影機陣列上的影像合成方式來進行合成。最後，再利用 $(0.3,0)$ 和 $(0.3,1)$ 上的影像以及視差圖來合成位於 $(0.3,0.4)$ 所見到的虛擬視點影像。

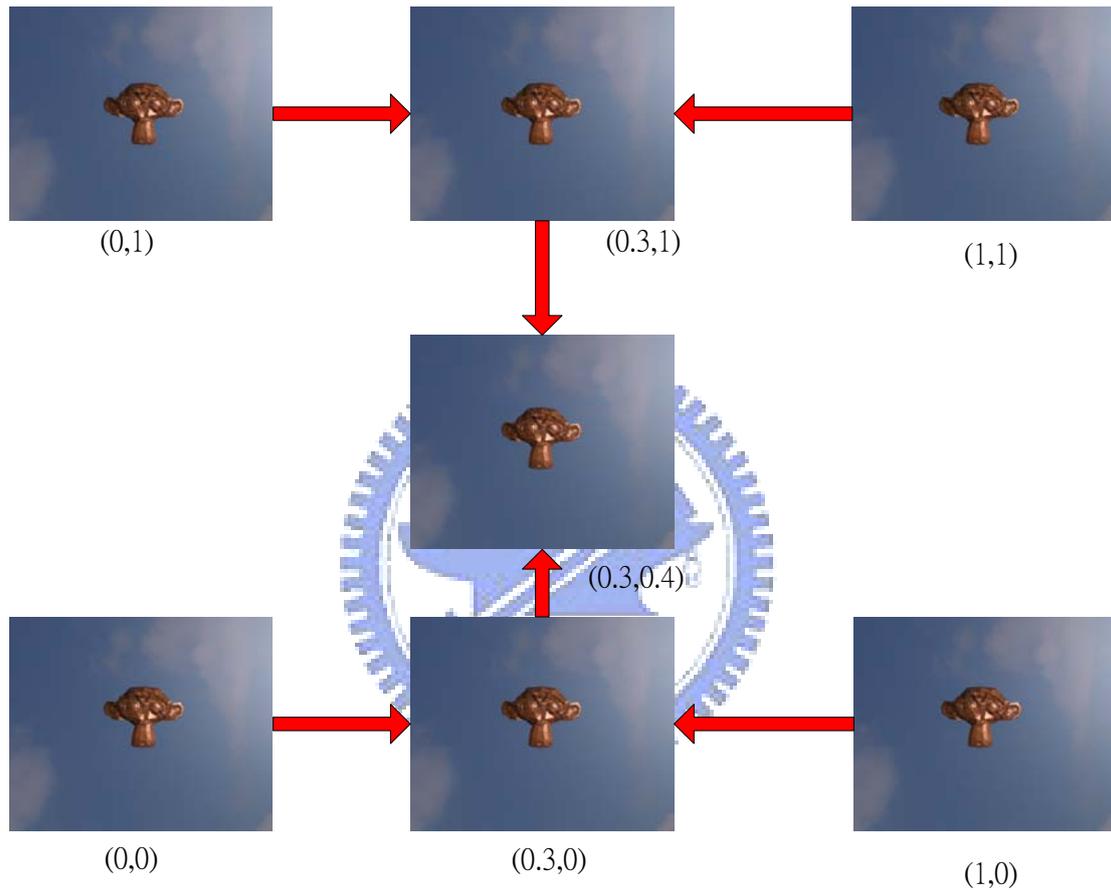


圖 4-37 兩階段的虛擬影像合成過程

在本節的最後，我們將使用二維攝影機陣列系統來合成虛擬視點影像的流程整理如圖 4-38。

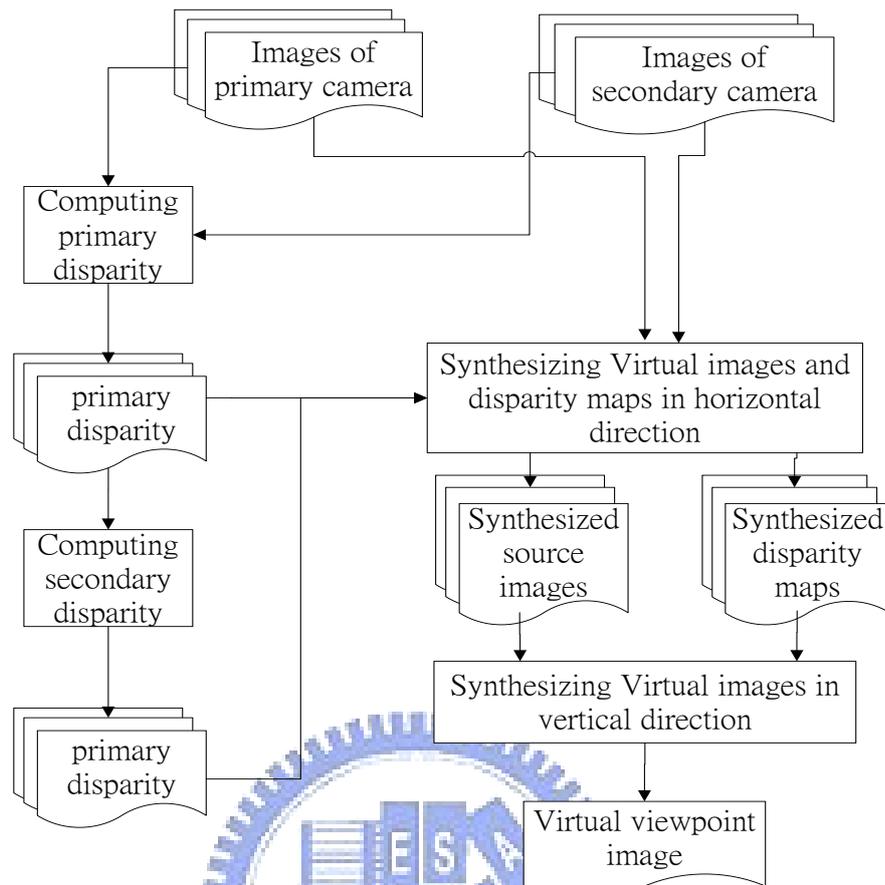


圖 4-38 在二維攝影機陣列系統上合成虛擬視點影像之流程

4.5 實驗結果

由於現今沒有一套標準的二維攝影機陣列測試影像，因此在本論文中我們使用免費的 3D 建模軟體 Blender 來產生所需的測試影像。使用軟體產生測試影像的優點是在比較合成影像時，我們可以使用理想的影像和實驗結果比較。而缺點是實際的攝影機陣列擷取影像常常會不盡理想，例如有每張影像的亮度不同、攝影機鏡頭偏移等問題。而使用軟體產生的測試影像的缺點是過於理想，導致和實際狀況有段差距。

在本節中，我們以五種不同運動行為的連續影像作為測試資料，如圖 4-39，並且比較每張單獨影像(未使用前後影像的資訊)在採用棋盤式視差估測與全部攝影機都使用完整估測的情況下，兩者影像的合成結果。

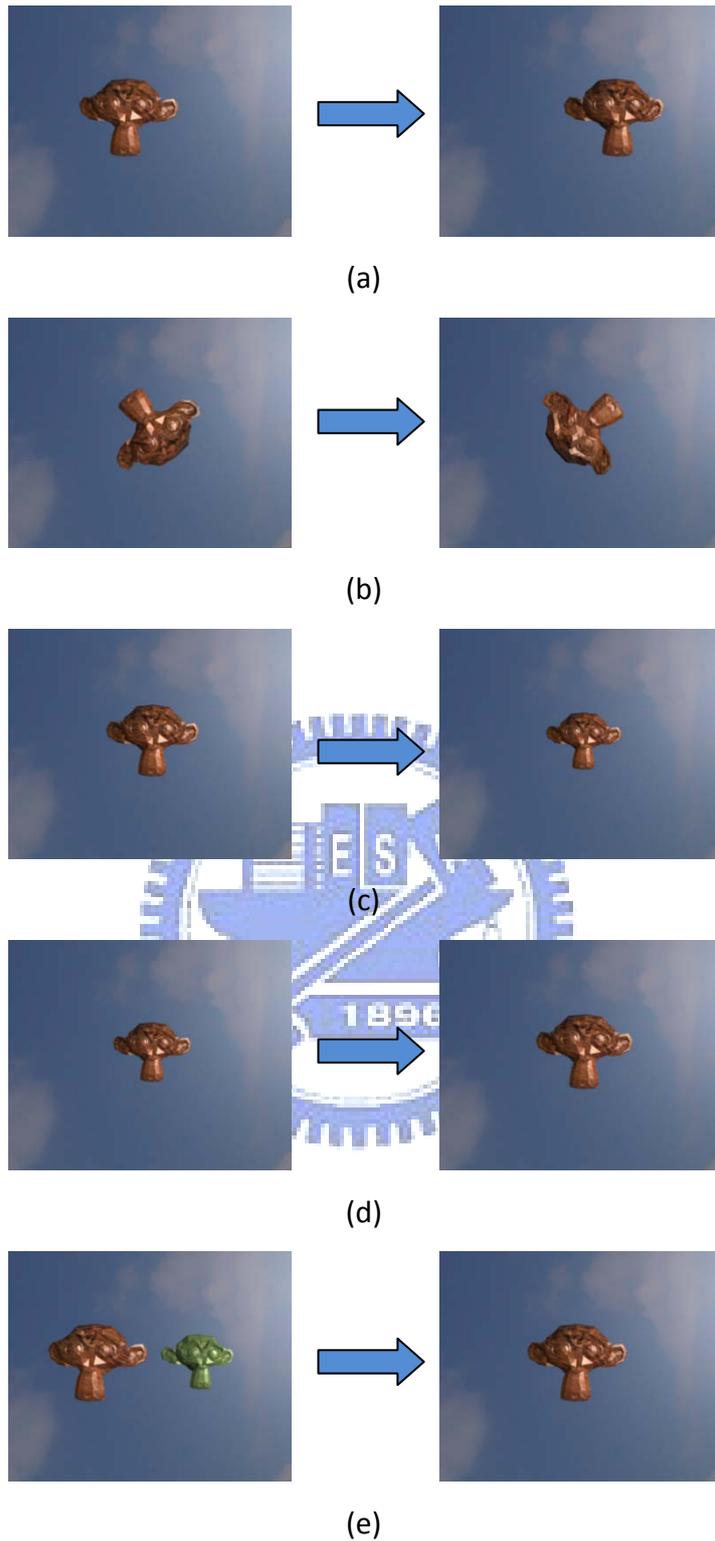


圖 4-39 各種不同的物體運動行為連續影像：

(a) Monkey-Translation(物體平移) (b) Monkey-Rotation(物體旋轉)

(c) Monkey-Leaving(物體遠離) (d) Monkey-Approaching(物體靠近)

(e) Monkey-Covering(兩物體互相遮蔽)

4.5.1 理想影像

圖 4-40 為圖 4-39 合成的虛擬視點影像之平均 PSNR，可以發現使用棋盤式視差估測的結果會比完整視差估測的結果高出約 0.005dB。而圖 4-41 到圖 4-45 是在使用兩種不同視差估測方法下，比較各張合成影像的 PSNR 大小。觀察實驗結果，可以發現兩種視差估測方式的結果在趨勢上是一致的，但是棋盤式視差估測基本上都略優於完整視差估測的結果。由於差距不到 0.01dB，在實質上並不明顯。

