行政院國家科學委員會專題研究計畫 期末報告

快速又準確的立體影像深度估測法

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 101-2221-E-216-034-
執 行 期 間 ：101 年 08 月 01 日至 102 年 10 月 31 日
執 行 單 位 ：中華大學資訊工程學系

計 畫 主 持 人：鄭芳炫

計畫參與人員：碩士班研究生-兼任助理人員：林建邑
碩士班研究生-兼任助理人員：曾楞竹
碩士班研究生-兼任助理人員：黃冠瑜

報 告 附 件 ：出席國際會議研究心得報告及發表論文

公 開 資 訊 ：本計畫可公開查詢

中 華 民 國 102 年 12 月 10 日

中 文 摘 要 ： 本研究主要是利用線段比對方式來找出一對左右眼的影像中物件的視差，由此獲得深度資訊。我們線段比對的方法分為三個步驟，第一步驟先偵測左影像的邊界，由此可得到影像中的每個線段，第二步驟計算左右影像中每個像素點顏色的差值當成本，第三步驟是利用第二步驟的成本以及線段比對的方式來找出左影像中在右影像的對應線段，便可得知每個像素點的視差，由此視差便可轉換為深度影像。
不過只以線段比對的方法會造成被遮蔽區域的大量錯誤。而我們用了三個偵測的方法來偵測遮蔽區域的錯誤，並且加以改善。第一個改善方法主要偵測的區域為視差差異劇烈的區域，此區域因為前景與背景的視差差異大，所以被遮蔽的區域大，所以錯誤率會比較高。第二個偵測的方法利用九個區塊來過濾出一些可能錯誤的像素點。第三個主要偵測的區域為視差差異平緩的區域，此區域前景與背景差異小，被遮蔽的區域小，但仍然會有錯誤的可能。找出錯誤的區域後我們找出錯誤點附近顏色相近的像素點，統計出顏色相近的像素點視差值，看哪個視差值的像素點多，我們便將此視差值取代錯誤像素點的視差值。由實驗可以看出我們改善的方法可以大量的改善被遮蔽區域的錯誤，以及由實驗中證實我們的方法可以即時且準確的評估深度資訊。

中文關鍵詞： 深度圖，立體影像，立體比對，線段

英 文 摘 要 ： This paper proposes a line segment method to estimate the depth information from a pair of rectified images. This method can achieve real-time and high quality stereo-matching. The first step uses a simple edge detection to find out the line segments in the reference image. The second step is to calculate the color difference of each pixel from a pair of rectified images, and the difference is saved to a cost matrix. The last step is to find out the minimum difference of each line segment as the corresponding line from the cost matrix. After finding the corresponding line, it can discover the disparity of each line, and use the disparity to convert depth. Unfortunately the line segments matching method is susceptible to cause error in occluded areas, so we propose three methods to refine the depth map. The first step is to detect areas with large variation of disparity value； the difference between the foreground and background disparity values is large

in occluded area. The larger the occluded region is, the higher the error rate. The second step is to use nine blocks to find the possible pixels with wrong depth. The last step is to detect areas with mild variation of disparity value. The difference between the foreground and background disparity value is small in these area, but there may still occur errors in these occluded area. We find out these error areas, and then count the disparity values of the similar color pixels around these area. The disparity value that has the most pixels replaces the wrong disparity values. From the experiments, it is proved that the proposed three refined methods can successfully correct errors in occluded areas and can accurately estimate the depth information in real-time.

英文關鍵詞： depth map, stereoscopic image, stereo matching, line segments

目錄

目錄.....I
摘要.....II
ABSTRACT.....II

1. 前言.....1

2. 研究目的.....1

3. 文獻探討.....2
3.1 深度資訊取得方法.....2
3.2 動態規劃方法.....5
3.3 其他比對方法.....6

4. 研究方法.....7
4.1 利用邊緣偵測找出線段.....9
4.2 計算成本矩陣.....10
4.3 找出對應點.....11
4.4 偵測遮蔽區域並且改善.....15
4.4.1 偵測被遮蔽區域視差差異劇烈的地方.....15
4.4.2 改善的方法.....17
4.4.3 偵測一些少數的錯誤視差值.....19
4.4.4 偵測被遮蔽區域視差差異較平緩的區域.....20