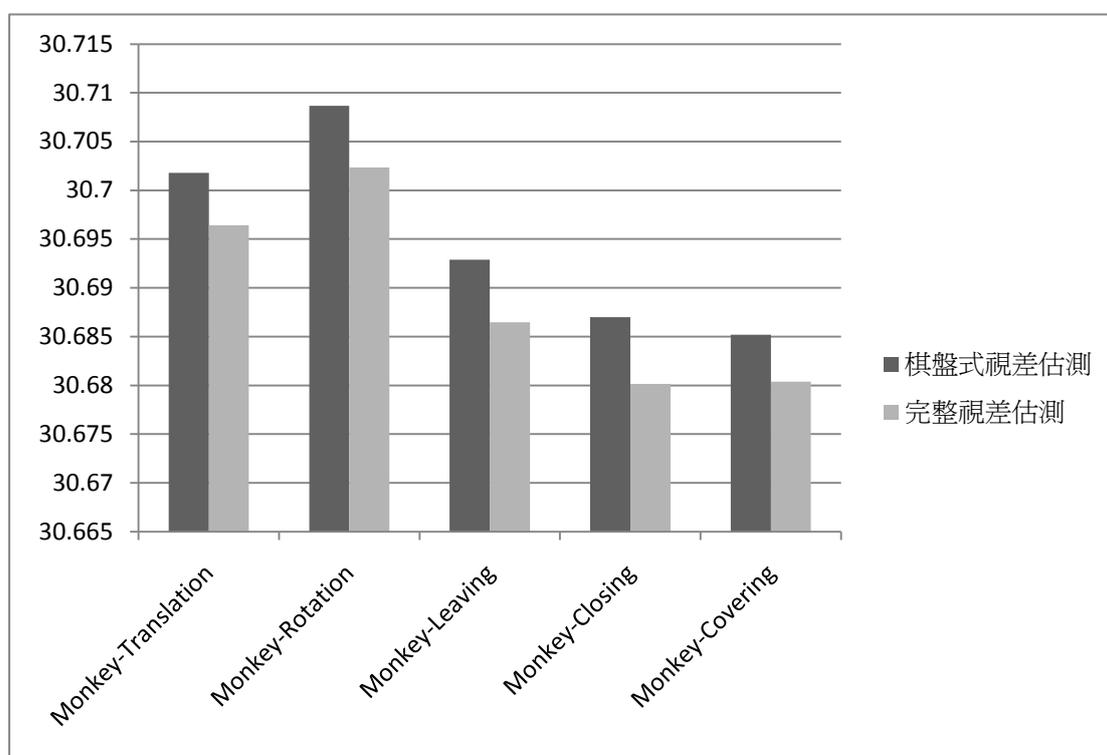


圖 4-40 兩種視差估測方式在不同測試影像上的平均 PSNR

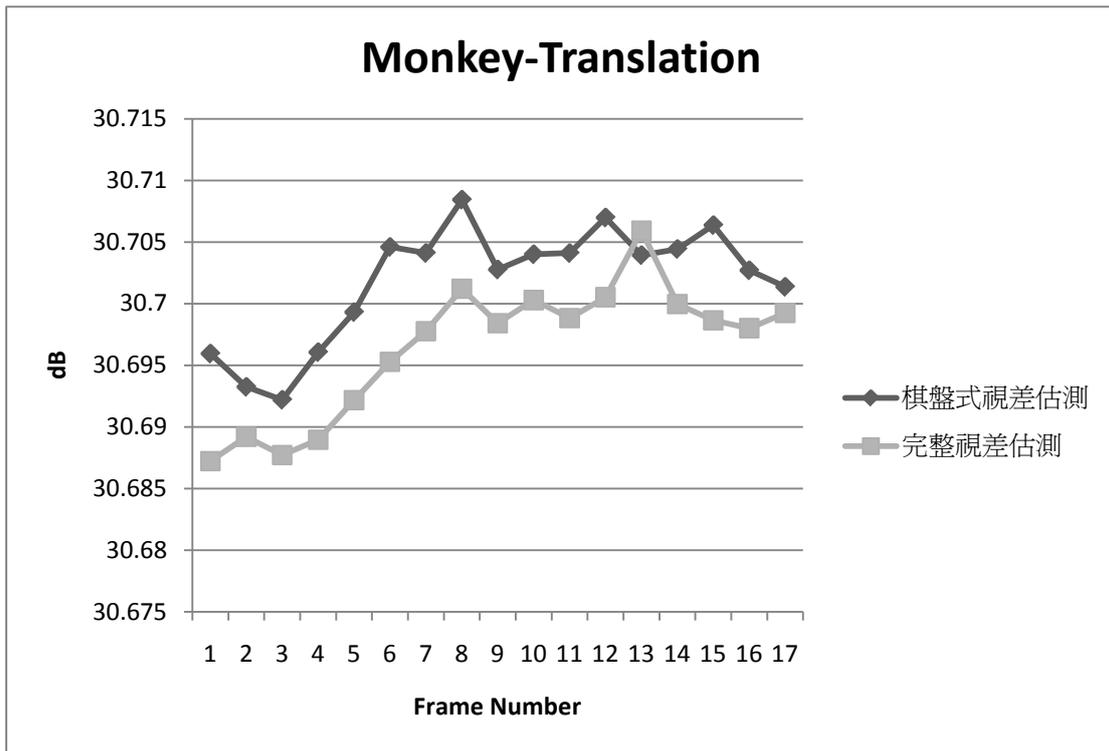


圖 4-41 兩種視差估測方式在測試影像 Monkey-Translation 上 PSNR

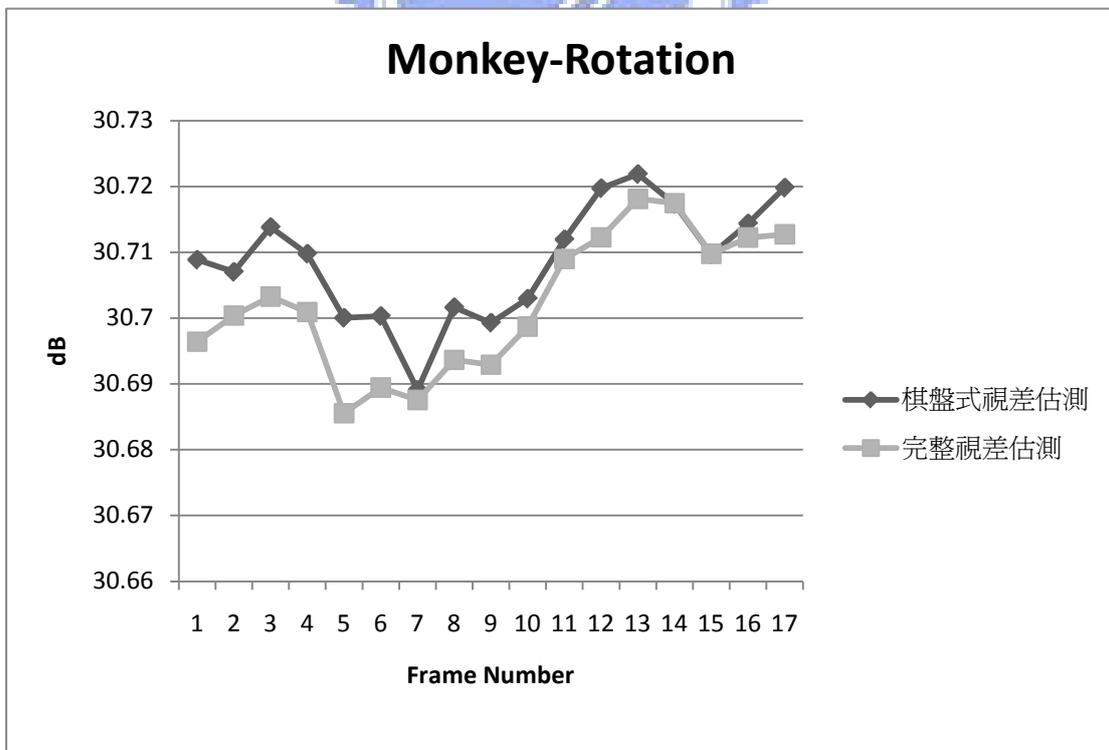


圖 4-42 兩種視差估測方式在測試影像 Monkey-Rotation 上 PSNR

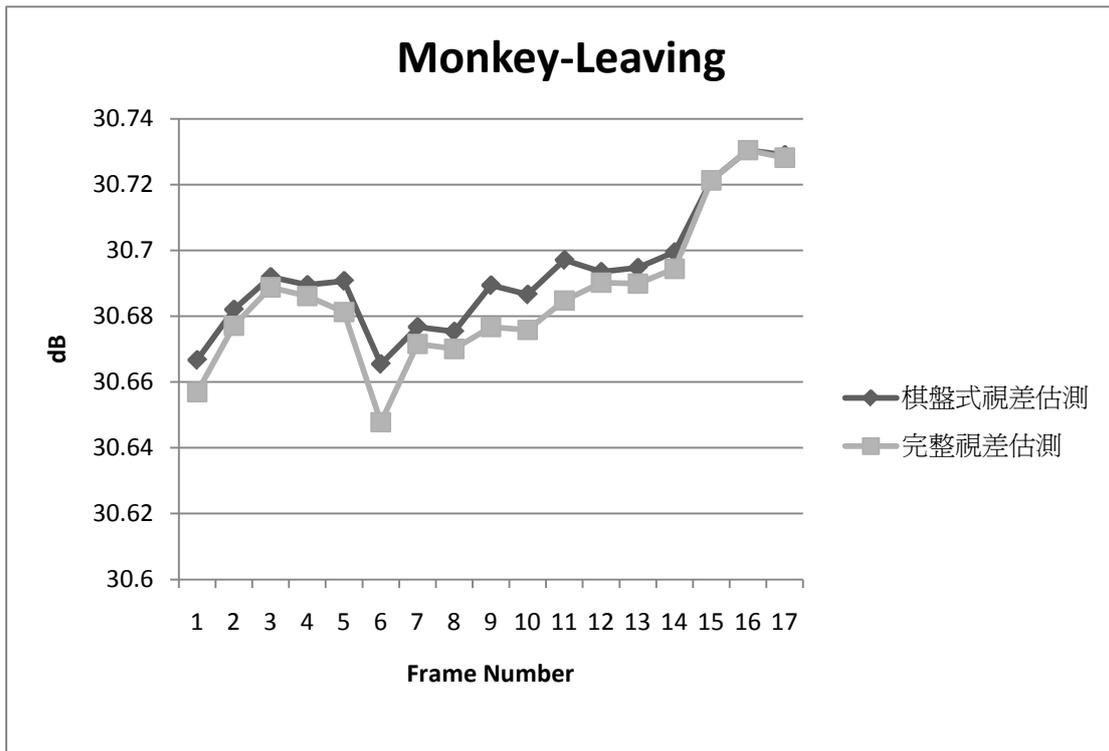


圖 4-43 兩種視差估測方式在測試影像 Monkey-Leaving 上 PSNR

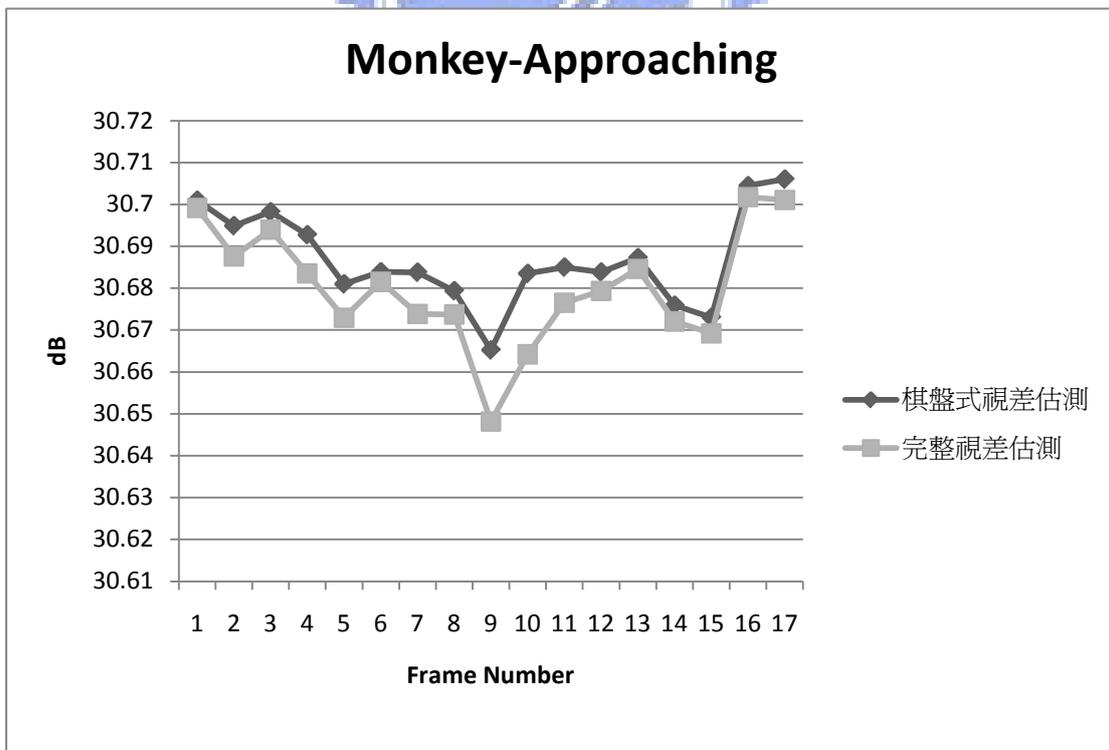


圖 4-44 兩種視差估測方式在測試影像 Monkey-Approaching 上 PSNR

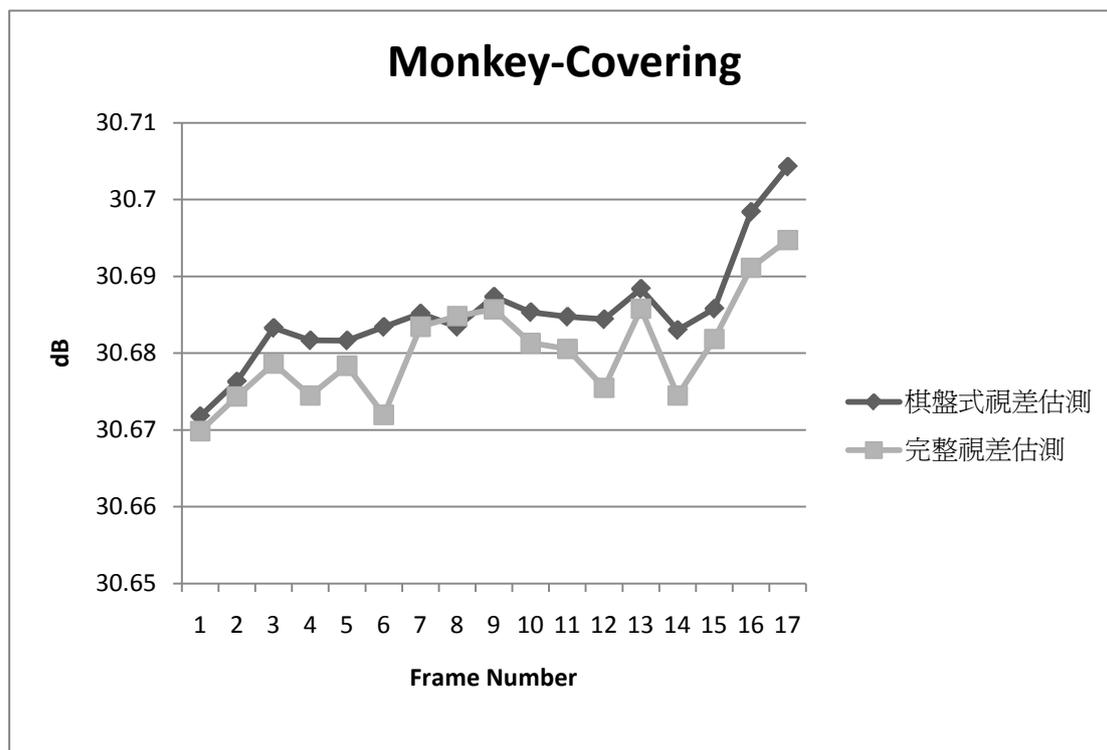


圖 4-45 兩種視差估測方式在測試影像 Monkey-Covering 上 PSNR

4.5.2 加入 AWGN 的影像

爲了讓實驗結果和現實情況更接近，我們將原始測試影像加上了 AWGN，其 variance 爲 1。影像加入 AWGN 後，在視覺上並不會有太大差異，但是各 pixel 的亮度會和原始影像略有不同，這可以用模擬出不同攝影機成像之間肉眼無法辨識的細微差異。圖 4-46 是添加 AWGN 後合成影像的平均 PSNR。和未加入 AWGN 時相比，雖然整體的 PSNR 有略爲下降，但是使用棋盤式視差估測的平均結果仍然較使用完整視差估測的結果略好。而圖 4-47 到圖 4-51 是比較各張合成影像的 PSNR，可以發現曲線趨勢基本上和未添加 AWGN 時類似，使用棋盤式視差估測得出的合成影像結果與使用完整視差估測的結果大致相同。

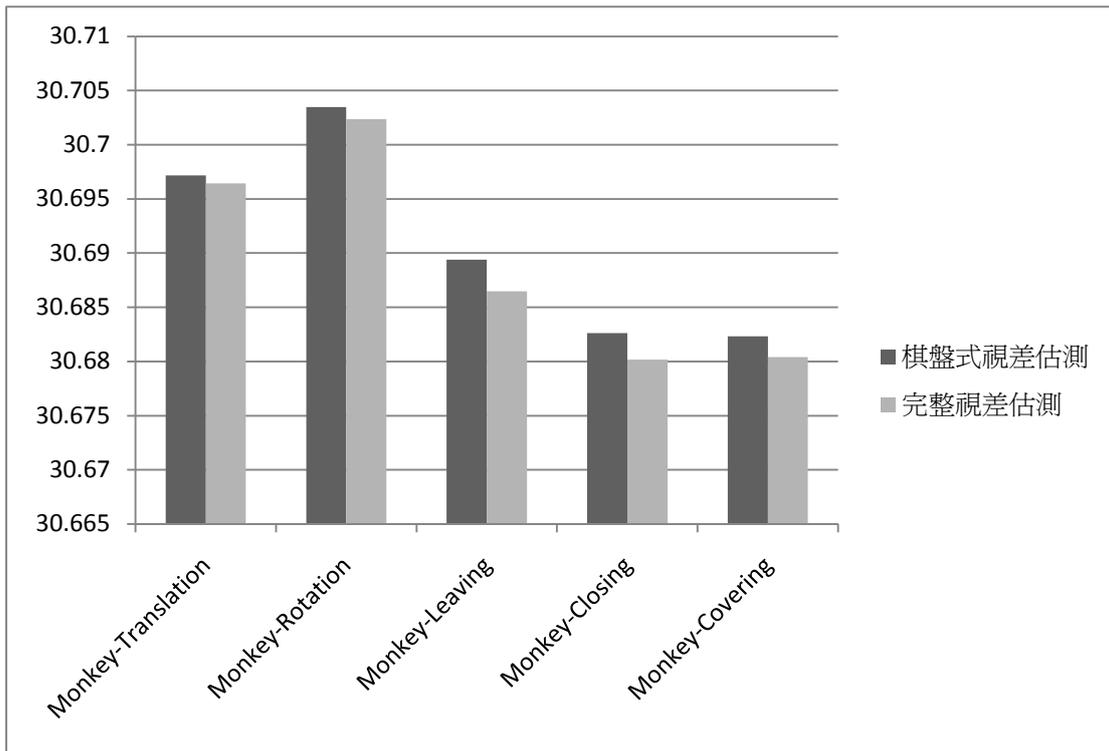


圖 4-46 兩種視差估測方式在添加 AWGN 的測試影像上的平均 PSNR

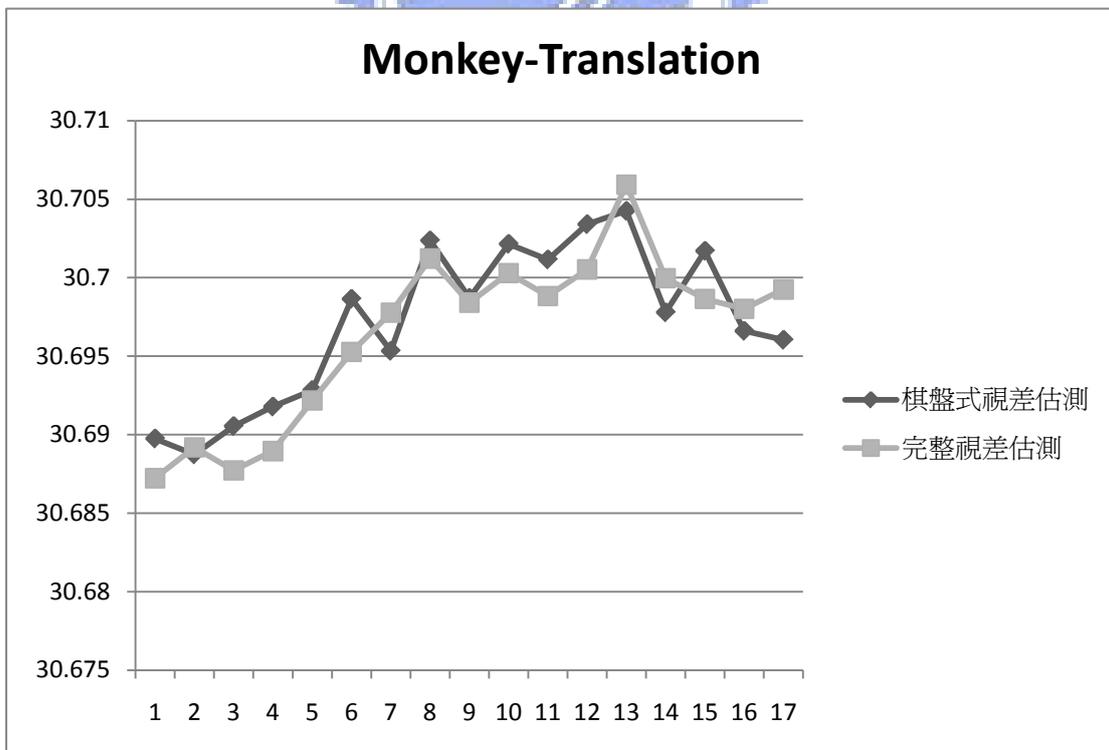


圖 4-47 兩種視差估測方式在測試影像 Monkey-Translation 添加了 AWGN 後的

PSNR

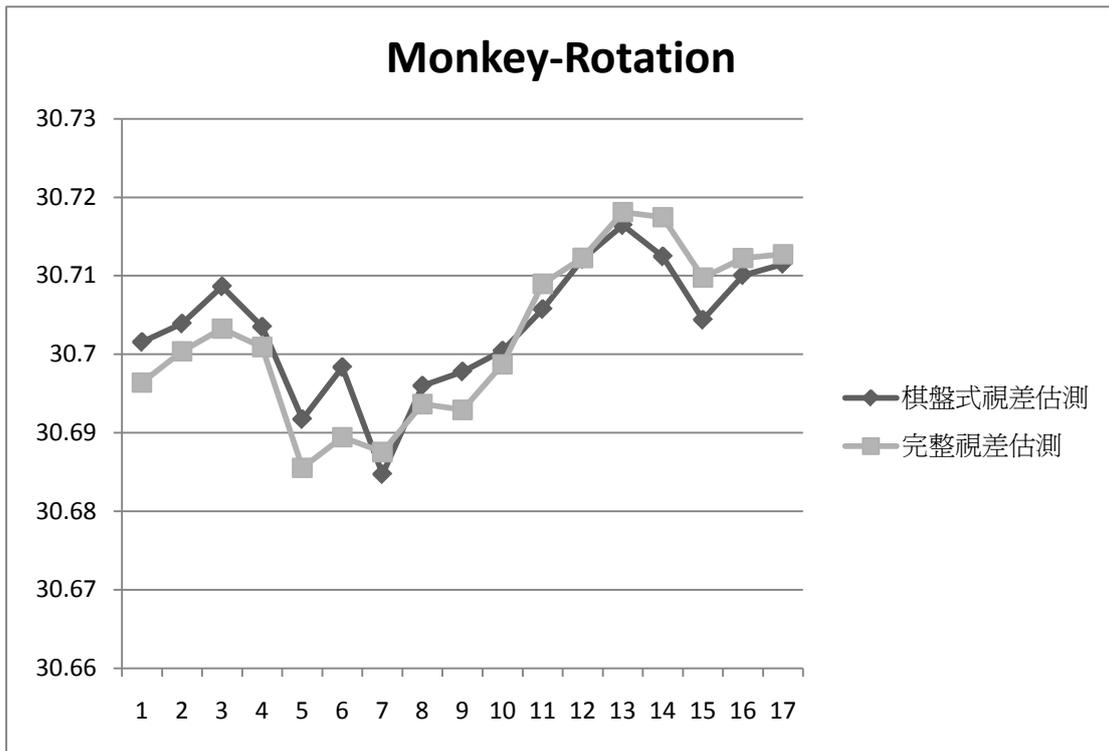


圖 4-48 兩種視差估測方式在測試影像 Monkey-Rotation 添加了 AWGN 後的 PSNR

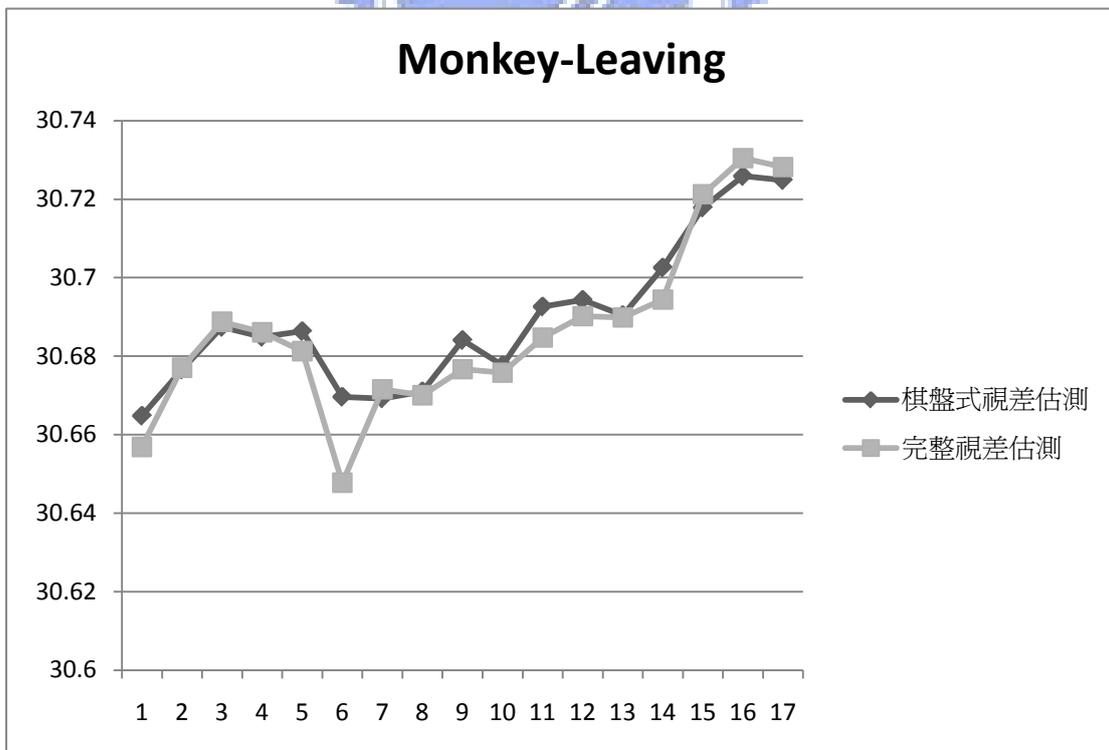


圖 4-49 兩種視差估測方式在測試影像 Monkey-Leaving 添加了 AWGN 後的 PSNR

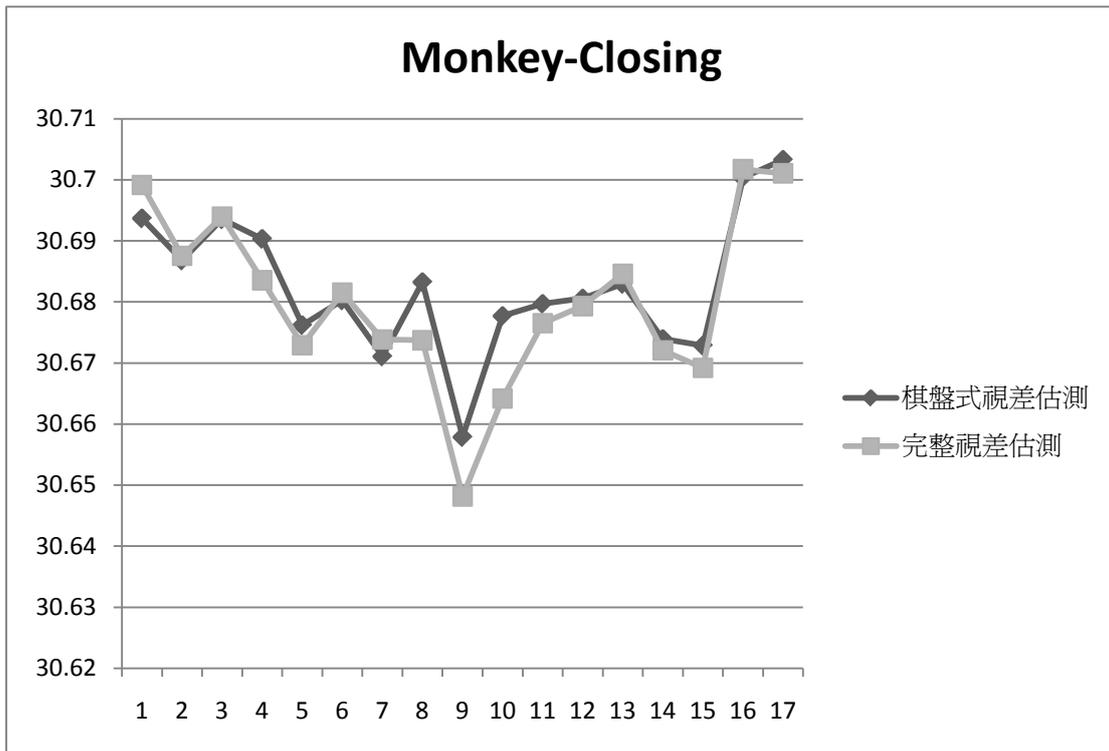


圖 4-50 兩種視差估測方式在測試影像 Monkey-Closing 添加了 AWGN 後的 PSNR

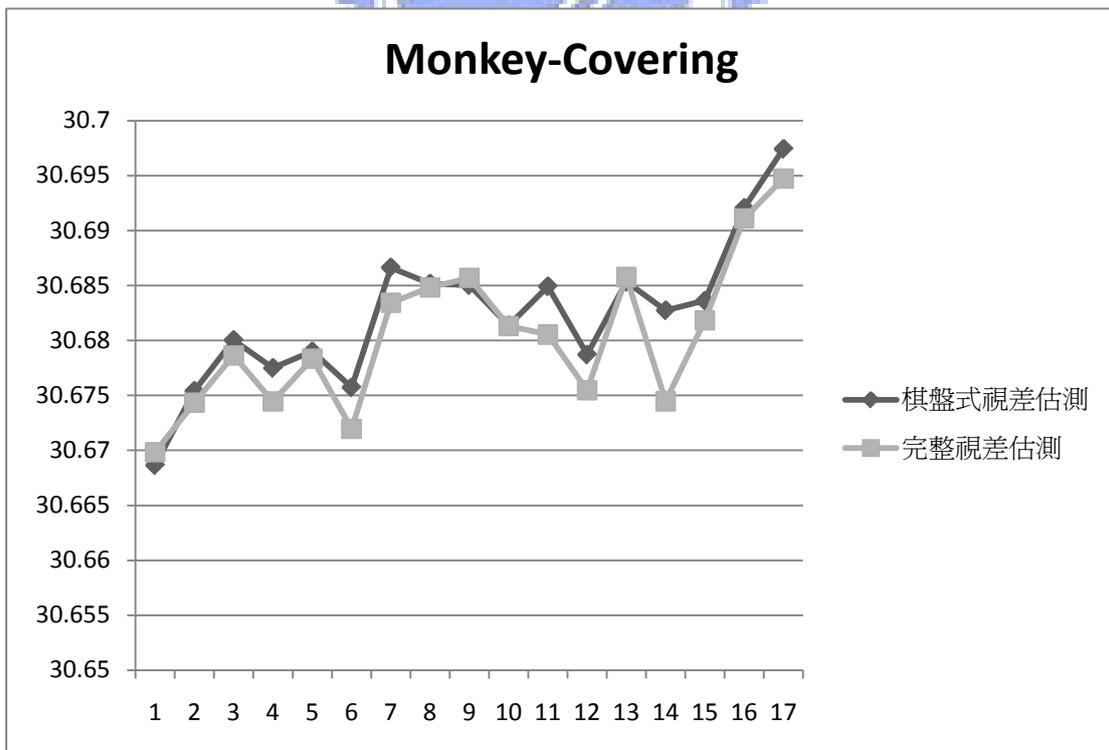


圖 4-51 兩種視差估測方式在測試影像 Monkey-Covering 添加了 AWGN 後的 PSNR

4.5.3 實驗討論

觀察以上兩個實驗結果，我們發現使用棋盤式視差估測整體結果會比使用完整視差估測略好，這和一般直觀想法剛好相反。圖 4-52(a)、(b)分別為在測試資料 Monkey-Translation 中使用完整視差估測和棋盤式視差估測得到的視差圖 D_{map} 、 $D_{MapEmbed}$ ，4-52(c)、(d)則是各自使用 D_{map} 及 $D_{MapEmbed}$ 所合成的虛擬視點影像 $I_{Virtual}$ 、 $I_{VirtualEmbed}$ 。以肉眼觀察，兩者之間並沒有明顯的差異。

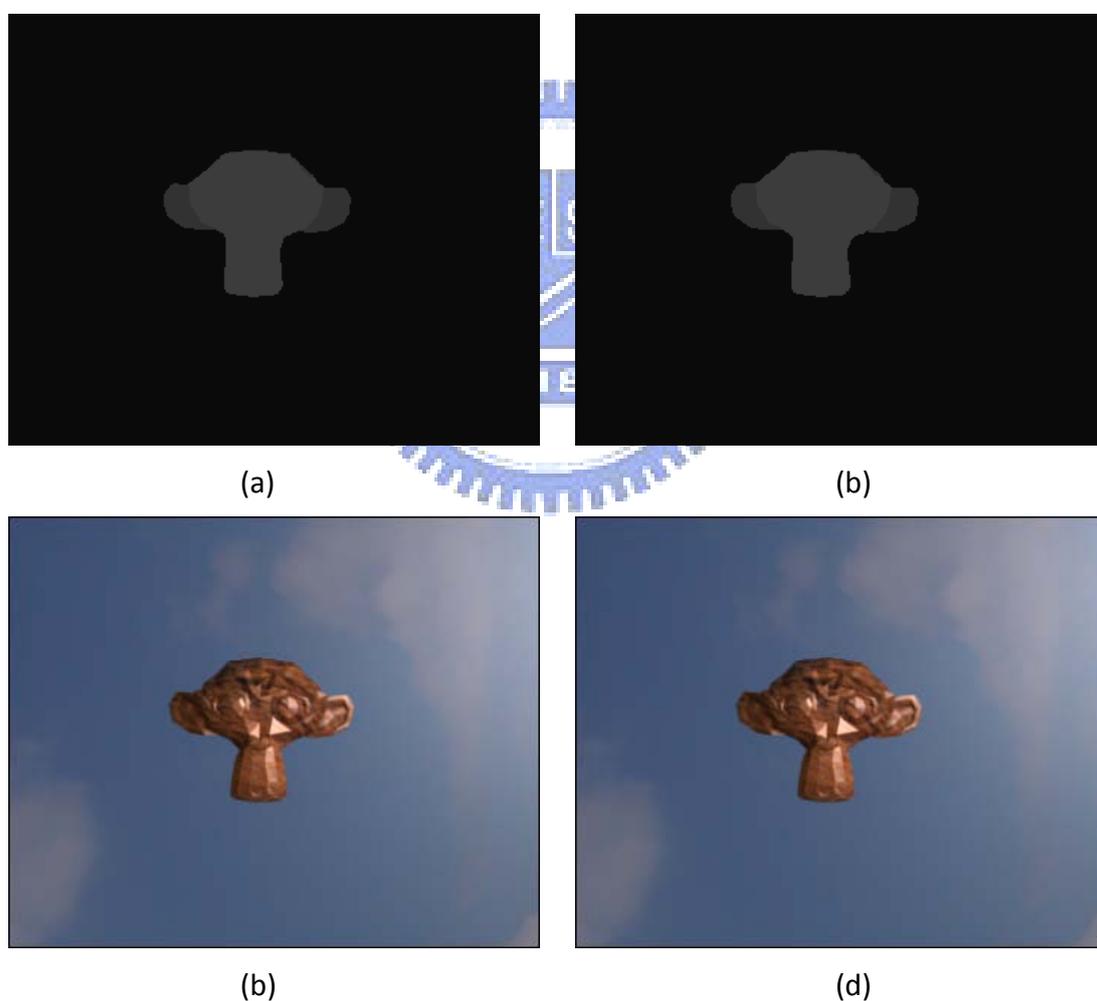


圖 4-52 (a) D_{map} (b) $D_{MapEmbed}$ (c) $I_{Virtual}$ (d) $I_{VirtualEmbed}$

圖 4-53(a)、(b)分別是 I_{Virtual} 、 $I_{\text{VirtualEmbed}}$ 和實際影像的誤差，而圖 4-53(c)則是 D_{map} 和 D_{MapEmbed} 之間的差異圖。經觀察發現，兩者主要的差異在於物體邊緣的區域，這與兩者視差圖之間的差異部份類似。由此可知，物體邊緣區域的視差值精確度會影響最後合成結果。

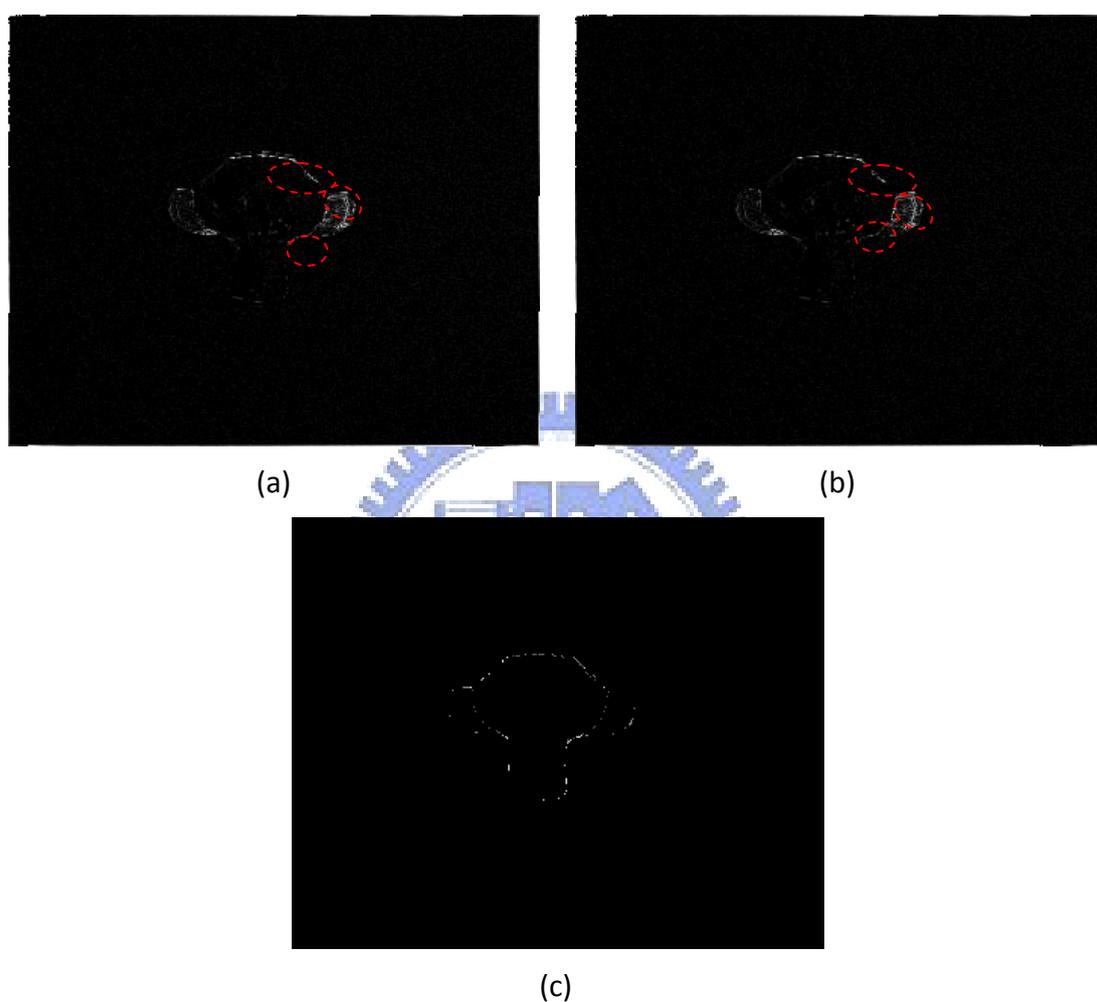


圖 4-53 (a) I_{Virtual} 和實際結果的誤差 (b) $I_{\text{VirtualEmbed}}$ 和實際結果的誤差

(c) D_{map} 和 D_{MapEmbed} 之間的差異

對於為何使用棋盤式視差估測的結果會比使用完整視差估測的結果更為精確，我們在本論文中提出了一個可能的解答。在完整的視差估測中，對於每張影像，我們都分別以對應點搜尋的方式求出四張參考不同位置影像的視差圖，在藉

由投票的方式從這四張視差圖中決定一個最佳解。所以，使用完整視差估測相當於參考四張影像來計算視差值，如圖 4-54。

在棋盤式視差估測中，主視差圖的取得方式和上述方法相同，所以主視差圖也是相當於參考四張影像來進行估測；副視差圖是參考鄰近四張已知的主視差圖，以投票的方式得到的，而每張主視差圖又是由鄰近四個方向的視差圖決定的。因此，一張副視差圖實際上相當於參考了十六張的影像資訊，如圖 4-55。這明顯比完整視差估測所參考的資訊量大很多，使得副視差圖中物體邊緣區域的視差值會比使用完整視差估測得到的結果更為精確。

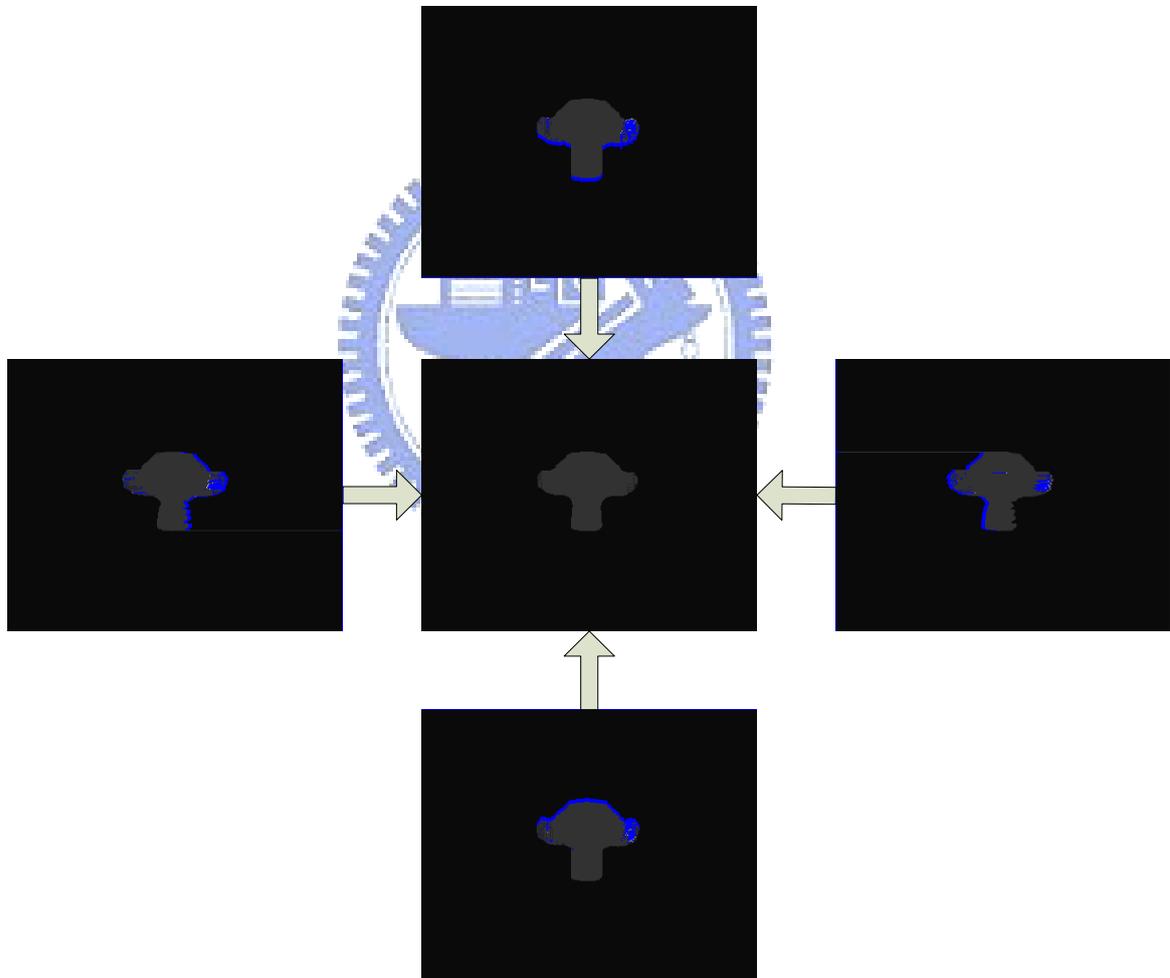


圖 4-54 使用完整視差估測相當於參考四張影像資訊

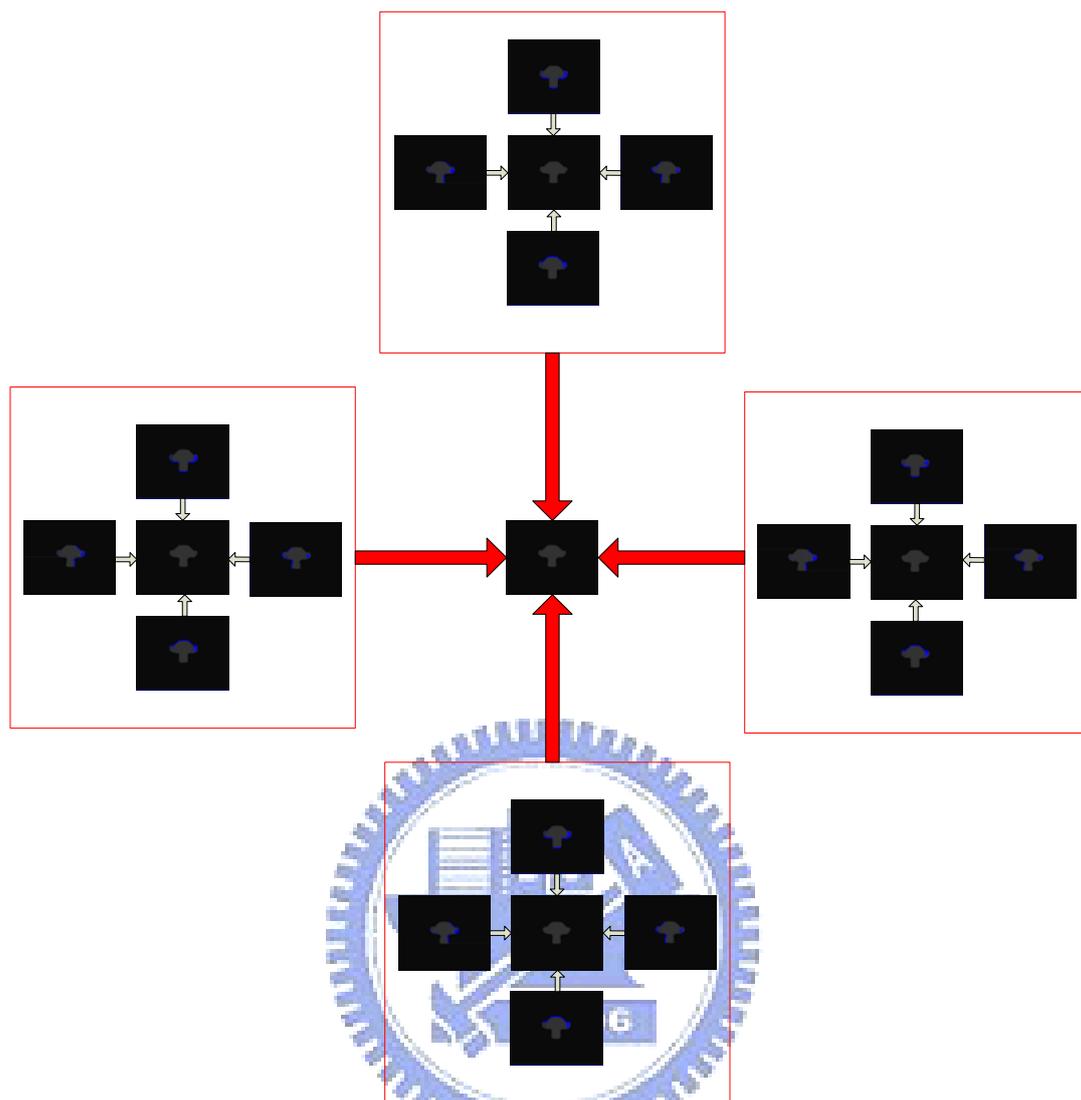


圖 4-55 使用完整視差估測相當於參考十六張影像資訊

第五章 適用於二維攝影機陣列之視訊合成方法

在第四章中，我們採用了棋盤式視差估測來降低計算二維攝影機陣列整體視差所需要的時間，但其運算量仍然很大。若要將虛擬視點影像實現在視訊訊號上，我們勢必要在降低整體的計算時間。

在一般視訊中，影像和影像之間存在著時間上的關聯性(temporal collocation)，亦即在時間軸上越接近的兩張影像，其畫面會越相似。由於視差相當於反應影像中各 pixel 的深度(depth)，因此視差圖在時間軸上的變化會和影像的變化一致。在視訊壓縮中，移動向量(motion vector)被用來描述連續影像在時間變化中所產生相對應的移動特徵。在本章中，我們將使用移動向量的資訊來降低估測連續影像視差所需要的時間。

5.1 使用移動向量進行視差估測

移動估測(motion estimation)是以marcoblock作為基本單位，並藉由找出參考影像之間互相匹配的marcoblock來判斷移動向量，如圖 5-1。在常見的移動估測方法中，以全域搜尋(Full Search)所得到的移動向量最準確，因為marcoblock會去計算在搜尋視窗內每一個位置的比對誤差，其表示式如下：

$$(i, j) = \arg \min_{x, y \in R} \sum_{k=0}^R \sum_{l=0}^R |I_t(x+k, y+j) - I_{t+T}(x+i+k, y+j+k)| \quad (22)$$

其中 I_t 和 I_{t+T} 為兩張在時間軸上間隔 $T-1$ 張frame的參考影像， R 為搜尋範圍， w 為marcoblock的大小， (i, j) 為移動向量。需要注意的是，搜尋範圍會和略過的frame數量成正比增加。雖然完全搜尋移動估測是最費時的移動估測，但基於準確性，我們仍然採用它作為判斷移動向量的方法。

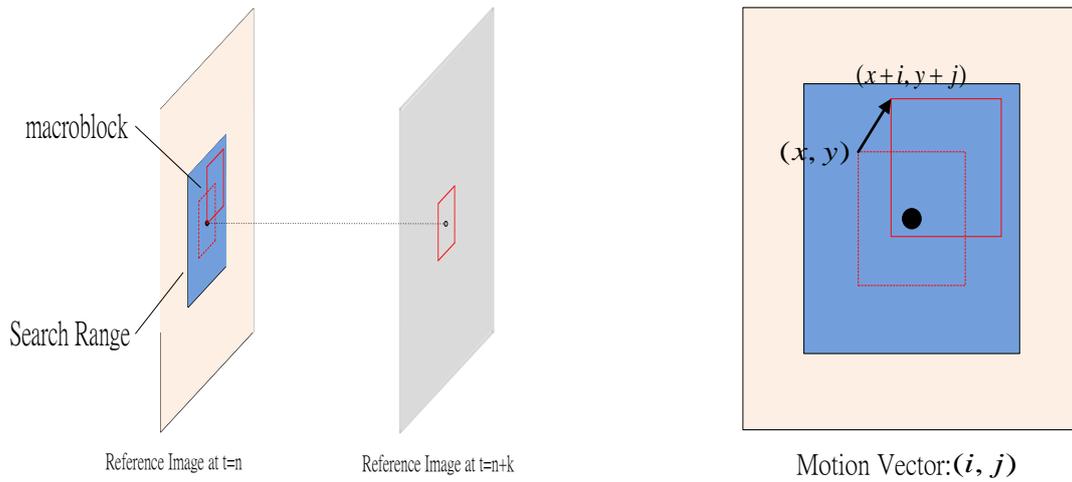


圖 5-1 macroblock 和動態向量之間的關係

在估測連續影像的視差時，我們會先以完整視差估測方式求出各張參考影像的視差圖，並計算參考影像彼此的移動向量。對於參考影像之間的影像，則以線性內插的方式合成其視差圖。圖 5-2 為參考影像跳過 7 張 frame 範例。假設兩張參考影像分別在時間軸上 $t=1$ 及 $t=9$ 的位置，彼此間存在一組對應的 macroblock，其視差值分別為 3 和 5。在 $t=4$ 上，該組 macroblock 的動態向量和影像交錯的區域之視差值為 $0.625 \times 3 + 0.375 \times 5 = 3.75$ ，並四捨五入為 4。

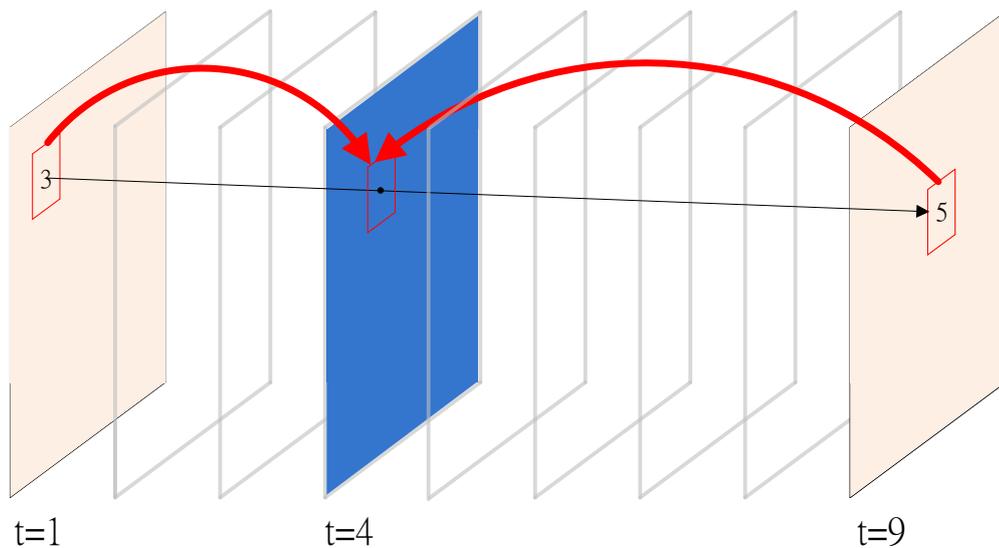


圖 5-2 使用移動向及線性內插的方式合成視差圖

圖 5-3(c)為參考移動向量來合成出來的視差圖，其中藍色區域是無視差值定義的區域，這表示該區域無任何移動向量經過。和使用完整視差估測方式所得的視差圖比較，經實驗觀察發現，無視差值區域的實際視差會和鄰近區域相同。因此，若無視差值區域周圍的視差值皆相同，則我們以該視差值來填補無視差值區域。圖 5-3(d)為填補後的結果，可以看出視差圖上只剩下少數物體邊緣的區域無視差定義。接著，我們各別處理每塊無視差值定義的區塊。對於每塊無視差值區域上各點，我們以動態規劃的方式參考鄰近四張影像來得到四個視差值，再以投票的方式選擇出現次數最多的視差值作為該點的視差值。需要注意的是，此時我們並不需要對整張影像進行搜尋，而只要搜尋無視差值區域以及其周遭範圍即可。最後的合成結果如圖 5-3(e)。和完整估測方式所得的視差圖比較，兩者之間的差異如圖 5-3(f)。

雖然上述的範例中，利用移動向量所得的視差圖錯誤僅發生在物體邊緣，但隨著參考影像之間距離的增加，移動估測所判斷的移動向量很容易與影像在時間上的實際變化不合，而導致視差圖中的錯誤增加。圖 5-4是兩張參考影像在時間軸上略過七張frame所合成的視差圖，可以看出在標記區域發生了嚴重的視差值錯誤。為了降低視差圖上的錯誤，我們在下一節中修正原有的全域搜尋移動估測來提昇移動向量精確度。

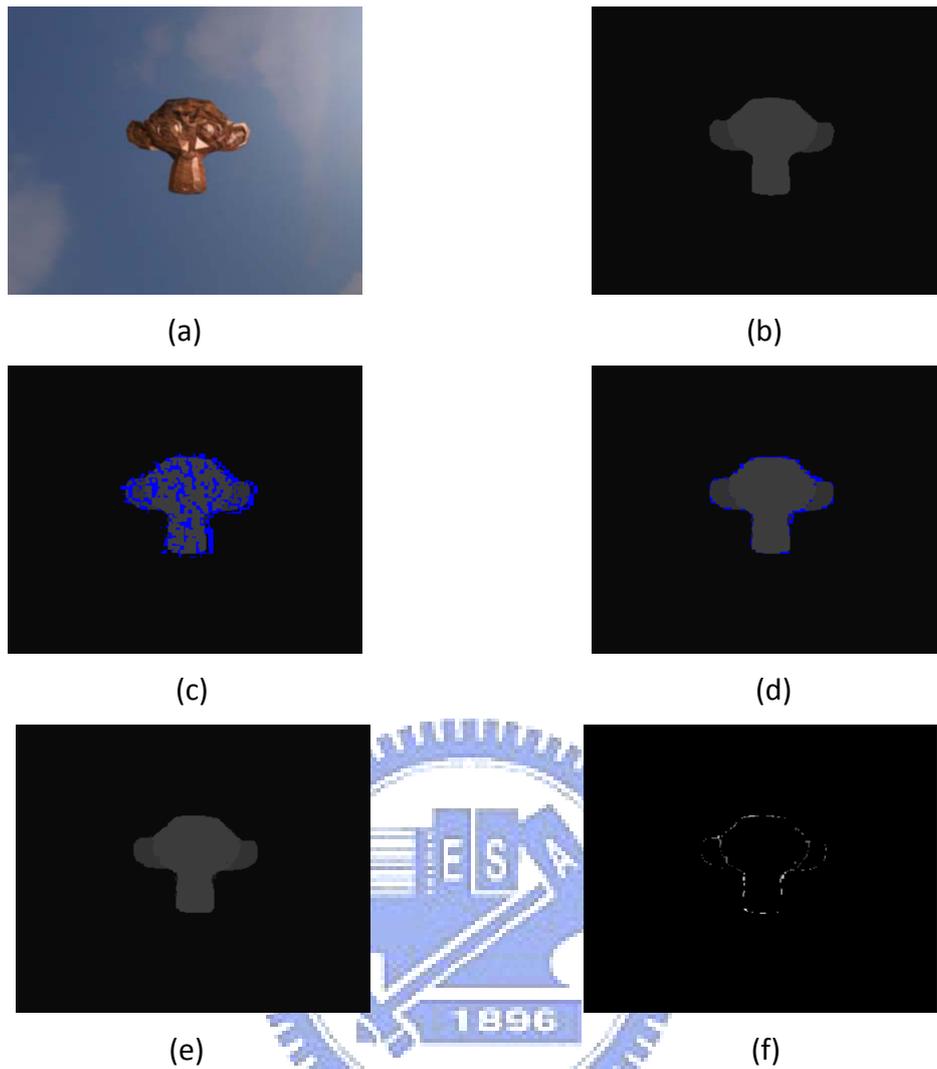


圖 5-3 (a)原始影像 (b)使用完整視差估測方式得到視差圖 (c)利用移動向量合成的視差圖 (d)使用周圍視差值填後補結果 (e)最後合成之視差圖 (f)兩種方式所得的視差圖之間的差異