5. 結果與討論.....20

參考文獻.....28

摘要

本研究主要是利用線段比對方式來找出一對左右眼的影像中物件的視差，由此獲得深度資訊。我們線段比對的方法分為三個步驟，第一步驟先偵測左影像的邊界，由此可得到影像中的每個線段，第二步驟計算左右影像中每個像素點顏色的差值當成本，第三步驟是利用第二步驟的成本以及線段比對的方式來找出左影像中在右影像的對應線段，便可得知每個像素點的視差，由此視差便可轉換為深度影像。

不過只以線段比對的方法會造成被遮蔽區域的大量錯誤。而我們用了三個偵測的方法來偵測遮蔽區域的錯誤，並且加以改善。第一個改善方法主要偵測的區域為視差差異劇烈的區域，此區域因為前景與背景的視差差異大，所以被遮蔽的區域大，所以錯誤率會比較高。第二個偵測的方法利用九個區塊來過濾出一些可能錯誤的像素點。第三個主要偵測的區域為視差差異平緩的區域，此區域前景與背景差異小，被遮蔽的區域小，但仍然會有錯誤的可能。找出錯誤的區域後我們找出錯誤點附近顏色相近的像素點，統計出顏色相近的像素點視差值，看哪個視差值的像素點多，我們便將此視差值取代錯誤像素點的視差值。由實驗可以看出我們改善的方法可以大量的改善被遮蔽區域的錯誤，以及由實驗中證實我們的方法可以即時且準確的評估深度資訊。

關鍵字:深度圖，立體影像，立體比對，線段

ABSTRACT

This paper proposes a line segment method to estimate the depth information from a pair of rectified images. This method can achieve real-time and high quality stereo-matching. The first step uses a simple edge detection to find out the line segments in the reference image. The second step is to calculate the color difference of each pixel from a pair of rectified images, and the difference is saved to a cost matrix. The last step is to find out the minimum difference of each line segment as the corresponding line from the cost matrix. After finding the corresponding line, it can discover the disparity of each line, and use the disparity to convert depth.

Unfortunately the line segments matching method is susceptible to cause error in occluded areas, so we propose three methods to refine the depth map. The first step is to detect areas with large variation of disparity value; the difference between the foreground and background disparity values is large in occluded area. The larger the occluded region is, the higher the error rate. The second step is to use nine blocks to find the possible pixels with wrong depth. The last step is to detect areas with mild variation of disparity value. The difference between the foreground and background disparity value is small in these area, but there may still occur errors in these occluded area. We find out these error areas, and then count the disparity values of the similar color pixels around these area. The disparity value that has the most pixels replaces the wrong disparity values. From the experiments, it is proved that the proposed three refined methods can successfully correct errors in occluded areas and can accurately estimate the depth information in real-time.

Keywords: depth map, stereoscopic image, stereo matching, line segments

1. 前言

在阿凡達 3D 電影造成全球的風潮後，2010 年已正是成為 3D 元年，3D 電影已成為主流。除了電影之外也看到許多 3D 影像的產品，像是 3D 視訊會議的應用和手機及相機也有 3D 顯示的功能以及電玩方面也有 3D 功能，平面影像已經漸漸不符合大眾所需求的。人類之所以能看出立體的效果主要是因為我們在觀看一個物體時，由於人類的眼睛兩眼的相距六公分左右，所以此物體在左右眼中會有一些位移，此位移就稱之為視差。也因為雙眼的視差，讓我們產生了立體深度的感覺。

2. 研究目的

目前 3D 拍攝技術主要可以分成兩種，第一種為陣列式攝影，此原理是同時利用兩台以上的攝影機，模擬人眼視覺拍攝左右眼的影像，兩台攝影機距離必須間隔 6~7 公分(雙眼之間的距離)，如圖 1。同時透過 3D 顯示器分別顯示左右眼影像畫面，及可有立體的效果。不過由於兩台攝影機同時拍攝，所以必須事先做校正，讓兩台攝影機拍攝的畫面高度，大小，角度及光影必需一致。所以事前的校正工作必須做好，否則無法觀看立體效果。現在有些廠商則開發出本身具有兩個鏡頭的攝影機，分別捕捉左右眼影像，便可解決校正上的麻煩。

第二種為深度攝影，這方式是利用傳統攝影機另外還搭配了深度攝影機拍攝，除了拍攝 2D 影像之外，還有深度攝影拍攝的深度圖。深度攝影機是透過紅外線碰到物體反射時間來判斷物體與攝影機之間的距離。此方式是目前最常用的立體影像格式之一，稱為 2D+Depth。不過這個儀器較為昂貴，還無法普及化。

圖1 左為陣列式攝影，右為深度攝影機。

另外還有一種方法是不需要靠儀器的方法，只需要左右眼影像，利用兩眼的影像來估算出視差值(位移量)，便可計算出深度資訊，產生深度圖，此方法也是我們研究的目的。我們可以利用估算出來的深度圖以及利用原始影像，去估算出其他不同視角的影像出來，將不同視角的影像配合顯示器的像素排列組合差排成一張後，如圖 2，將此影像放在 3D 顯示器上播放，便可看到立體效果。不過此方式的重點在於如何估算出好的深度圖，如果產生出好的深度圖，才能產生出較好的影像(不同視角的影像)。下面我們便開始探討如何估算出深度資訊。

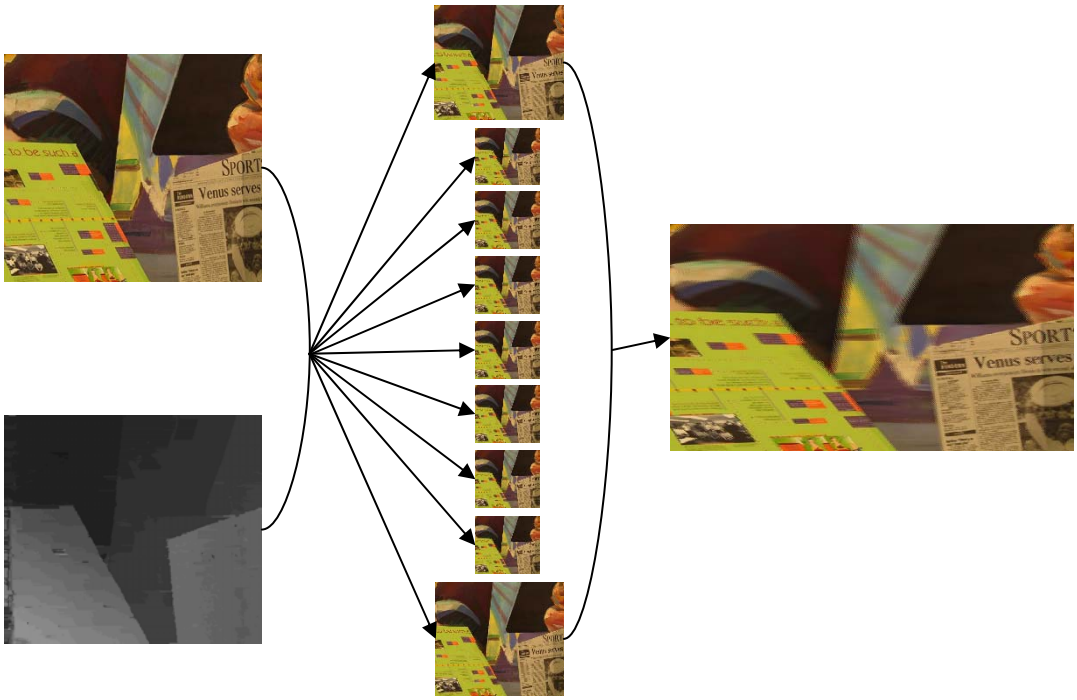


圖2 2D影像轉換成3D影像的過程。

3. 文獻探討

3.1 深度資訊取得方法

立體影像是必須由雙眼才能看得到具有深度及層次感的影像，它必須是要有兩張或兩張以上的影像，並具有視差的兩張影像才能觀看出立體效果。這兩張影像就是模擬人眼的左右眼睛所製成的影像，因此要觀看到具有深度的立體影像就必須要運用到雙眼不可。何謂深度以及何謂視差呢?下面我們依序探討。