圖 5-4 (a)參考影像間略過七張 frame 所合成的視差圖 (b)和使用完整視差估測的視差圖之間的差異

5.2 參考視差資訊進行移動估測

一般移動估測中僅使用參考影像的資訊來判斷兩個marcoblock之間的相似度，所以移動向量容易受到物體表面材質以及雜訊的影響。和參考影像相比，視差圖的影像資訊具有兩種特性：(1)影像結構較簡單，因為視差圖僅反應影像中物體和攝影機之間的距離。在兩張視差圖間尋找對應的marcoblock時，錯誤的對應會帶來明顯的cost，如圖 5-5。(2)視差圖本身是使用整體最佳解的方式求得，因此較不受物體表面材質及雜訊所影響。

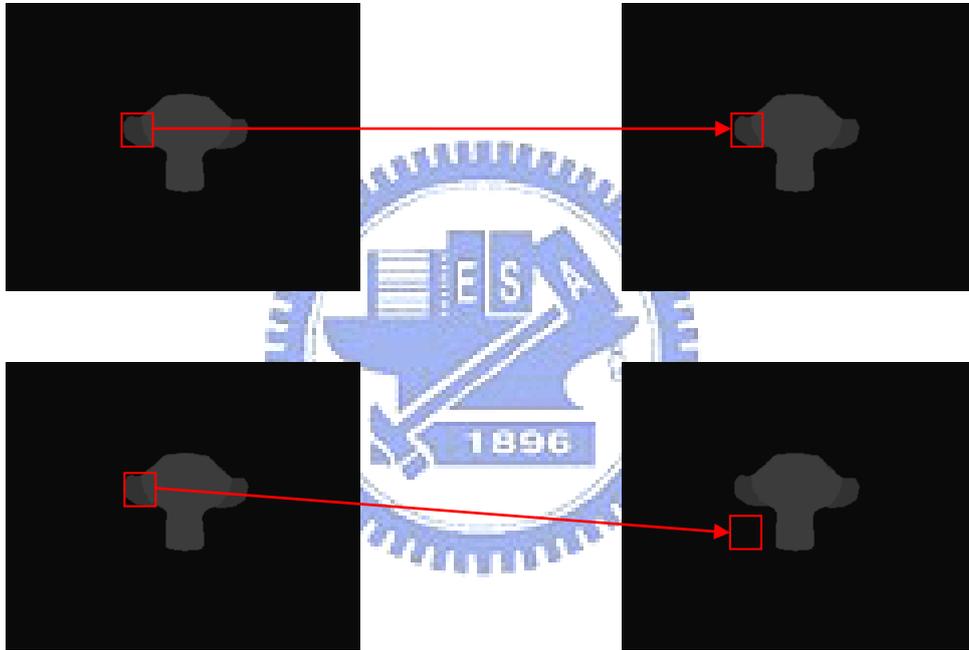


圖 5-5 (a)視差圖上 marcoblock 對應到正確位置的情形 (b)視差圖上 marcoblock 對應到不正確位置的情形

基於以上特性，我們加入視差圖資訊來提昇移動偵測精確度，並將(22)式改寫如下：

$$(i, j) = \arg \min_{x, y \in R} \sum_{k=0}^w \sum_{l=0}^w [D_t(x+k, y+j) - D_{t+T}(x+i+k, y+j+k)]^2 + \alpha [I_t(x+k, y+j) - I_{t+T}(x+i+k, y+j+k)]^2 \quad (23)$$

其中 D_t 和 D_{t+T} 分別是參考影像 I_t 和 I_{t+T} 的視差圖，而 α 為一個比例常數。圖 5-6

是 $\alpha = 1$ 時參考影像間跳過7張frame所合成出來的視差圖。和圖5-4相比，可以明顯看出在背景的視差值錯誤明顯變少了。

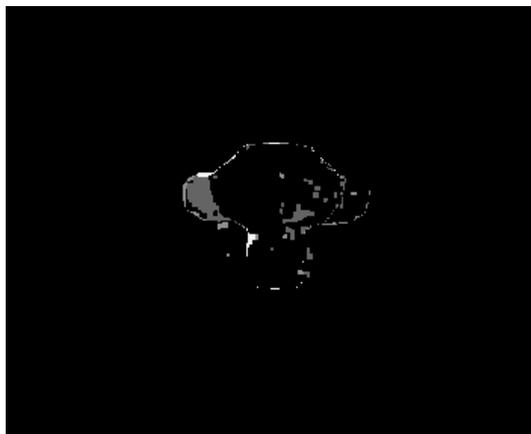


圖 5-6 加入視差資訊後合成的視差圖和使用完整視差估測的視差圖之間的差異

由於視差圖和原始影像之間並非一對一且可為分的函數關係，所以 α 值的選擇為一個經驗解。以下我們假定完整視差估測方式所得的視差圖為ground truth，並藉由測試和圖 4-39中五種連續影像中常見的物體運動行為來討論該如何選擇 α 值。

圖 5-7到圖 5-11為係數 α 和不同運動行為之間的關係，其中Skipping k Frame表示兩張參考影像之間略過了k張frame。觀察這五張表可以發現，對於不會改變物體視差值的運動行為，例如平移、旋轉及覆蓋，在 α 值在小於0.1的時候合成出來的視差圖結果較好。這表示在該類運動行為中，視差圖資訊的比重需要比原始影像資訊大。而物體靠近或遠離攝影機時， α 值在大於1時有較好得結果。這是因為遠離及靠近的運動行為會改變物體本身視差值，導致應該是互相對應的marcoblock會因視差值不同而得到較大的cost。這類運動行為會降低視差值在移動估測中的可靠性。

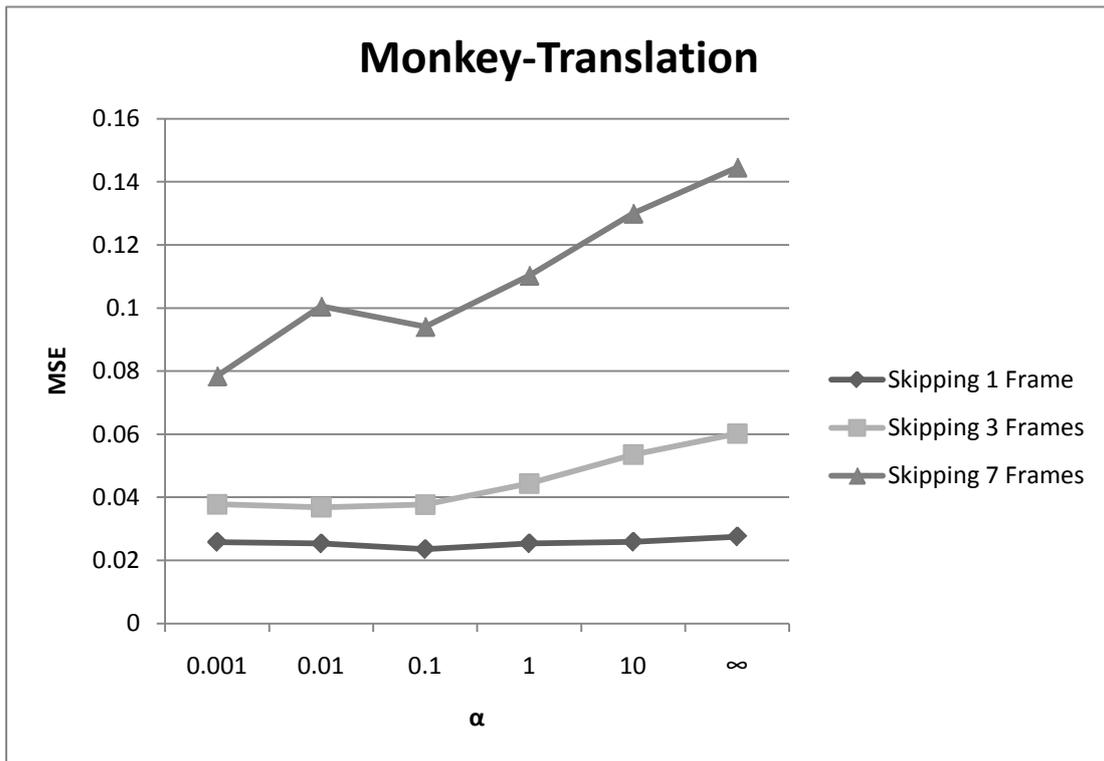


圖 5-7 物體平移和係數 α 之間的關係

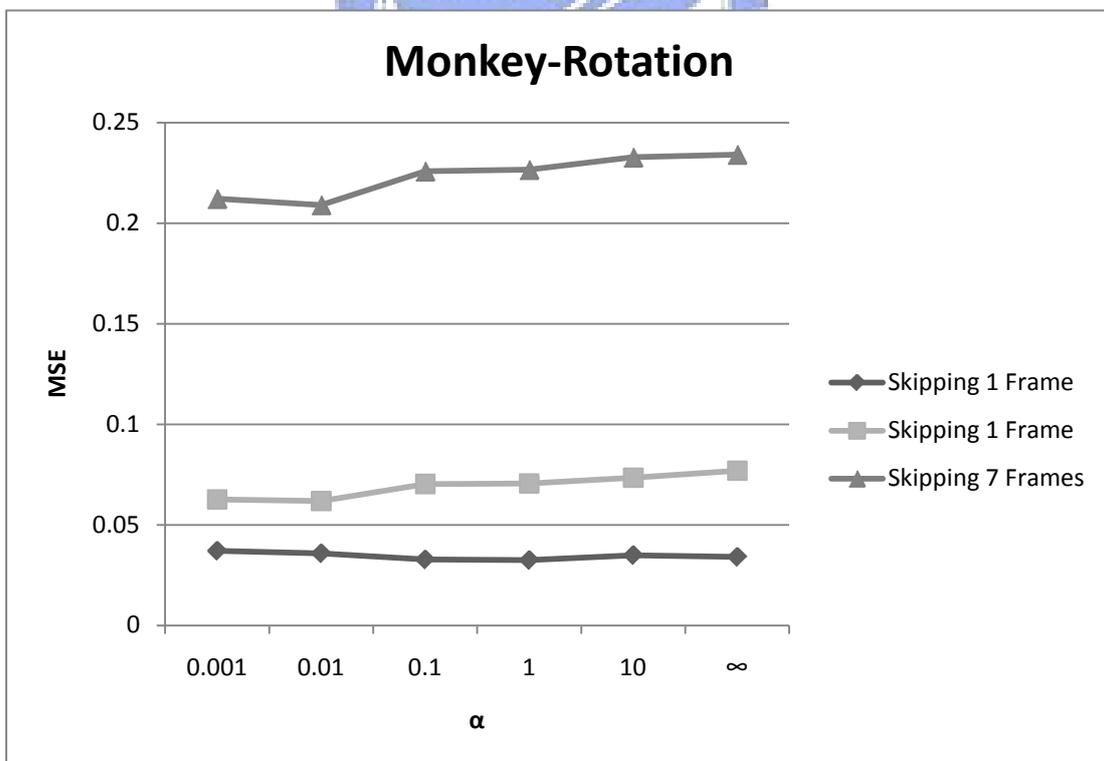


圖 5-8 物體旋轉和係數 α 之間的關係

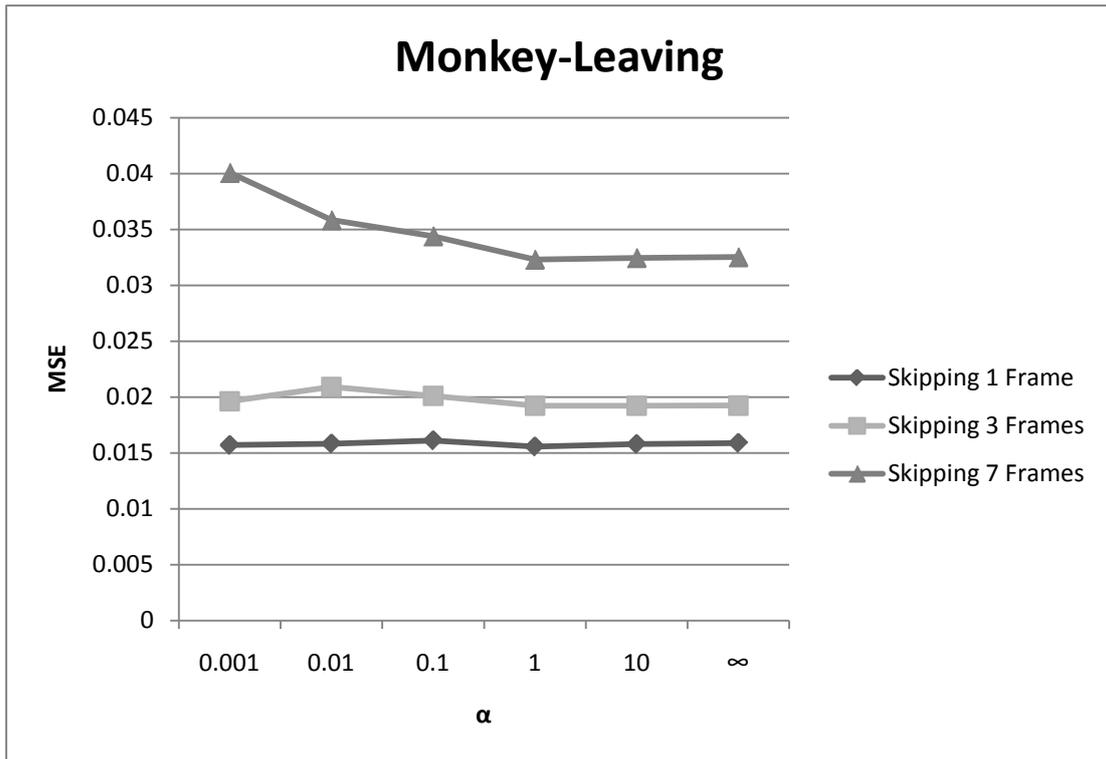


圖 5-9 物體遠離和係數 α 之間的關係

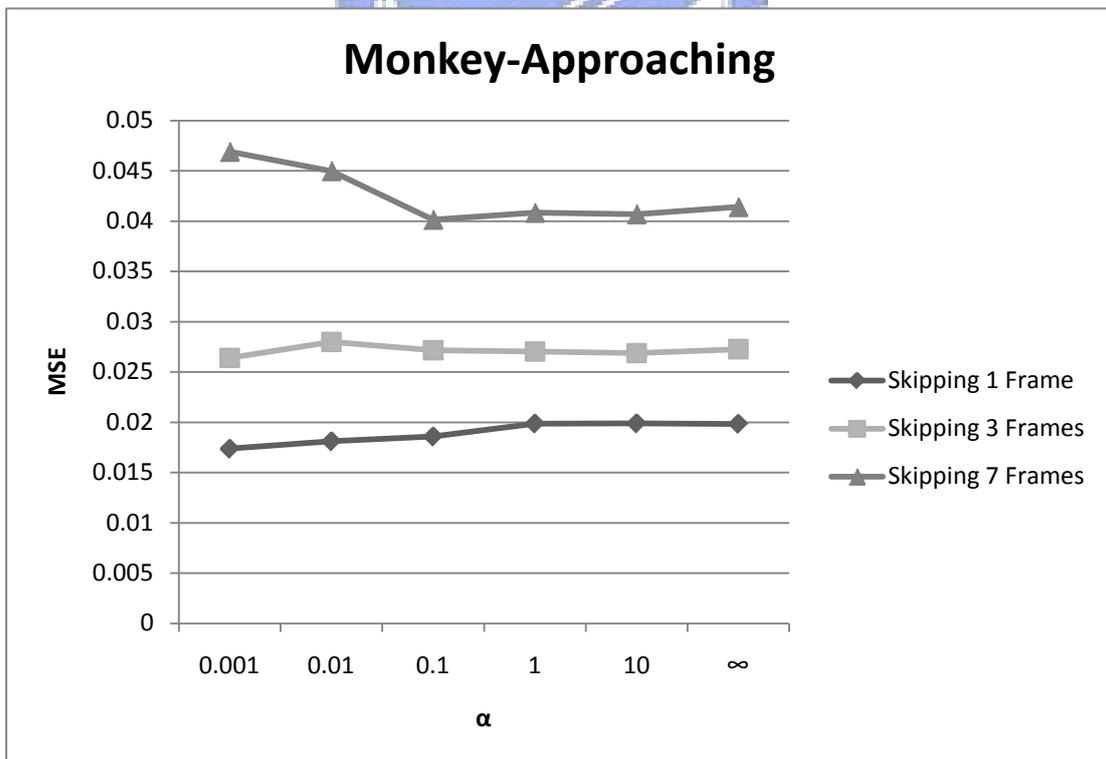


圖 5-10 物體靠近和係數 α 之間的關係

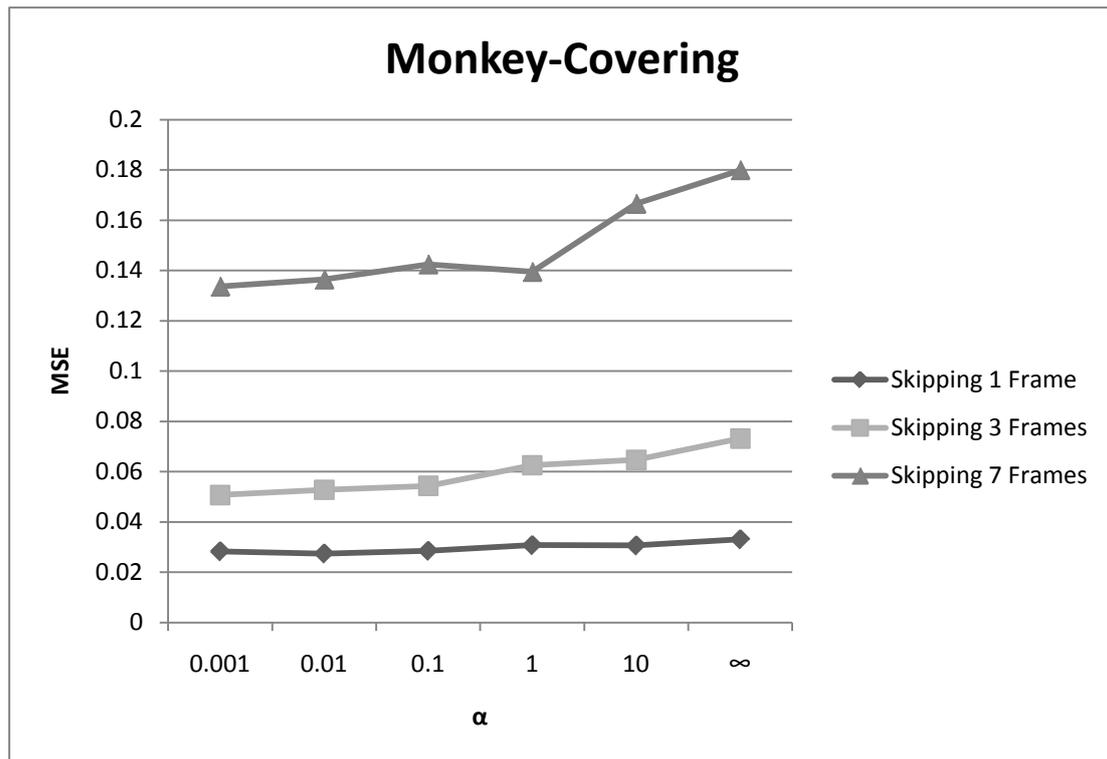


圖 5-11 物體彼此覆蓋和係數 α 之間的關係

圖 5-12到圖 5-14是將略過frame數量相同的運動行為放在一起比較。隨著參考影像之間略過的frame數量增加，係數 α 對於平移、旋轉以及覆蓋這三種運動行為影響越強；對於遠離和接近這兩種運動行為，不論略過的frame數量多少，係數 α 對結果的影響都明顯小很多。此外，物體遠離及靠近在視差圖上的誤差也明顯小於另外三種運動行為。因此，係數 α 在選擇上以平移、旋轉及覆蓋為主要考量。經實驗觀察，係數 α 在0.01到0.1之間對於大部連續影像都有良好得效果。