深度資訊取得的方法有很多種像是深度攝影機或是深度感應器(kinect)，如圖 3，但是這種都必須靠器材來取得，外出並不方便攜帶使用。目前大眾所使用的攝影工具還是以一般的數位相機或是數位攝影機為主，所以只能夠得到一般的平面影像，而我們用另外一種方式由 2D 影像計算得到深度資訊。而本篇論文主要討論的就是這種由 2D 影像記算得到深度資訊方法。深度資訊就是我們必須取得場景中各點相對於攝影機的距離，場景中各個點相對於攝影機的距離可以用深度圖(depth map)來表示，深度圖為一個灰階的影像，我們設定 0~255 的灰階值表示，灰階值越接近 255(白色)的像素點表示此像素點在場景中距離攝影機越近；反之灰階值越接近 0(黑色)的像素點表示此像素點在場景中距離攝影機越遠。

圖3 深度感應器。

何謂視差呢?因為人眼觀看景物時可分為遠、中、近三個層次，人的眼球會隨者景物的遠近而自動調整到一個最舒適的視覺角度來觀看，當左右眼各有不同的視角時，就會產生出視差。視差又因景物的遠、中、近的不同，可分成正視差、零視差、及負視差，如圖 4 所示。(1)正視差:兩眼焦點視線在螢光幕前並沒有任何交叉，則其影像將呈現在螢光幕後。(2)零視差：兩眼焦點視線的交叉點落在螢光幕上，則其影像將會呈現在螢光幕上。(3)負視差：兩眼焦點視線在螢光幕前有交叉情況，則其影像將會呈現在螢光幕前的交叉點上。

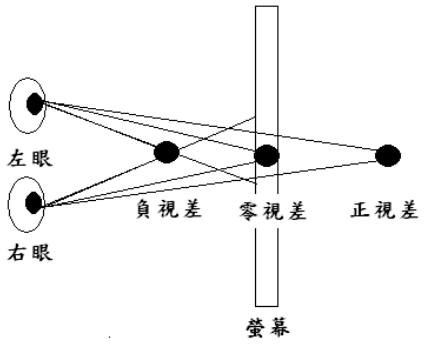


圖4 正視差、零視差、及負視差示意圖。

2D 影像比對得到深度資訊的方法已經研究了很多年了，而技術也越來越純熟，此技術主要是將一對左影像及右影像來做比對，得知左右影像中物件的視差(disparity)來估算出深度值(depth)。利用左右影像來估算出每個物件的視差值(disparity)後，再將視差值正規化到 0~255 的灰階值，此灰階值變是深度值了。Middlebury[1]的網站上，如圖 5，這網站提供了很多的測試圖以及提供了正確的深度圖(ground truth)，不過它提供的測試圖只有負視差及零視差的情況，如圖 6，左影像中的物件出現在右影像中會往左位移，所以比對上左影像中的像素點去右影像找對應點時，只須往左邊的方向找尋，不需要往右邊找尋(因為沒有正視差的情況)，正視差的情況是左影像中的物件出現在右影像中會往右位移。而零視差的情況是左影像的物件出現在右影像中不會有位移的情況。另外 Middlebury 網站提供了另一個功能，可由你的演算法來估測出這四組測試圖(Tsukuba、Venus、Teddy、Cones)的深度圖後，上傳這四組深度圖，網站可以幫你評估出你的深度圖的正確率，並且幫你與其他方法做正確率排名，由此排名，你可知道哪些方法比你的好。網站上已經有 124 的方法作排名，並且許多方法都

有公開他的研究論文讓大家做參考，讓大家互相研究比較各個方法的優缺點，讓此技術更上一層。目前第一名的方法評估出來的準確率已達到 96%，而最後一名有 80%的準確度。

Stereo Evaluation • Datasets • Code • Submit

Middlebury Stereo Evaluation - Version 2

New features and main differences to version 1.
Submit and evaluate your own results.

☐ Open a new window for each link

Error Threshold = 1 (Error Threshold...) — Sort by nonocc ▼ — Sort by all ▼ — Sort by disc ▼