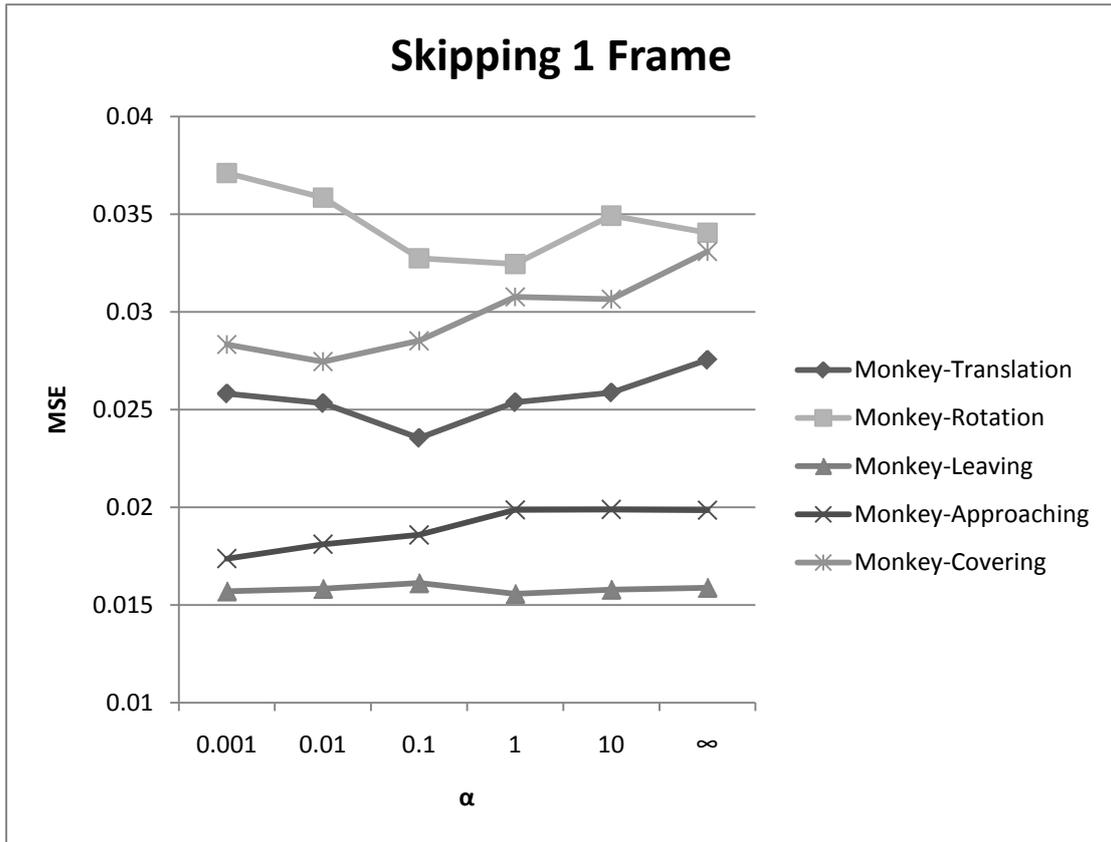


圖 5-12 參考影像之間略過一張 frame 和係數 α 的關係

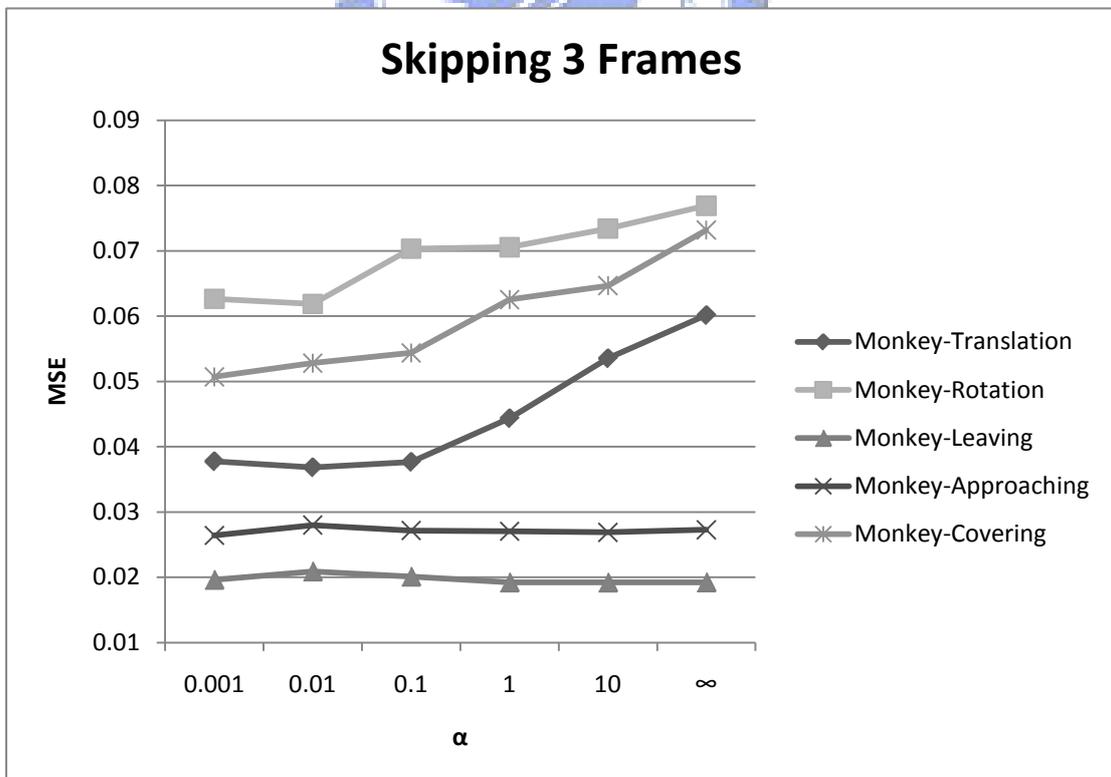


圖 5-13 參考影像之間略過三張 frame 和係數 α 的關係

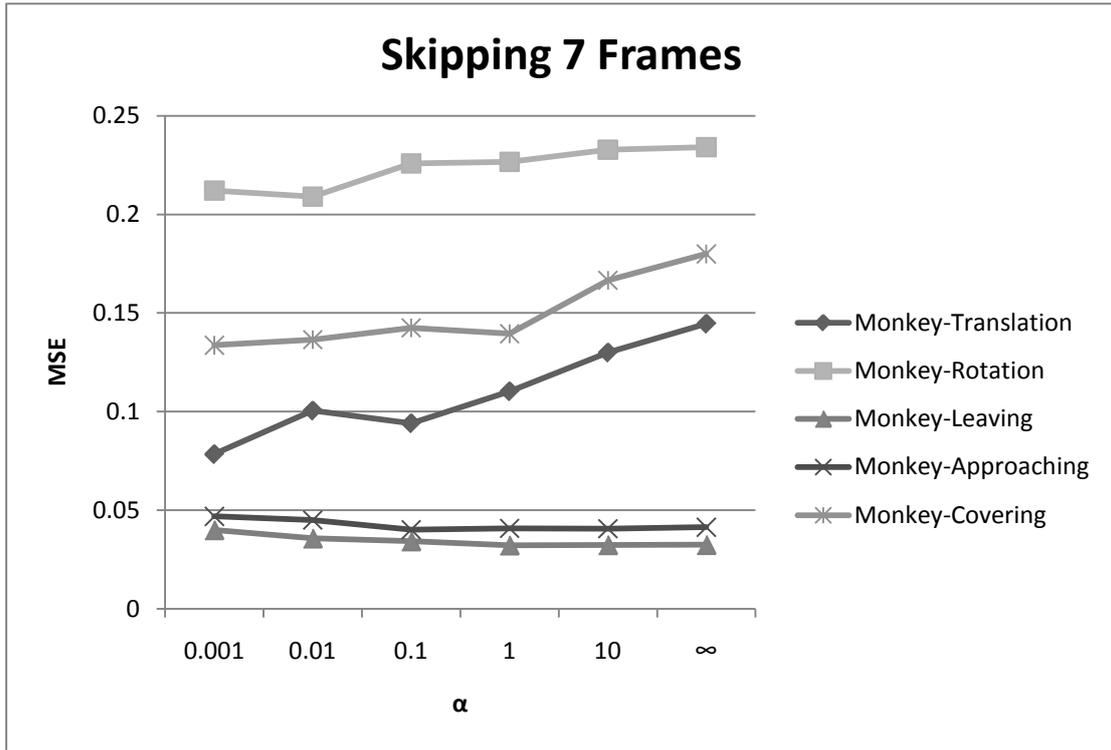


圖 5-14 參考影像之間略過七張 frame 和係數 α 的關係

在本節的最後，我們將兩張利用影像 I_T 、 I_{T+k} 知間的移動向量來估測略過影像的視差圖之流程整理如圖 5-15。

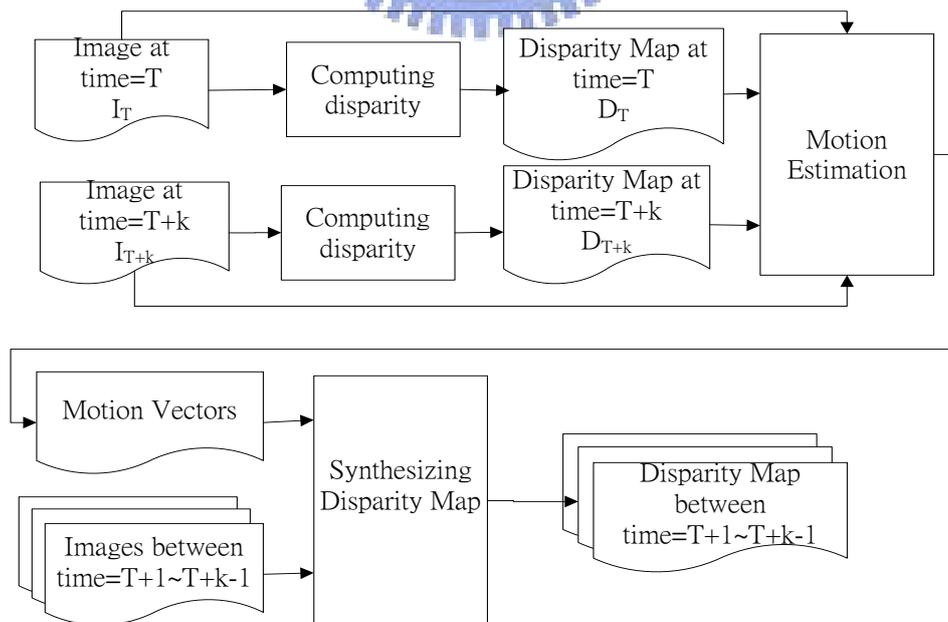


圖 5-15 利用移動向量來加速視差估測的流程

5.3 實驗結果

在本節中，我們將測試兩種物體不同運動行的連續影像，如圖4 -39和圖 5-16。我們以Blender合成的影像作為標準來比較參考影像彼此間隔的frame數量對各種運動行為所造成的影響。

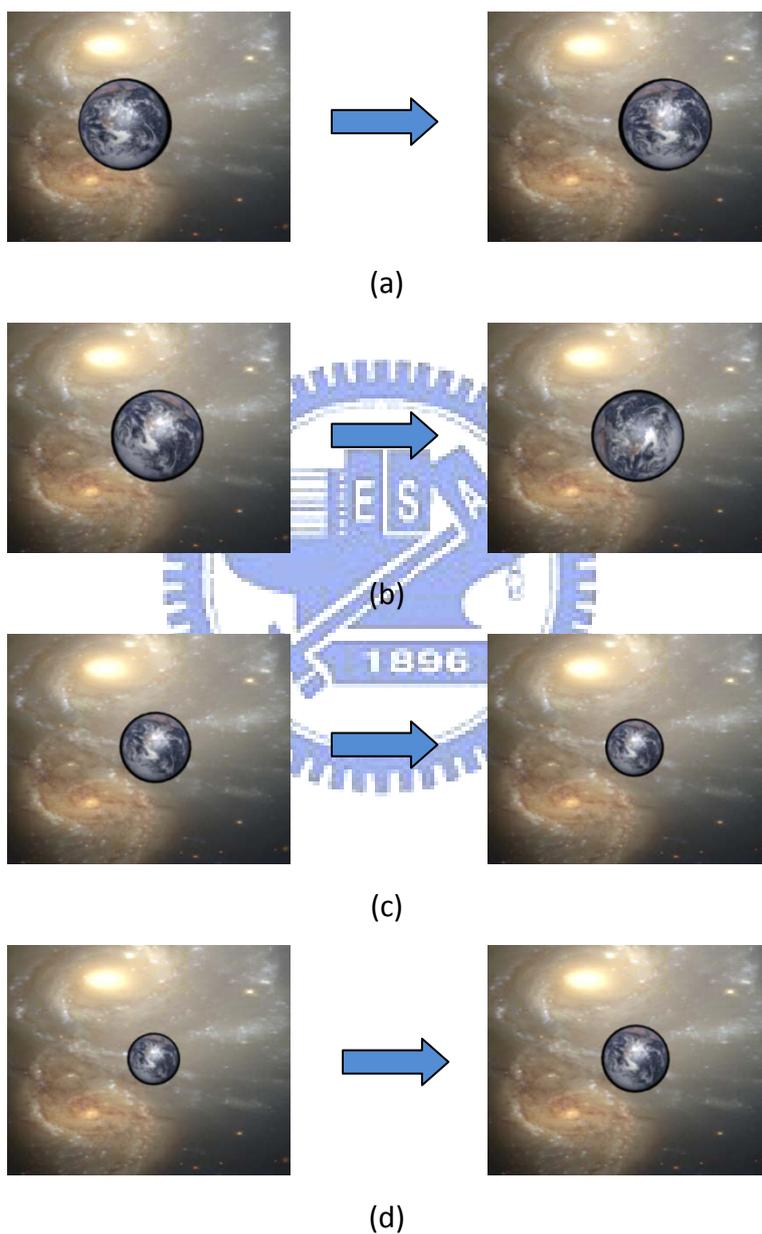


圖 5-16 不同的物體運動行為連續影像: (a)Earth-Translation(物體平移)

(b)Earth -Rotation(物體旋轉) (c)Earth -Leaving(物體遠離)

(d)Earth -Approaching(物體靠近) (e)Earth -Covering(兩物體互相遮蔽)

5.3.1 理想連續影像

圖 5-17、18為合成一張虛擬視點影像的平均花費時間與參考影像在時間軸上略過的frame數之間的關係。經實驗觀察，在略過一張frame時，計算時間可以省下約原來的三分之一。而略過三張frame時，可以省下將近原來二分之一的時間。但是當略過的frame數超過3時，整體曲線的趨勢已經趨於平緩，甚至有開始攀升的情形出現。這是因為隨著略過的frame數增加，進行移動估測時的搜尋面積也成指數增加，以至於在搜尋對應的marcoblock所花費的時間抵銷了因減少視差估算次數所帶來的效益。

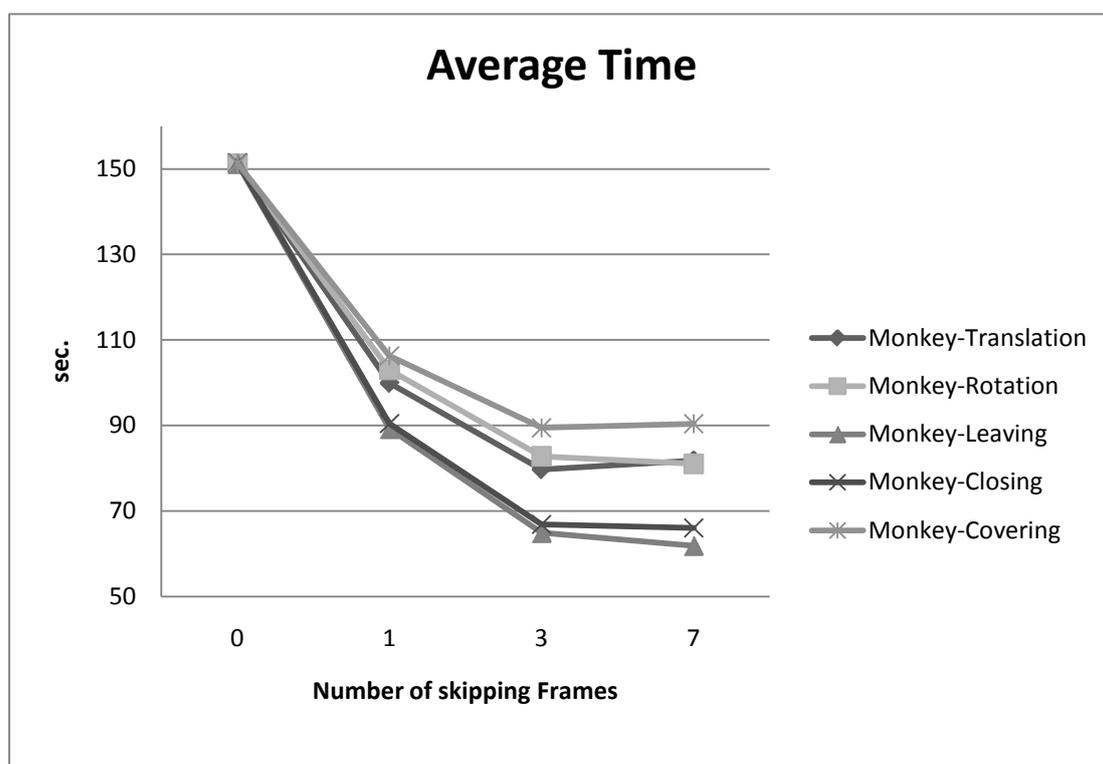


圖 5-17 合成圖 4-39中各種運動行為的影像所需平均時間和略過的frame數之關係

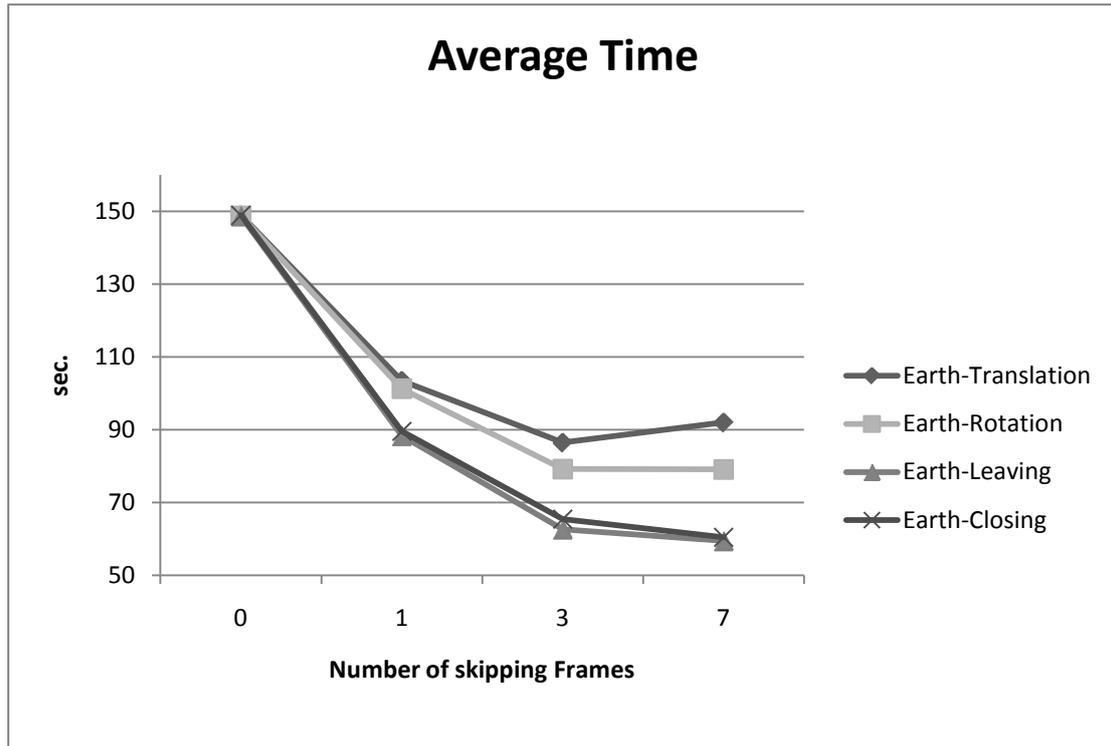


圖 5-18 合成圖 5-16中各種運動行為的影像所需平均時間和略過的frame數之關係

圖 5-19和圖 5-20分別為圖 4-39和圖 5-16合成結果的平均PSNR，其中橫軸代表參考影像之間略過的frame數量，縱軸為平均PSNR值。從圖中的趨勢可以看出，物體靠近以及遠離兩種運動行為受到略過的frame數量的影響很小；而另外三種運動行為在略過的frame數量超過3時，其平均PSNR就會快速的下降。

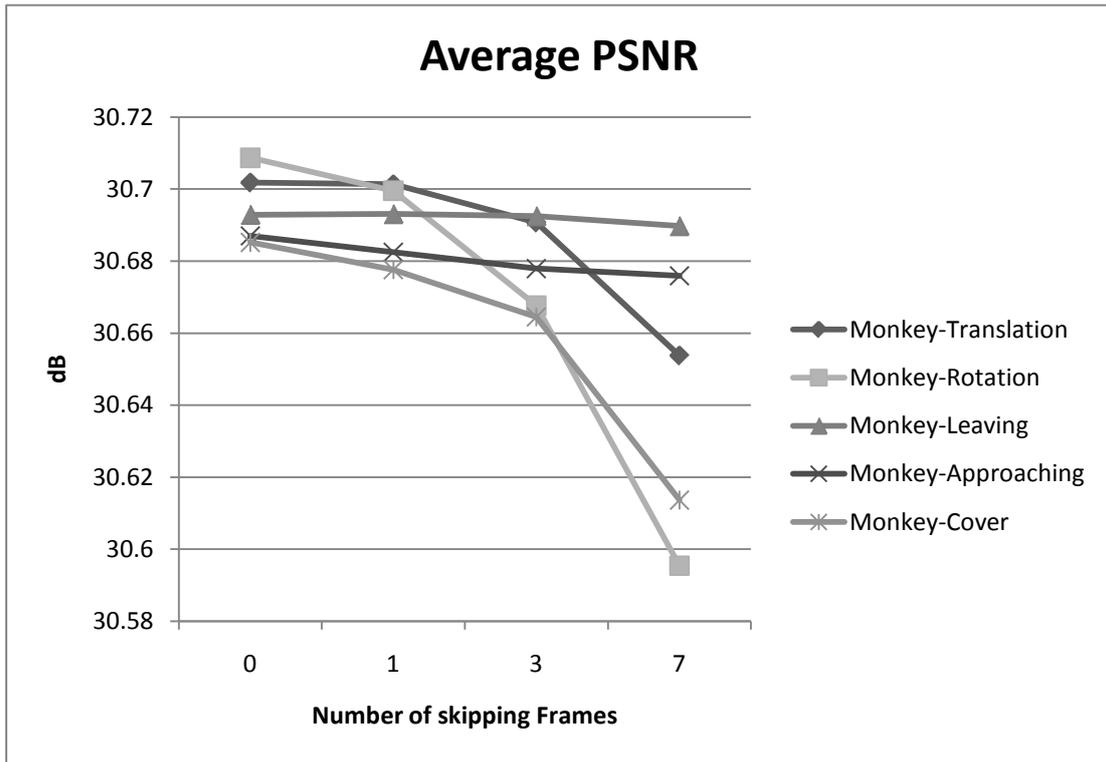


圖 5-19 在圖 4-39 中各種運動行為的合成影像品質和略過的 frame 數量之關係

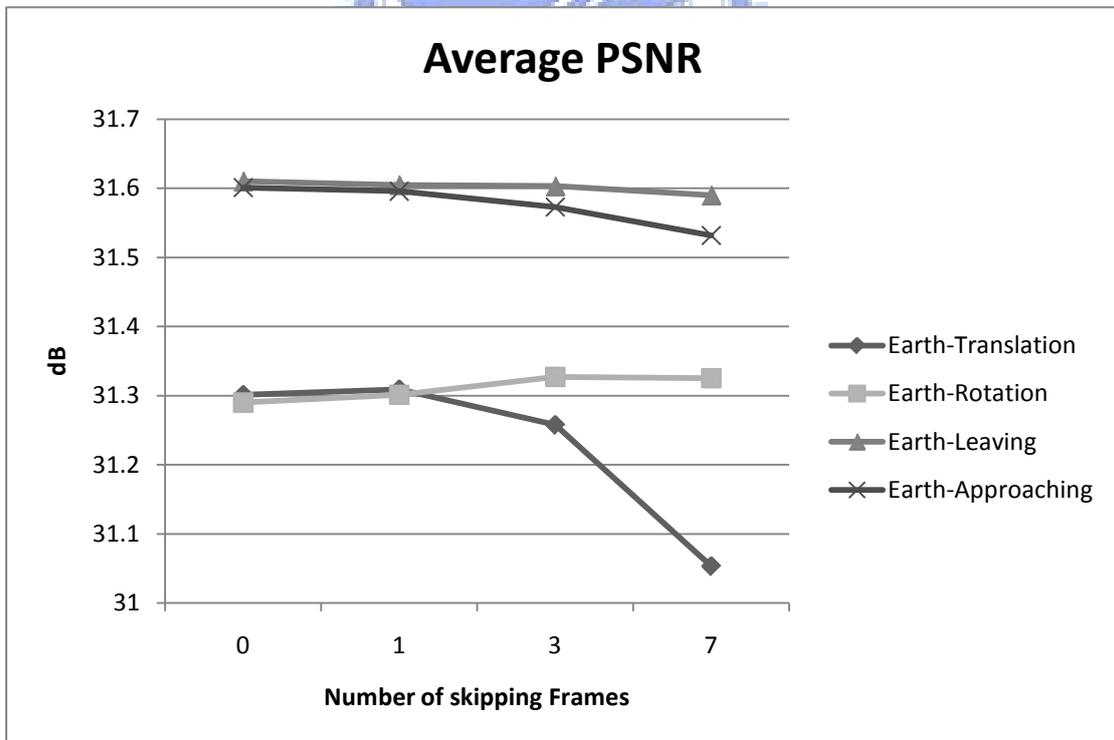


圖 5-20 在圖 5-16 中各種運動行為的合成影像品質和略過的 frame 數量之關係

圖 5-21到圖 5-29是比較同一串連續影像在略過不同的frame數量下各張合成影像的PSNR大小。從圖中可以發現在平移、旋轉及覆蓋三種運動行爲中，基本上略過得frame數越多，合成影像的品質也就越低；但是在物體靠近以及遠離的運動行爲中，會發生略過較多frame所合成的影像品質比略過較少frame的結果還更好的情形。

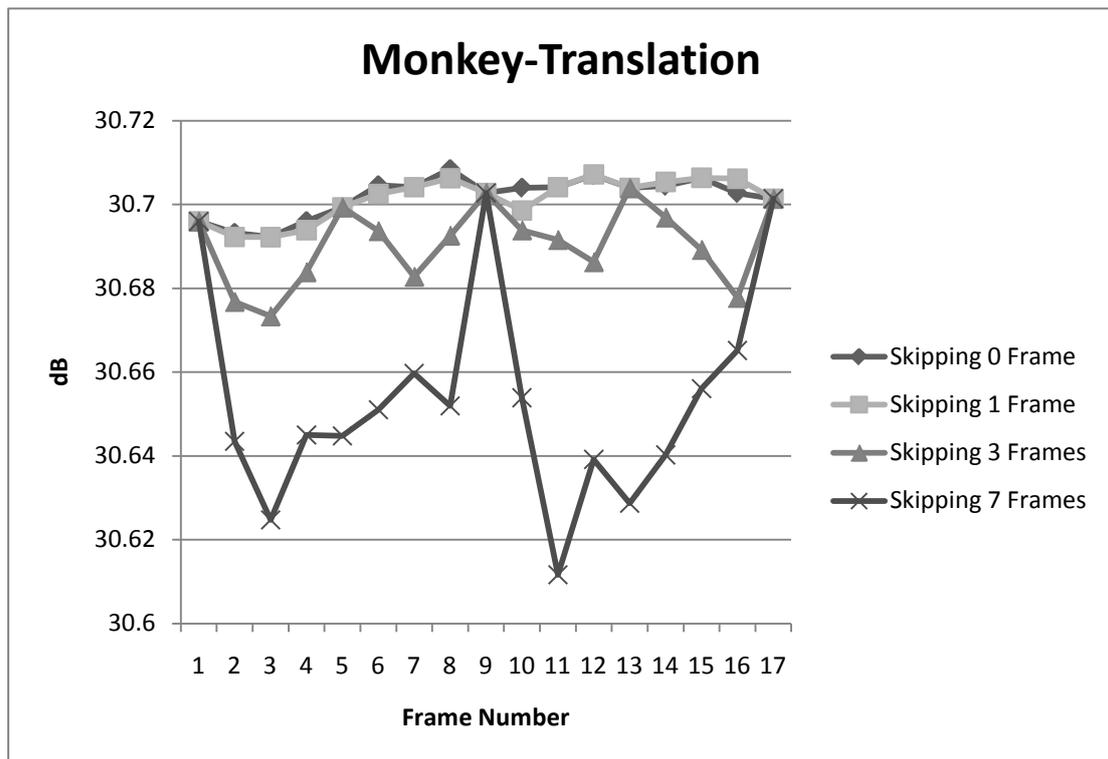


圖 5-21 連續影像 Monkey-Translation 在略過不同數量 frame 的情況下的各張合成影像品質

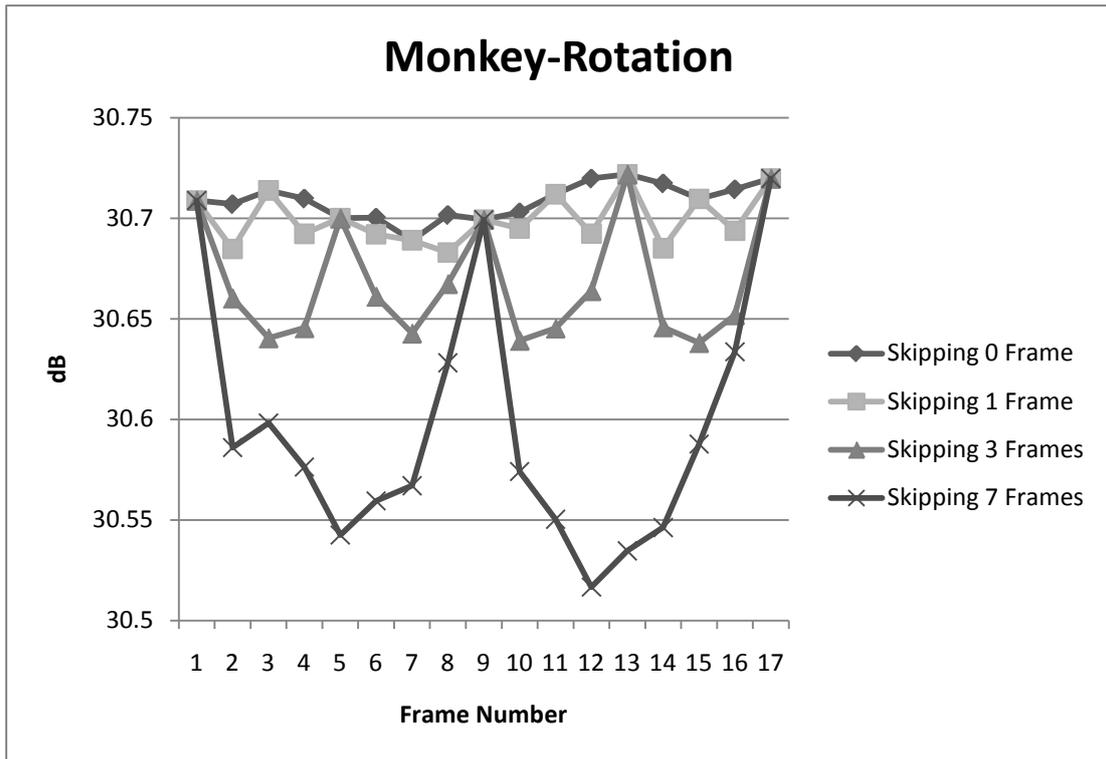


圖 5-22 連續影像 Monkey-Rotation 在略過不同數量 frame 的情況下的各張合成影像品質

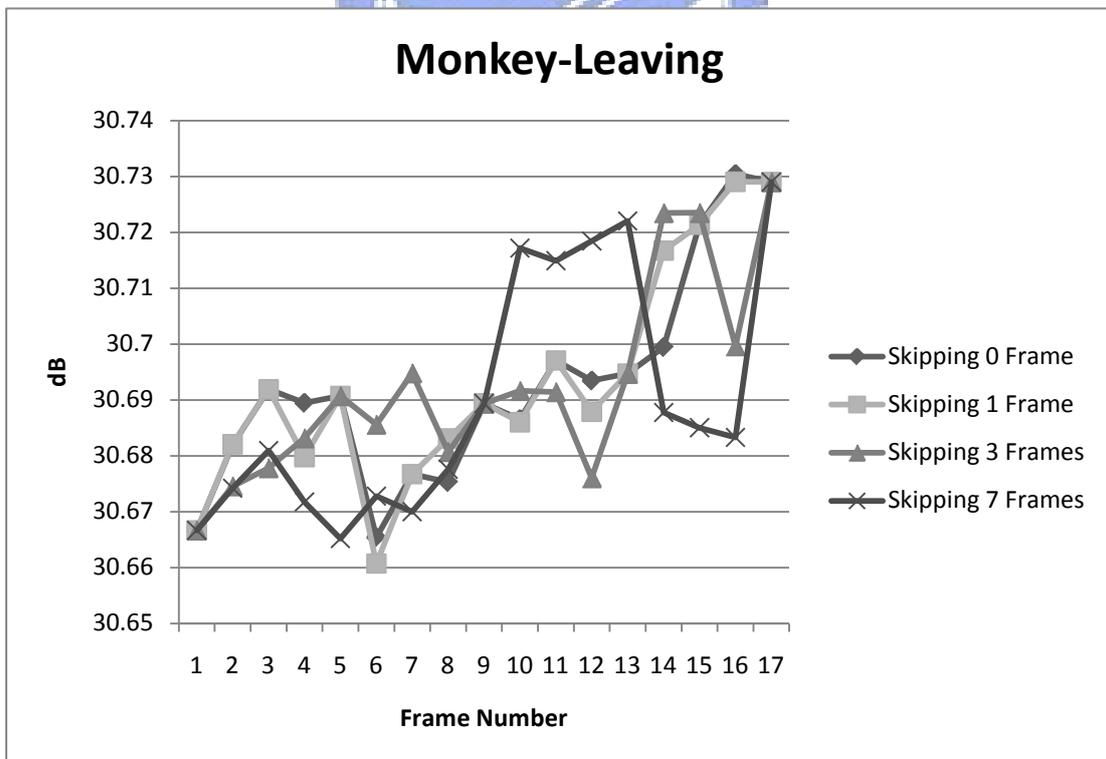


圖 5-23 連續影像 Monkey-Leaving 在略過不同數量 frame 的情況下的各張合成影像品質

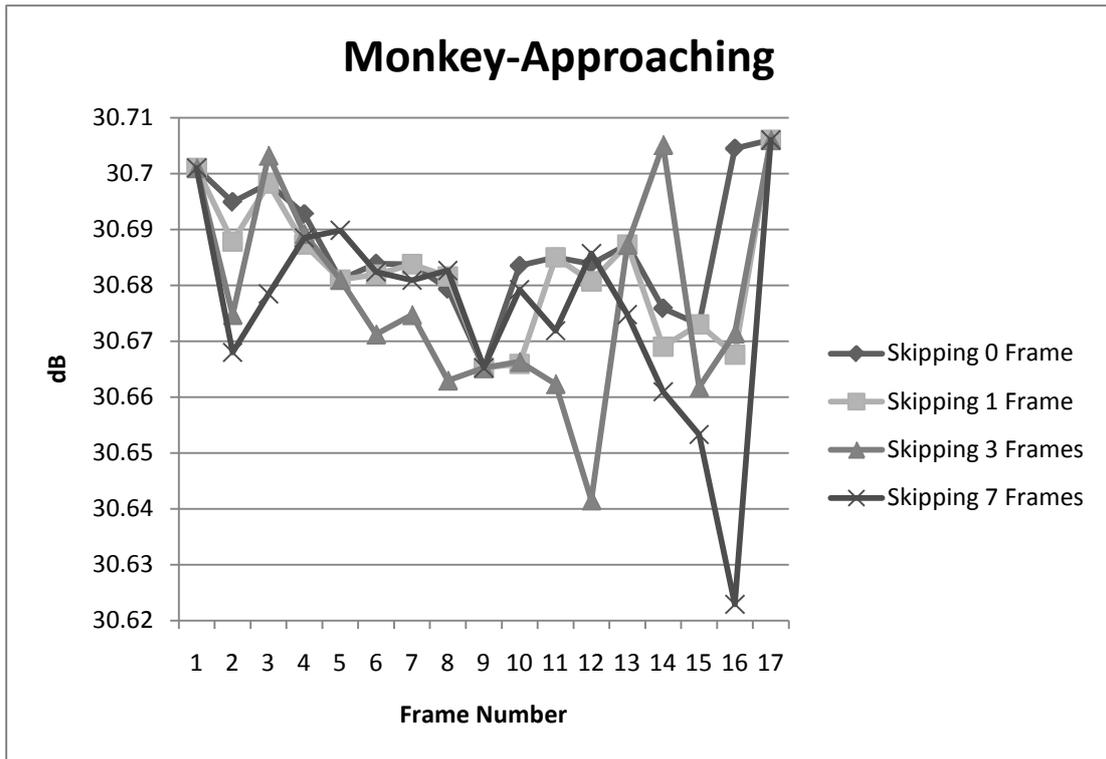


圖 5-24 連續影像 Monkey-Approaching 在略過不同數量 frame 的情況下的各張合成影像品質

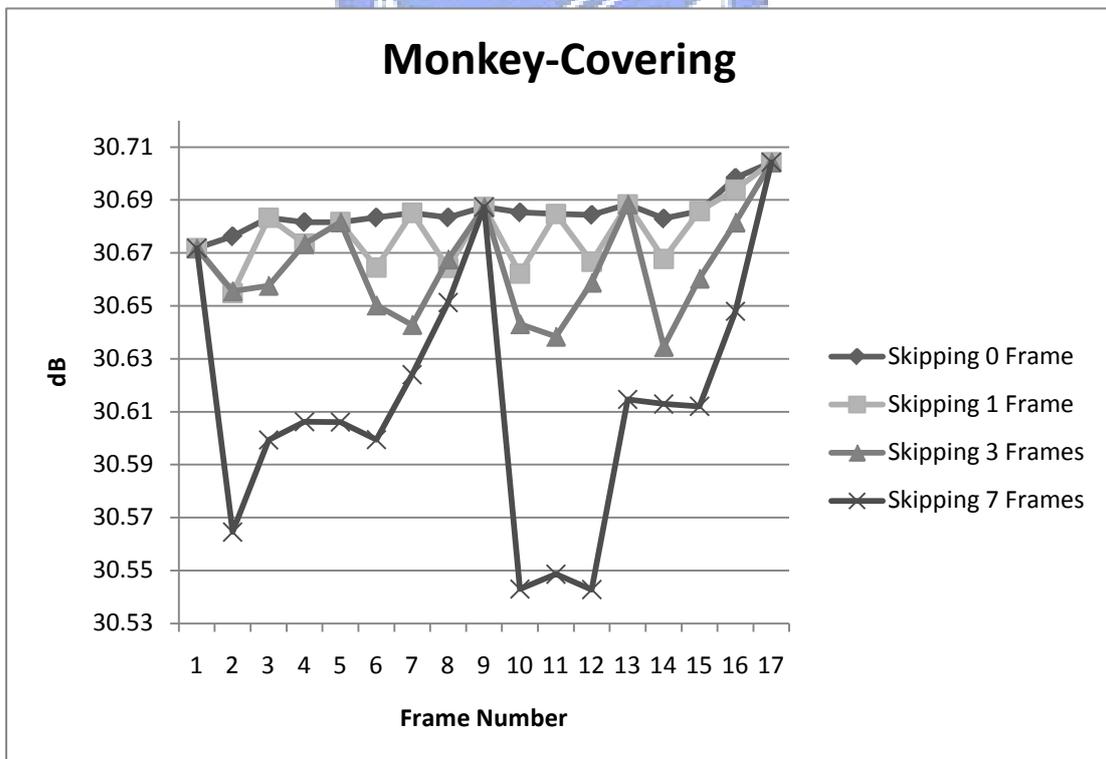


圖 5-25 連續影像 Monkey-Covering 在略過不同數量 frame 的情況下的各張合成影像品質

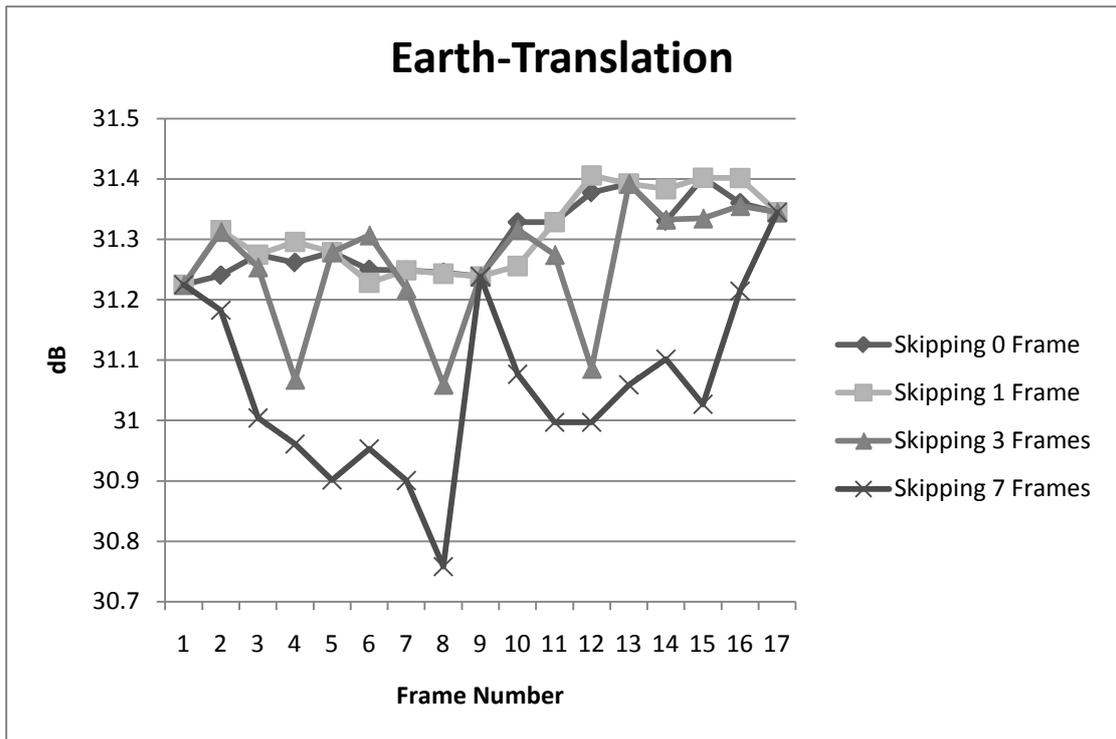


圖 5-26 連續影像 Earth-Translation 在略過不同數量 frame 的情況下的各張合成影像品質

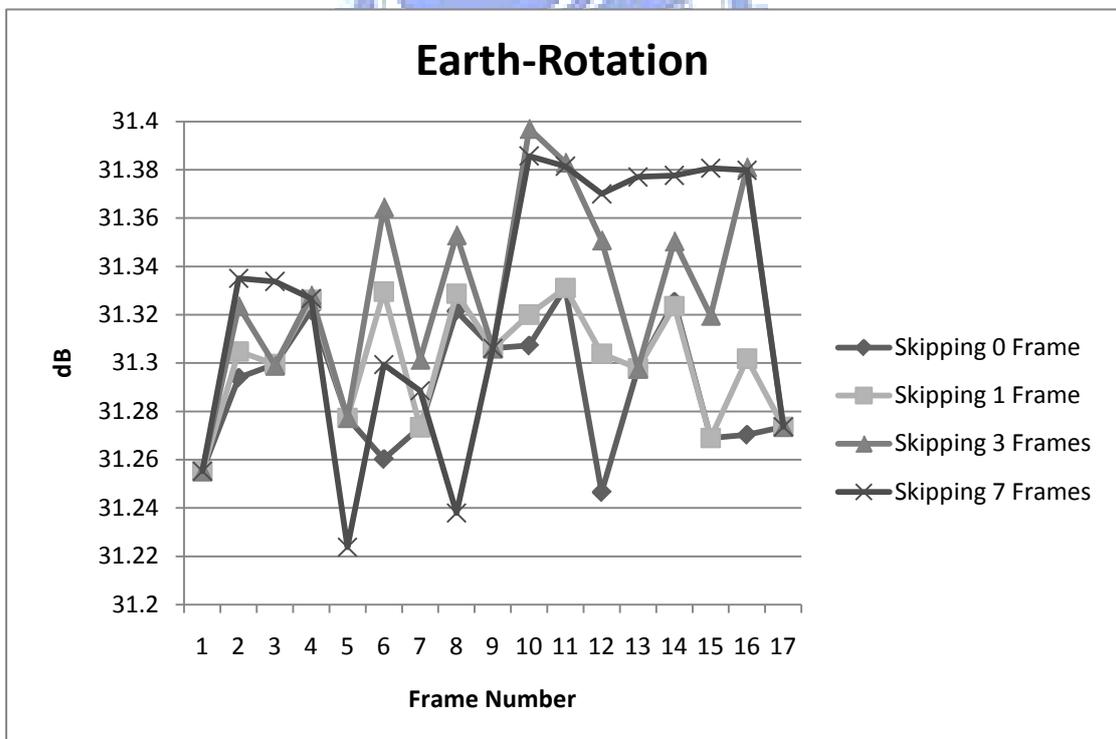


圖 5-27 連續影像 Earth-Rotation 在略過不同數量 frame 的情況下的各張合成影像品質

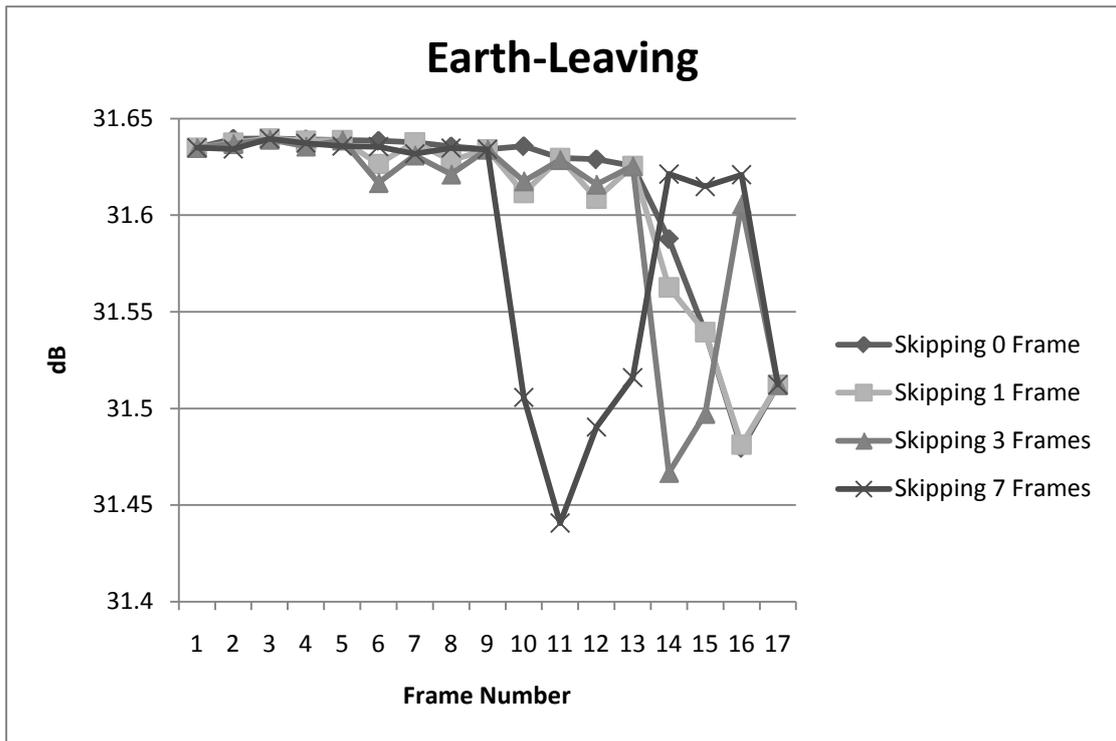


圖 5-28 連續影像 Earth-Leaving 在略過不同數量 frame 的情況下的各張合成影像品質

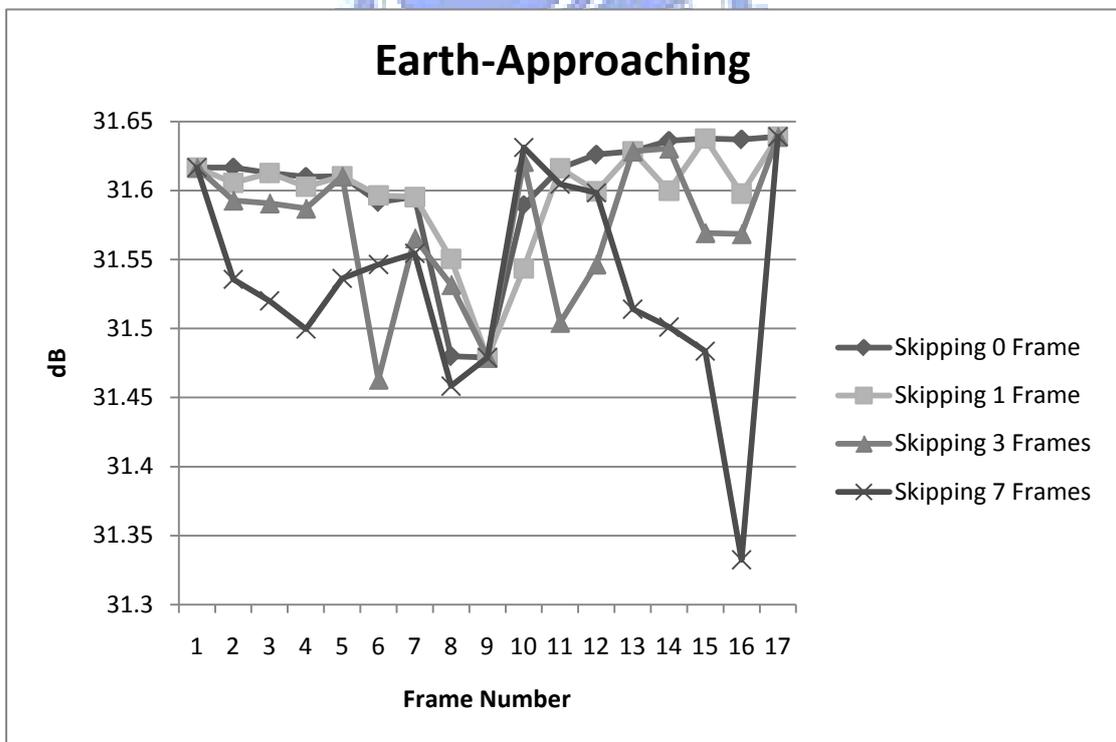


圖 5-29 連續影像 Earth-Approaching 在略過不同數量 frame 的情況下的各張合成影像品質

5.3.2 加入 AWGN 的連續影像

在上一節中，我們發現在合成的虛擬視點影像中，有時會出現參考影像間略過的 frame 較多但影像品質反而較好的情況。這有可能是因為使用 Blender 所合成的原始影像過於理想，而和現實攝影機所擷取的影像有段差距造成的。為了讓實驗結果能更接近現實中的狀況，我們在原始的連續影像中加入 AWGN。影像加入 AWGN 後，視覺上並不會有太大差異，但是 pixel 的亮度會和原始影像有所不同，這可以模擬出不同攝影機成像之間肉眼無法辨識的細微差異。

圖 5-30和圖 5-31是添加AWGN後合成影像的平均PSNR。和未添加AWGN時相比，整體的PSNR約下降了0.1dB。

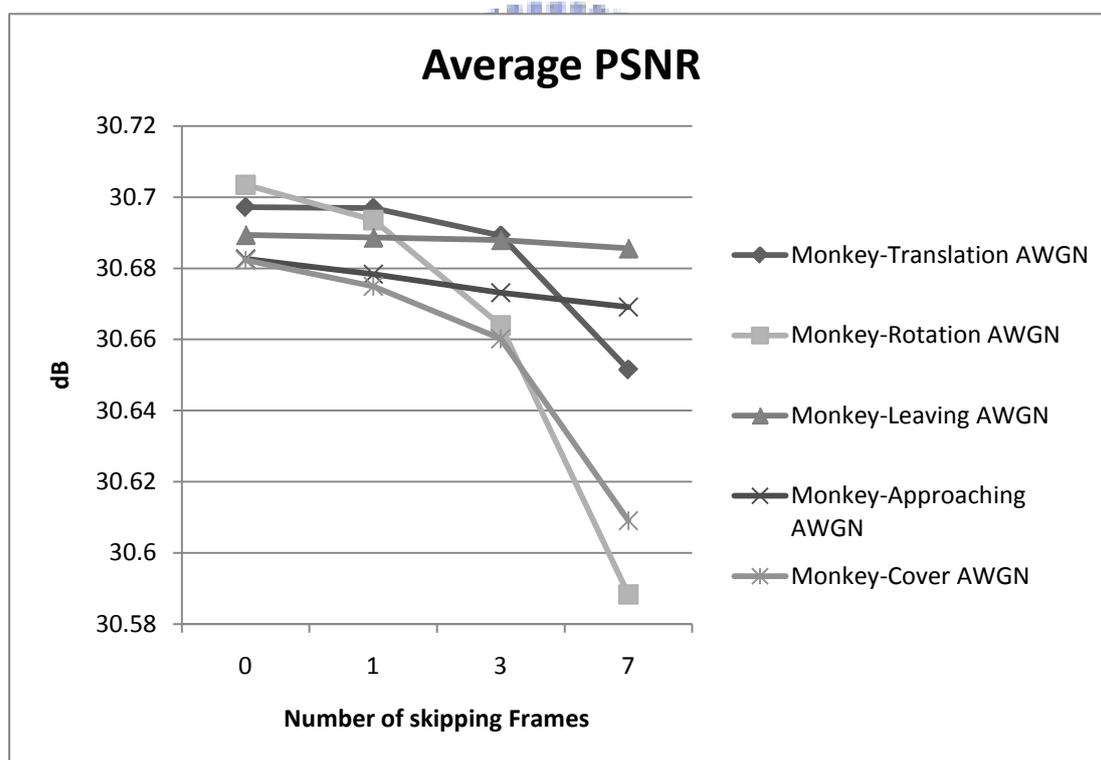


圖 5-30 在圖 4-39 添加了 AWGN 後，各種運動行為的合成影像品質和略過的 frame 數量之關係

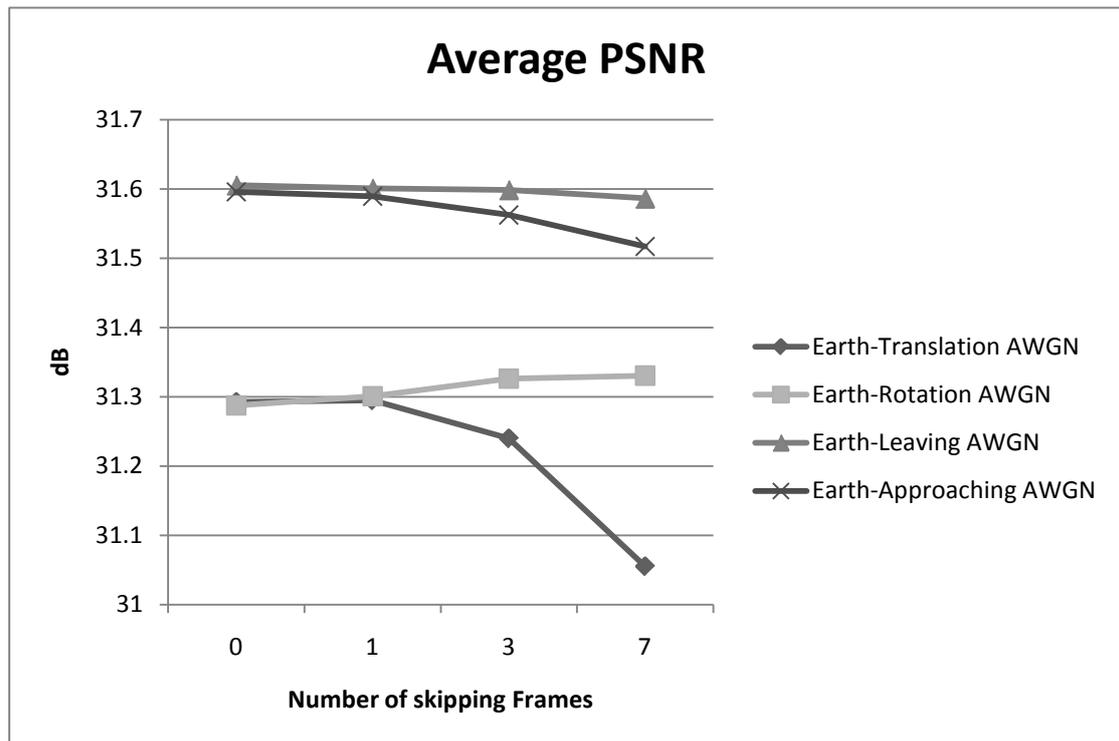


圖 5-31 在圖 5-16 添加了 AWGN 後，各種運動行為的合成影像品質和略過的 frame 數量之關係

圖 5-32 到圖 5-40 則是各張合成影像的 PSNR，可以看出除了 PSNR 下降之外，整體的曲線趨勢和未添加 AWGN 時非常相似，在物體遠離及靠近兩種運動行為中，也有出現略過較多的 frame 反而 PSNR 提昇的情況，顯然過於理想的影像並不是造成此現象的主要原因。

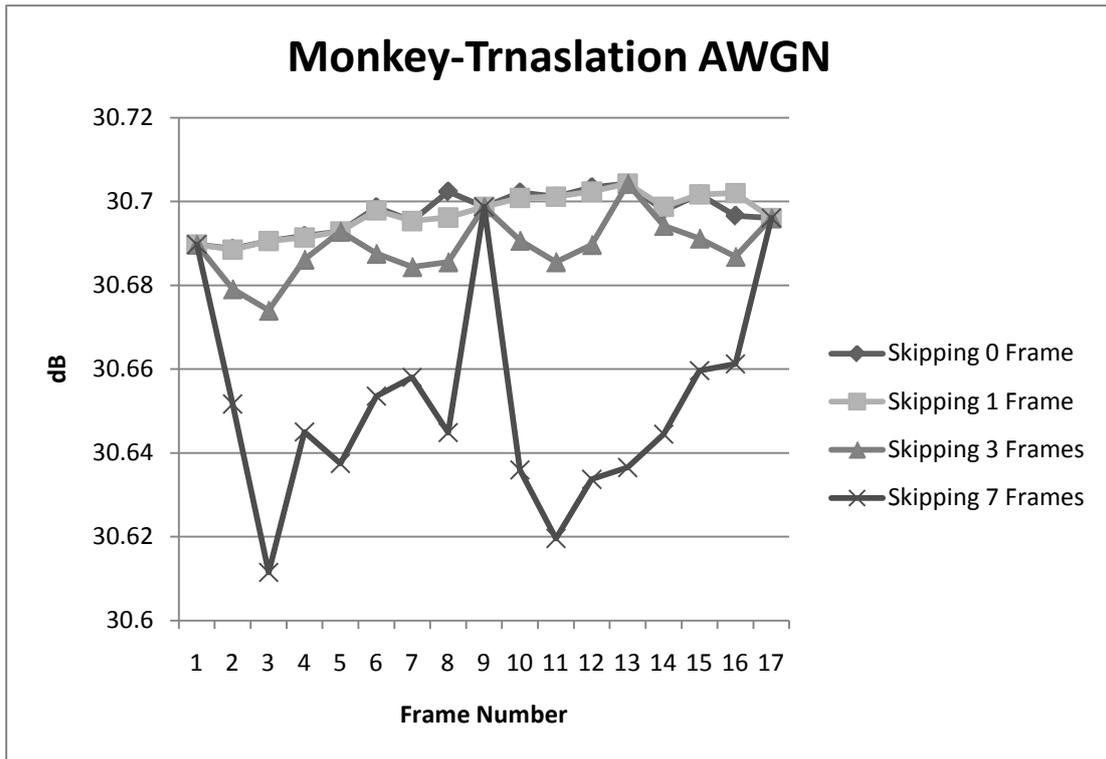


圖 5-32 連續影像 Monkey-Translation 加入 AWGN 後，在略過不同數量 frame 的情況下各張合成影像品質

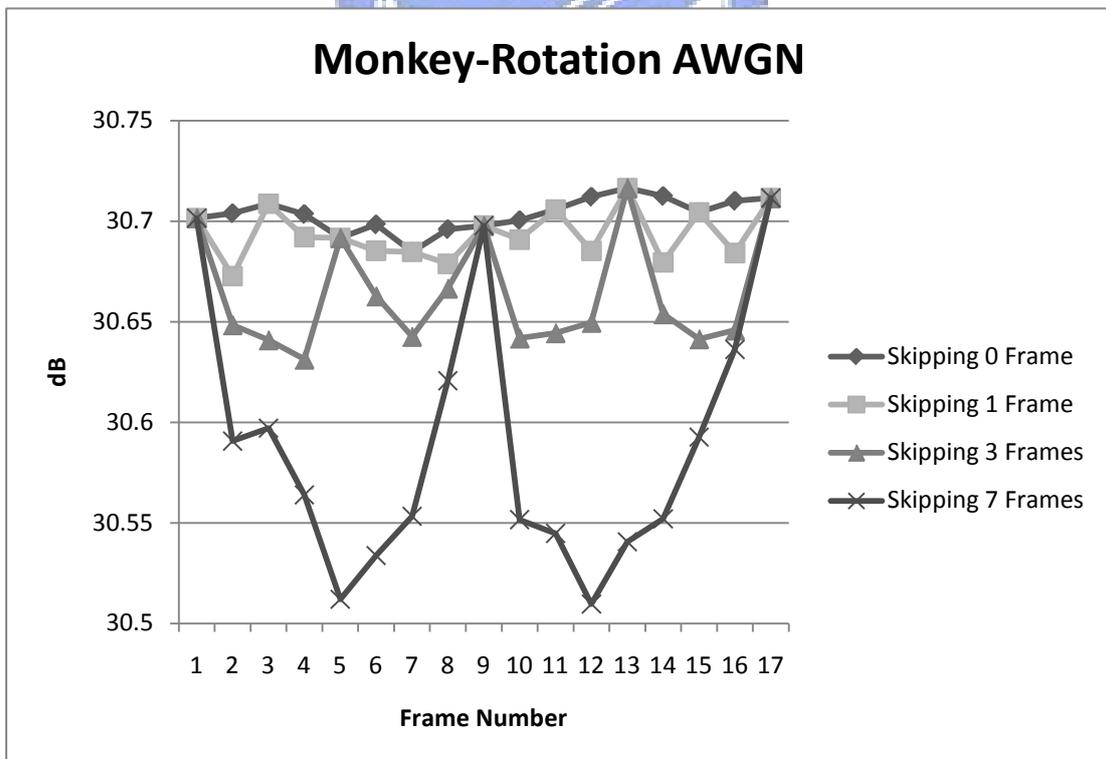


圖 5-33 連續影像 Monkey-Rotation 加入 AWGN 後，在略過不同數量 frame 的情況下各張合成影像品質

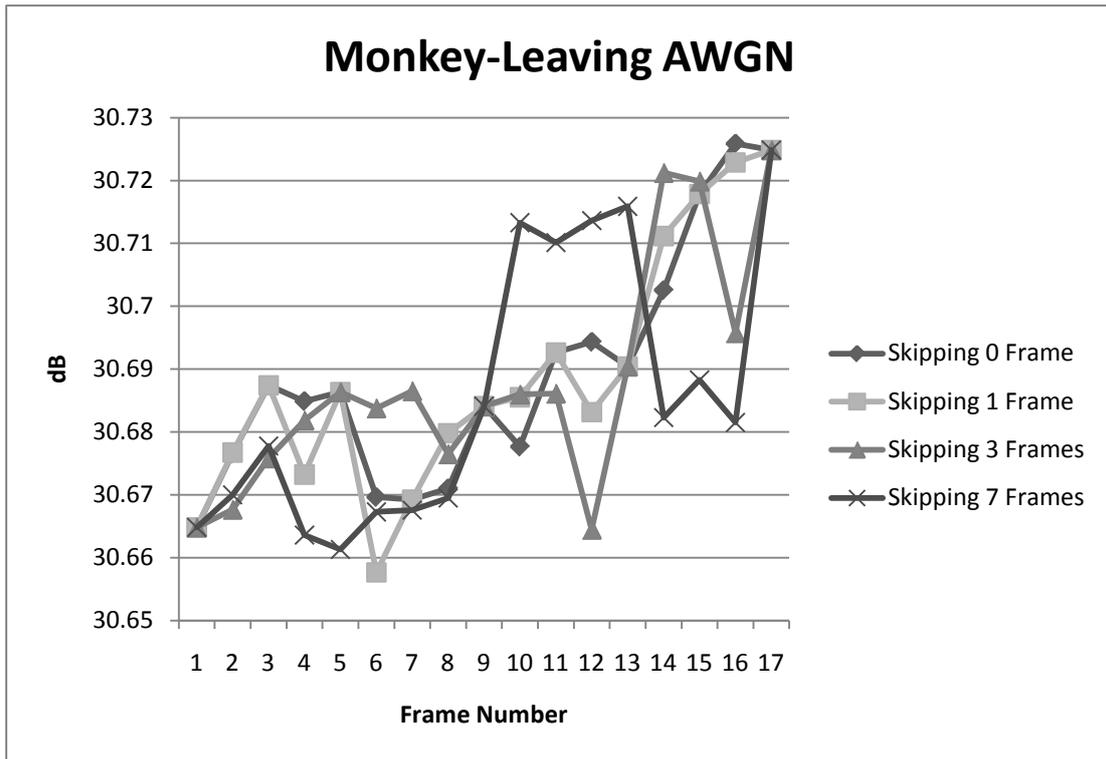


圖 5-34 連續影像Monkey-Leaving加入AWGN後，在略過不同數量frame的情況下
的各張合成影像品質

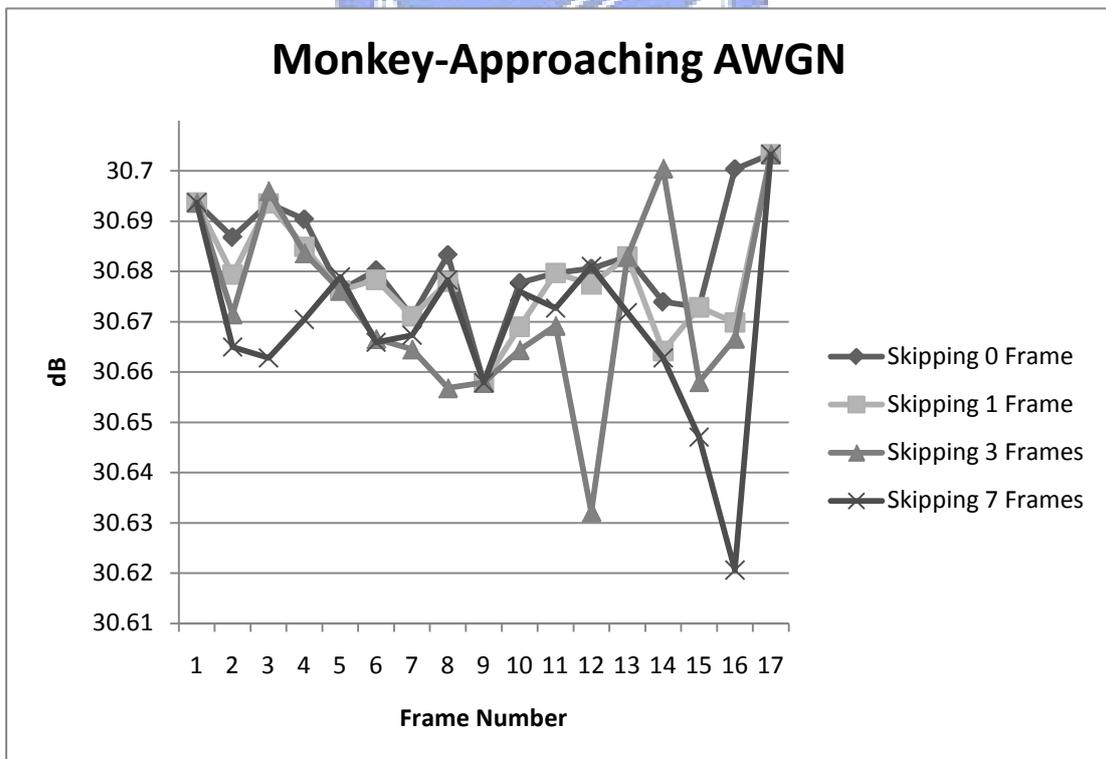


圖 5-35 連續影像 Monkey-Approaching 加入 AWGN 後，在略過不同數量 frame
的情況下的各張合成影像品質

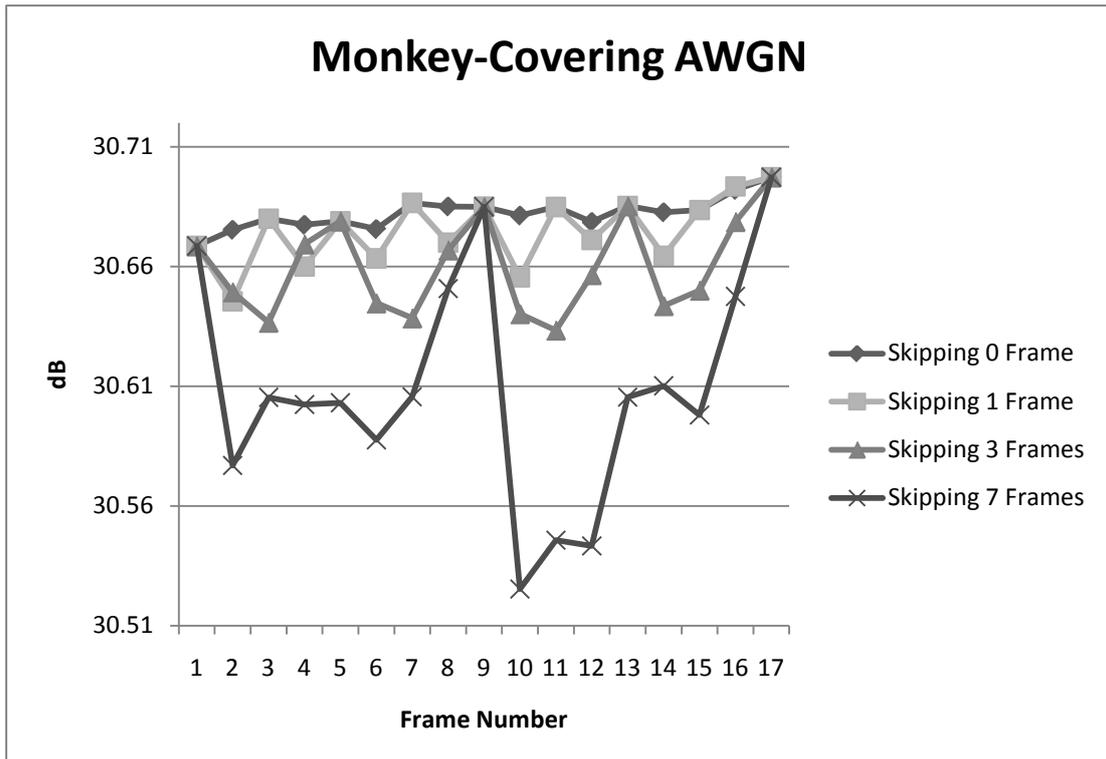


圖 5-36 連續影像 Monkey-Covering 加入 AWGN 後，在略過不同數量 frame 的情況下，在各張合成影像品質

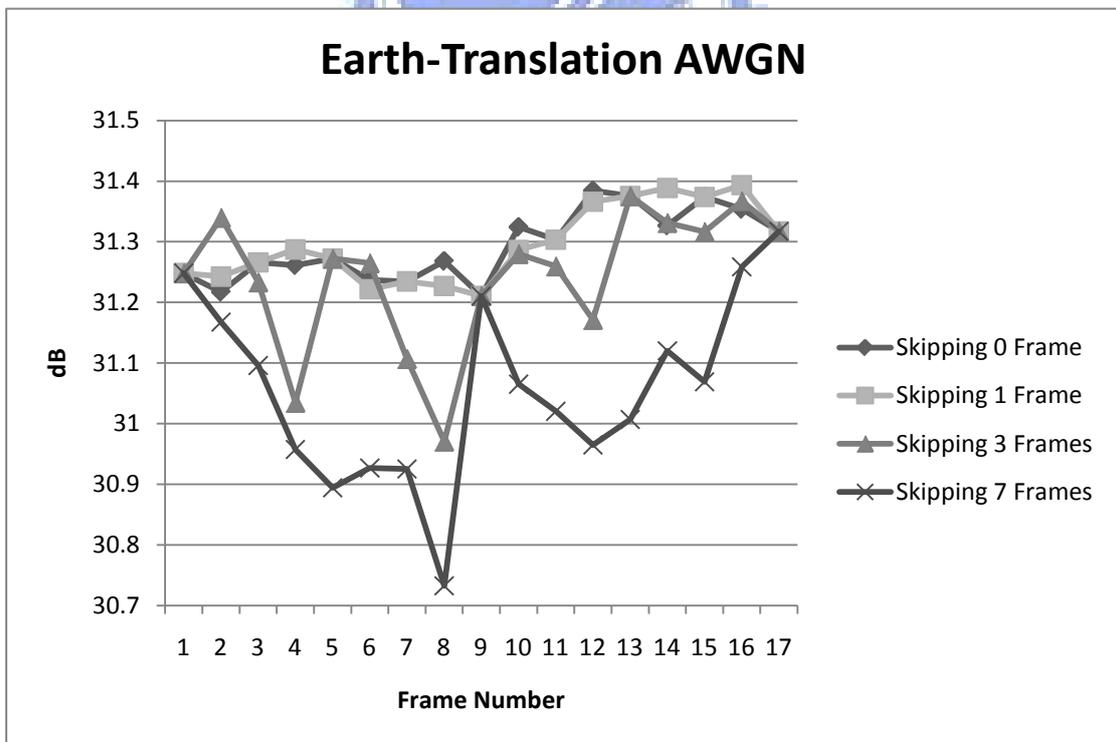


圖 5-37 連續影像 Earth-Translation 加入 AWGN 後，在略過不同數量 frame 的情況下，在各張合成影像品質

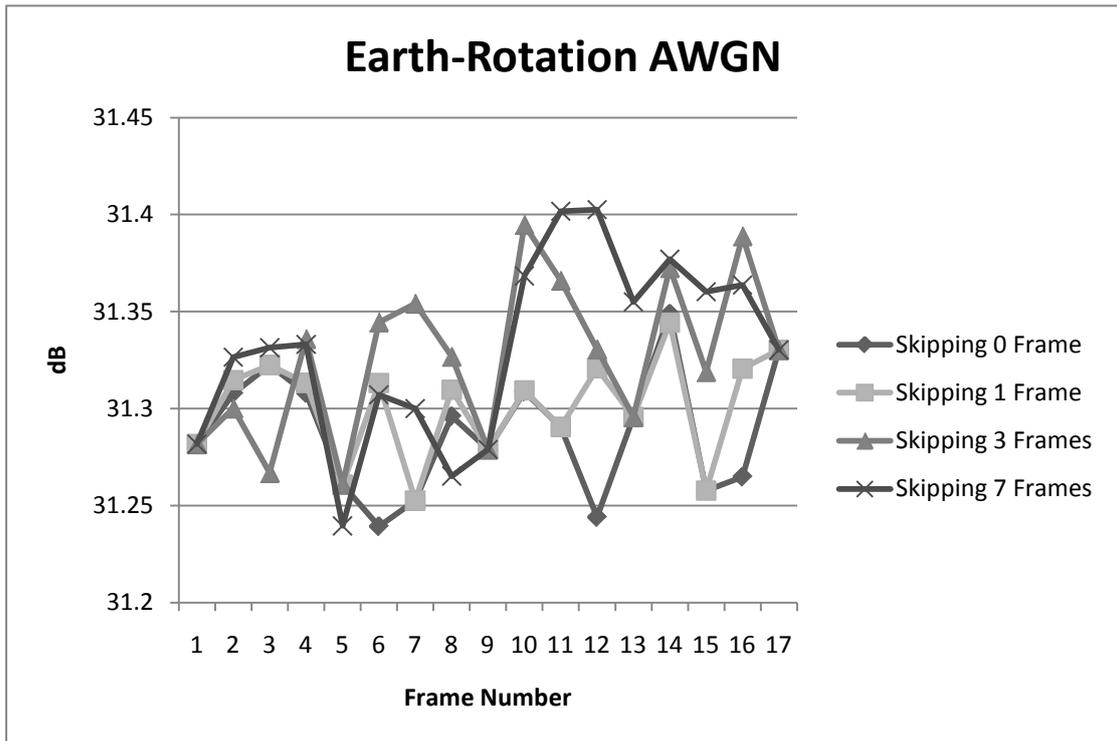


圖 5-38 連續影像 Earth-Rotation 加入 AWGN 後，在略過不同數量 frame 的情況下的各張合成影像品質

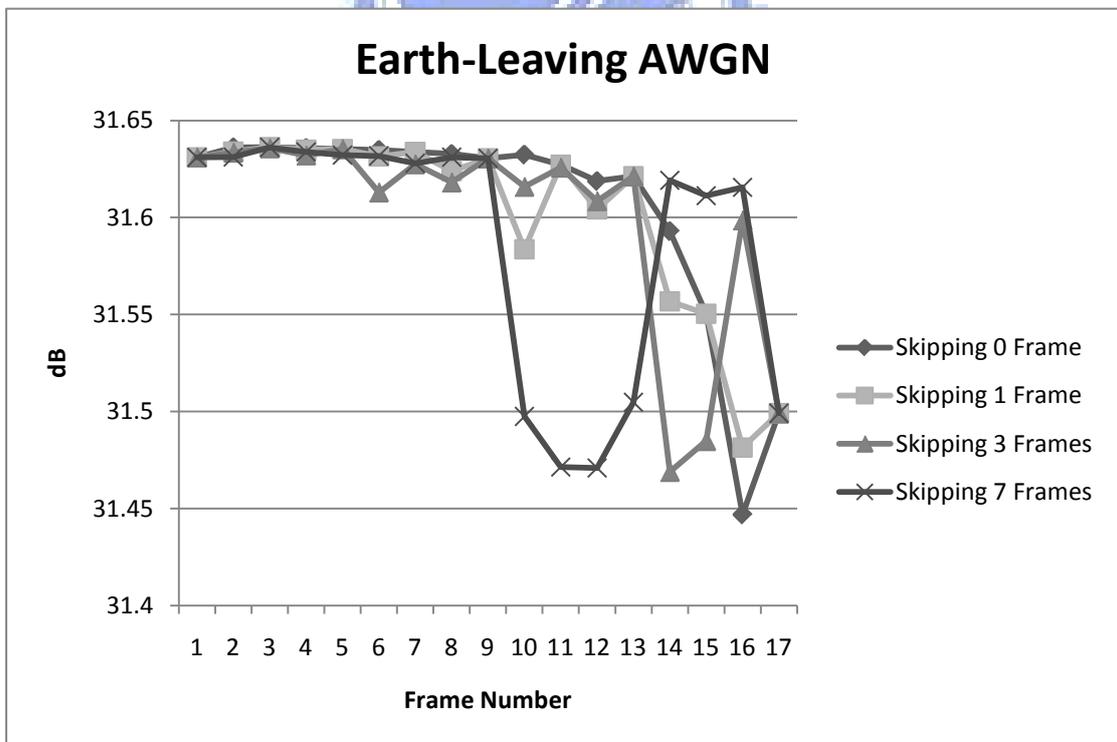


圖 5-39 連續影像 Earth-Leaving 加入 AWGN 後，在略過不同數量 frame 的情況下各張合成影像品質

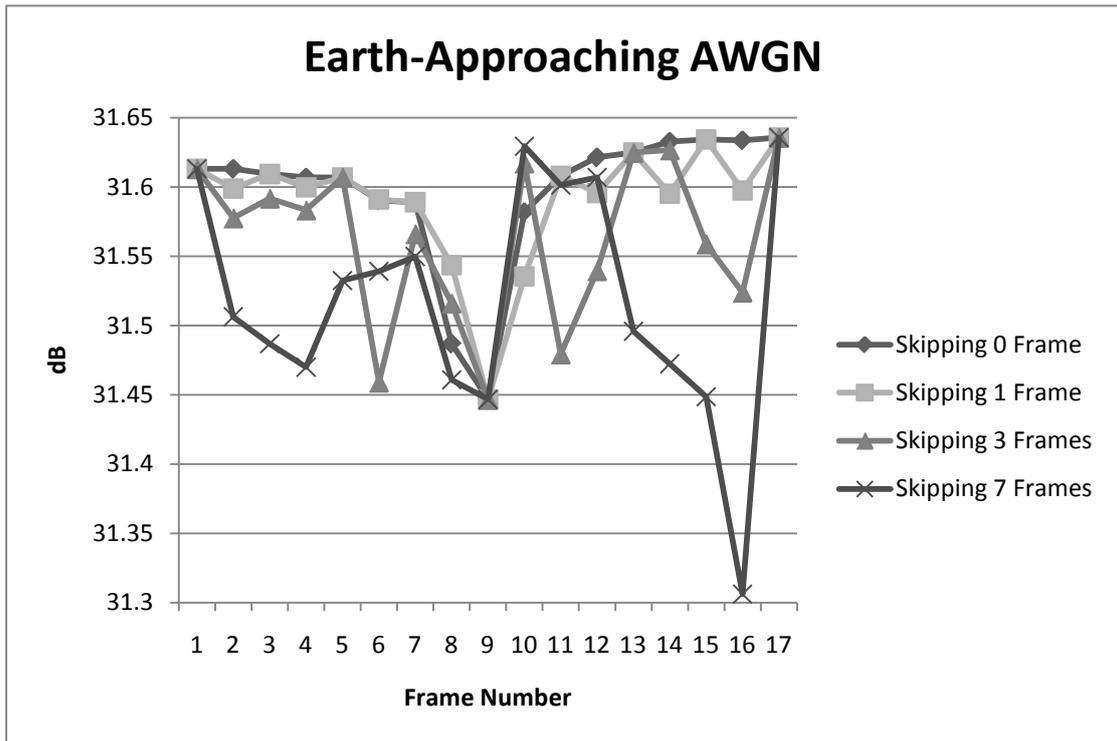


圖 5-40 連續影像 Earth-Approaching 加入 AWGN 後，在略過不同數量 frame 的情況下，各張合成影像品質



5.3.3 實驗討論

觀察實驗結果可以發現，參考影像間距越遠但合成結果越的好情況和原始影像是否理想關係不大，而應該和物體運動方式有關。圖 5-41(a)(b)分別為在測試資料 Monkey-Leaving 中使用完整視差估測方式得到的視差圖 D_{map0} ，與利用間隔三張 frame 的移動向量所得視差圖 D_{map3} ，5-41(c)(d)則是各自利用 D_{map0} 及 D_{map3} 各自合成的虛擬視點影像。在視差圖 D_{map0} 中能判斷出物體本身前後距離；而在視差圖 D_{map3} 中，該物體僅有一種視差值。可見使用完整視差估測方式所得到的視差結果是較為精確的。但比較兩張合成影像和實際影像之間的差異圖，我們發現參考 D_{map0} 所合成的結果，發生嚴重錯誤的區域其位置和視差值較高的位置相同，如圖 5-41(e)。而參考精確度較差的 D_{map3} 合成出來的結果卻沒有出現嚴重的錯誤，如圖 5-42(f)。因此，我們認為這可能和視差圖 D_{map0} 中視差值較高的區域有關。

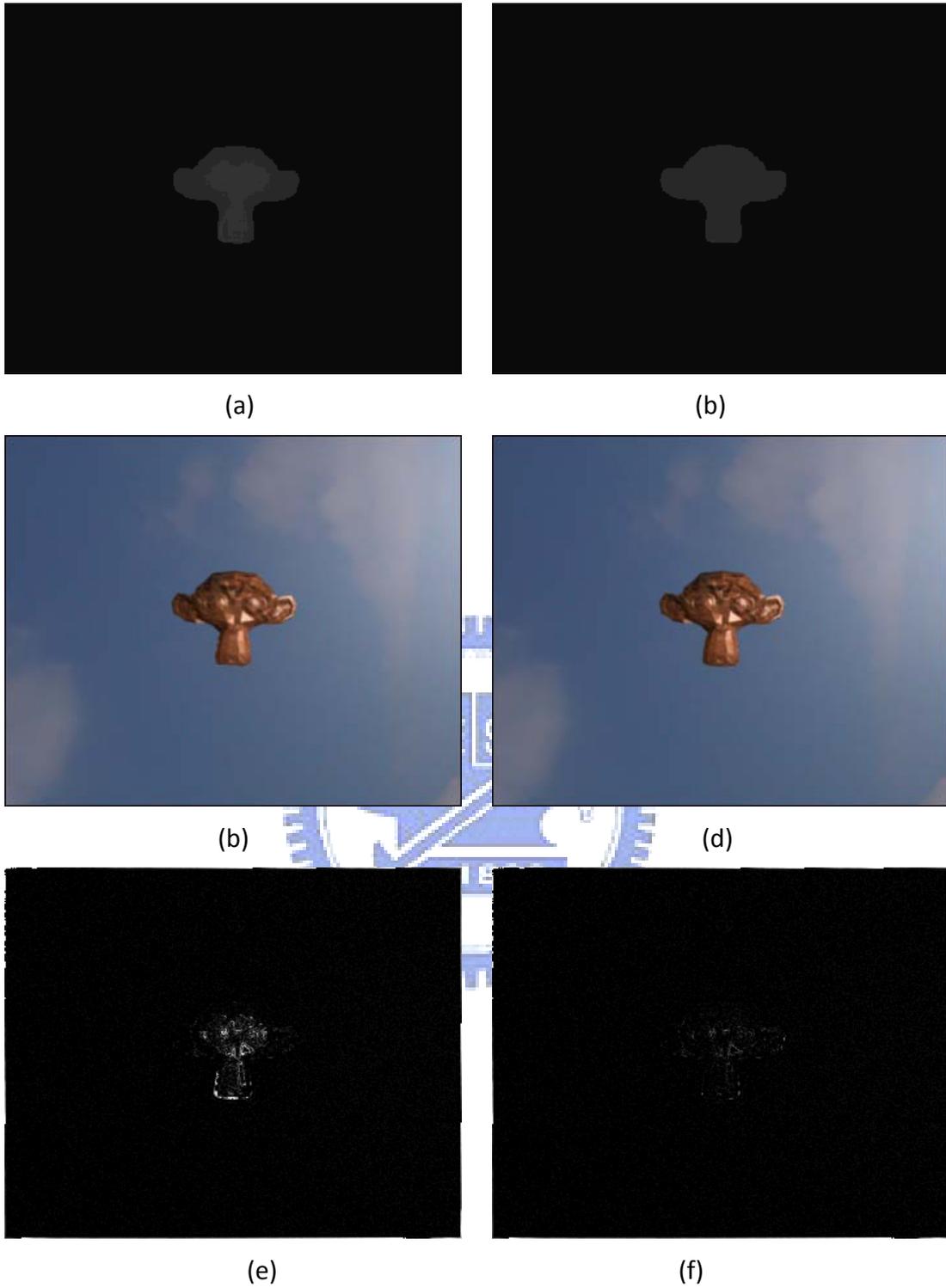


圖 5-41 (a) D_{map0} (b) D_{map3} (c)使用 D_{map0} 合成的影像 (d)使用 D_{map3} 合成的影像
 (e)使用 D_{map0} 合成的影像和實際影像的誤差 (f)使用 D_{map3} 合成的影像和實際影像
 的誤差

在圖 5-42 中，我們同時比較以完整視差估測方式所得到兩張相鄰的視差圖。我們可以發現在這兩張視差圖上，視差值最高的區域在外觀上有著明顯的差異，尤其是圖中標記的區域。這是因為實際上的視差值可能會有小數值，但在實驗中，我們將視差值都視為整數，因此無法完整地描述物體前後移動時細微的連續變化。這會造成最後利用視差值來對應內插位置時出現錯誤，例如圖 5-43 中，可以發現在物體上出現許多因對應位置視的視差值彼此不合，而無法合成的區域(圖中黑色部份)，需要使用鄰近區域以內插方式來填補。

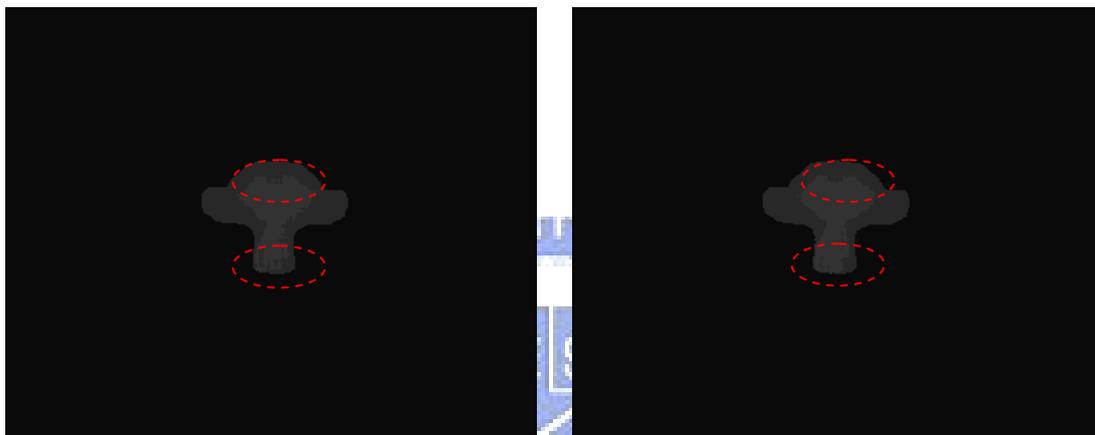


圖 5-42 使用完整視差估測方式得到的兩張相鄰影像視差

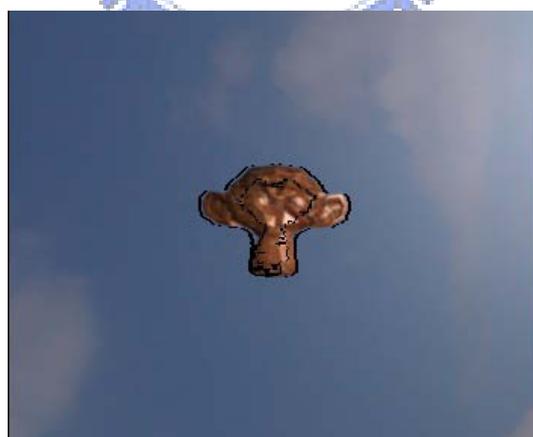


圖 5-43 利用圖 5-42 所合成出來的影像

而在圖 5-44 中是使用參考影像相距三張 frame 的移動向量所得到兩張相鄰影像的視差圖。雖然這兩張視差圖和實際的視差情形差距較大，但是兩張視差圖

中的物體外觀整在體上顯得比圖 5-42 中的結果更為一致，這反而能夠降低最後影像合成時對應位置視差值不合的情況。其結果如圖 5-45，可以看出在物體上不會出現無法合成的區域，因此影像品質反而比圖 5-43 來的高。



圖 5-44 參考移動向量資訊得到的兩張相鄰影像視差

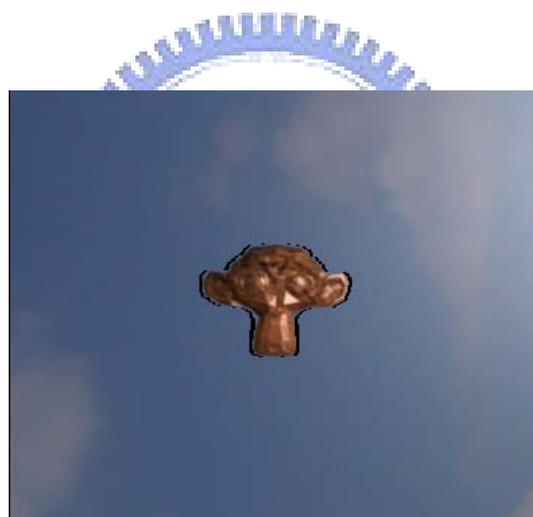


圖 5-45 利用圖 5-44 所合成出來的影像

從以上討論中，我們發現視差圖精確程度和最後合成影像的品質彼此關係不一定是絕對的。在視差估測中，要精確地計算物體邊緣周遭的視差值是很困難的，因為在物體邊緣部份很容易受到遮蔽效應的影響，這使得視差圖中的物體會常常和實際影像中的外觀有所出入。對於一個在判斷精確的視差圖上的物體，視差圖通常能夠表現出該物體本身的前後距離。但若相鄰的視差圖之間對於該物體的外觀不一致，會發生在合成虛擬影像時應該互相對應的位置但是視差值卻不同的情

況，導致該物體上出現無法合成的區域，進而將低了影像品質；反之，較不精確的視差圖之間對於該物體的外觀能有一致或相近的描述，則最後得合成影像仍然會有不錯的品質。因此，在利用視差圖合成虛擬視點影像時，我們要追求的應該是能夠穩定描述影像中物體資訊的視差圖，而不是最接近現實情況的視差圖。



第六章 結論與未來工作

6.1 結論

多重視點視訊已逐漸成為多媒體領域中的研究主項，而自由視點視訊是其中的重要議題。故本論文以其為研究對象，將現有的影像擷取設備由一維攝影機陣列拓展為二維攝影機陣列，並利用空間和時間上的相關性來提昇虛擬視點影像的合成結果。

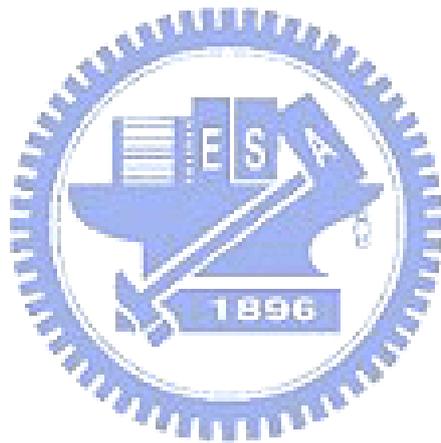
在空間相關性上，我們使用四張由不同搜尋方向得到視差圖來修正因遮蔽效應引起的 occlusion 問題，並搭配一個與對應點搜尋方向相關的投票機制來提昇視差圖的精確度。由於二維攝影機陣列中的攝影機數量非常龐大，為了降低整體系統計算視差圖的次數，我們提出了棋盤式視差估測。此種視差估測演算法是利用相鄰視差圖之間的平移特性，以四張鄰近的主視差圖來合成被其包圍的副視差圖。棋盤式視差估測使得計算視差圖的次數降為原來的一半，而且合成出來的副視差圖仍然保有高度的精確度。

在時間相關性上，我們使用移動向量來降低視差估測所需要的時間。為了取得更準確的移動向量，我們採用全域搜尋移動估測，並加入已知的視差圖作為參考。由實驗顯示，視差圖資訊和原始影像資訊在移動估測中的權重以 1:0.1 為一個較理想的比例。而比較合成一張影像所需的時間和略過的 frame 數量之間的關係，以略過三張 frame 的結果最佳，其計算時間可以降至原本的三分之一；當略過的 frame 數量超過三張時，則會因為移動估測的搜尋範圍太大而無法有效降低計算時間，甚至會出現計算時間增加的情形。隨著略過的 frame 數量增加，合成出來的影像品質會逐漸呈現遞減的趨勢，但是變化不大。在實驗中最差的情況下，略過一張 frame 僅使得 PSNR 下降 0.02dB，而略過三張 frame 時 PSNR 只下降 0.05dB。同時考量時間效益以及影像品質，略過三張 frame 是一個較為理想的選擇。

6.2 未來工作

由於本論文中使用電腦合成影像作為測試資料，因此並未考量在現實攝影機陣列中容易遇到的問題，例如攝影機校正、不同攝影機擷取影像的亮度不同等等。為了能夠更接近實際應用，未來實驗所用的測試影像應該要由實際攝影機陣列取得。

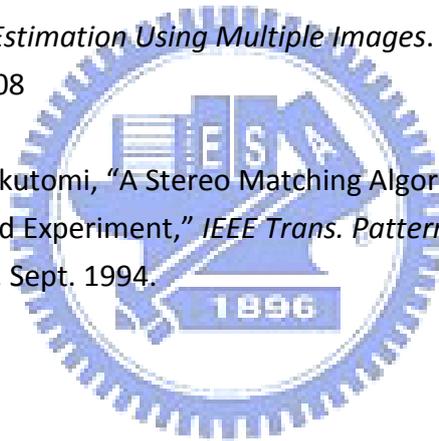
為了取得精確的移動向量，我們在實驗中採用了全域搜尋移動估測。但此方法搜尋範圍太大，導致無法有效地降低整體時間。在視訊壓縮技術成熟的今日，我們或許可以測試各種不同的移動估測方式，來找出一個兼顧影像品質和時間效能的演算法。



參考文獻

- [1] *Applications and Requirements for 3DAV*, ISO/IEC JTC1/SC29/WG11 N5877, July 2003.
- [2] *Report on 3DAV Exploration*, ISO/IEC JTC1/SC29/WG11 N5878, July 2003.
- [3] M. Tanimoto, "FTV (free viewpoint television) creating ray-based image engineering," *IEEE Int. Conf. Image Process.*, Genova, Italy, 2005.
- [4] M. Tanimoto, "FTV (free viewpoint television) for 3-D scene reproduction and creation," *IEEE Conf. Comput. Vision Pattern Recog.*, New York, NY, 2006.
- [5] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. ACM Annu. Computer Graphics Conf.*, Aug. 1996, pp. 11–20.
- [6] M. Tanimoto, "Overview of free viewpoint television," *Signal Processing: Image Communication*, vol. 21, no. 6, pp.454-461, July 2006.
- [7] R. C. Bolles and H. H. Baker, "Epipolar-plane image analysis: A technique for analyzing motion sequences," in *Proc. IEEE 3rd Workshop Computer Vision: Representation and Control*, Bellaire, Oct. 1985, pp. 168-178.
- [8] Baker, H. and R. Bolles, "Generalizing epipolar-plane image analysis on the spatiotemporal surface," in *DARPA Image Understanding Workshop*, Cambridge, MA. April 6-8, pp. 1022-1030.
- [9] T. Naemura and H. Harashima, "Real-Time Video-Based rendering for augmented spatial communication," *Proc. VCIP*, vol. 3653, SPIE Press, Bellingham, Wash., 1999, pp. 620-631.
- [10] N. Grammalidis and M. G. Strintzis, "Disparity and occlusion estimation in mutiocular systems and their coding for the communication of multiview image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 328–344, June 1998.

- [11] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 2, pp. 139-154, March, 1985.
- [12] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int'l J. Computer Vision*, vol. 47, pp. 7-42, Apr. 2002.
- [13] G. V. Meerbergen, M. Vergauwen, M. Pollefeys, and L. V. Gool, "A hierarchical symmetric stereo algorithm using dynamic programming," *Int'l J. Computer Vision*, 47(1/2/3):275–285, April-June 2002.
- [14] R. Yang and M. Pollefeys. "Multi-resolution real-time stereo on commodity graphics hardware," *Proc. of CVPR*, v1. pp. 211-220. 2003.
- [15] S. T. Hsu, *Disparity Estimation Using Multiple Images*. National Chaio Tung University, M.S., 2008
- [16] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, Sept. 1994.



自傳

徐崇毓，民國七十三年出生於台中縣豐原市。民國九十六年畢業於國立中正大學電機系，同年進入國立交通大學電子所攻讀碩士學位。指導教授為杭學鳴博士，研究方向為多重視點視訊之研究。於民國九十七年取得碩士學位。