Algorithm	Avg. Rank ▼	Tsukuba ground truth nonocc	Tsukuba all ▼	Tsukuba disc	Venus ground truth nonocc	Venus all ▼	Venus disc	Teddy ground truth nonocc	Teddy all ▼	Teddy disc	Cones ground truth nonocc	Cones all ▼	Cones disc	Average percent of bad pixels (explanation)
ADCensus [94]	8.3	1.07 15	1.48 13	5.73 17	0.09 2	0.25 7	1.15 3	4.10 8	6.22 3	10.9 7	2.42 8	7.25 7	6.95 9	3.97
CoopRegion [41]	10.1	0.87 4	**1.16** 1	4.61 3	0.11 4	0.21 3	1.54 7	5.16 18	8.31 12	13.0 15	2.79 21	7.18 6	8.01 27	4.41
AdaptingBP [17]	10.2	1.11 19	1.37 7	5.79 19	0.10 3	0.21 4	1.44 5	4.22 10	7.06 7	11.8 11	2.48 10	7.92 14	7.32 13	4.23
RVbased [116]	13.4	0.95 9	1.42 11	4.98 8	0.11 6	0.29 11	**1.07** 1	5.98 25	11.6 36	15.4 32	2.35 6	7.61 8	6.81 8	4.88
RDP [102]	13.8	0.97 10	1.39 9	5.00 9	0.21 24	0.38 19	1.89 14	4.84 12	9.94 21	12.6 13	2.53 11	7.69 10	7.38 14	4.57
DoubleBP [35]	14.0	0.88 6	1.29 4	4.76 6	0.13 8	0.45 27	1.87 13	3.53 6	8.30 11	9.63 4	2.90 27	8.78 35	7.79 21	4.19
OutlierConf [42]	14.6	0.88 5	1.43 12	4.74 5	0.18 17	0.26 9	2.40 24	5.01 14	9.12 18	12.8 14	2.78 20	8.57 27	6.99 10	4.60
SubPixDoubleBP [30]	19.7	1.24 27	1.76 31	5.98 23	0.12 7	0.46 29	1.74 10	3.45 5	8.38 13	10.0 6	2.93 30	8.73 32	7.91 23	4.39
SurfaceStereo [79]	19.8	1.28 32	1.65 21	6.78 39	0.19 19	0.28 10	2.61 35	3.12 3	**5.10** 1	**8.65** 1	2.89 26	7.95 16	8.26 35	4.06
SubPixSearch [127]	20.0	2.04 72	2.48 62	6.40 32	0.14 11	0.40 23	1.74 10	4.00 7	6.39 4	11.0 9	2.24 3	6.87 4	6.50 3	4.18
WarpMat [55]	22.8	1.16 20	1.35 6	6.04 24	0.18 18	0.24 6	2.44 28	5.02 15	9.30 19	13.0 17	3.49 45	8.47 26	9.01 50	4.98
ObjectStereo [98]	24.3	1.22 26	1.62 17	6.36 30	0.59 68	0.69 48	4.61 69	4.13 9	7.59 8	11.2 10	**2.20** 1	6.99 5	**6.36** 1	4.46
HEBF [123]	26.7	1.10 18	1.38 8	5.74 18	0.22 25	0.33 16	2.41 26	6.54 45	11.8 40	15.2 29	2.78 19	9.28 47	8.10 29	5.41
PatchMatch [112]	27.5	2.09 74	2.33 58	9.31 73	0.21 23	0.39 21	2.62 36	2.99 2	8.16 9	9.62 3	2.47 9	7.80 11	7.11 11	4.59
CrossLMF [126]	28.7	2.46 82	2.78 70	6.26 28	0.27 38	0.38 20	2.15 18	5.50 21	10.6 25	14.2 19	2.34 4	7.82 12	6.80 7	5.13
BMVC-124 [132]	28.8	1.96 67	2.21 55	9.22 71	0.30 42	0.49 30	3.57 57	**2.88** 1	8.57 14	8.99 2	2.22 2	6.64 3	6.48 2	4.46
GC+SegmBorder [57]	29.4	1.47 48	1.82 33	7.86 61	0.19 20	0.31 12	2.44 28	4.25 11	5.55 2	10.9 8	4.99 86	**5.78** 1	8.66 43	4.52
Undr+OvrSeg [48]	30.0	1.89 64	2.22 57	7.22 49	0.11 5	0.22 5	1.34 4	6.51 41	9.98 22	16.4 45	2.92 29	8.00 17	7.90 22	5.39
CostFilter [95]	31.9	1.51 52	1.85 39	7.61 56	0.20 22	0.39 22	2.42 27	6.16 31	11.8 42	16.0 38	2.71 16	8.24 20	7.66 18	5.55
InfoPermeable [109]	32.5	1.06 14	1.53 14	5.64 14	0.32 44	0.88 61	4.15 62	5.60 22	13.0 58	14.5 22	2.65 15	9.16 45	7.69 19	5.51
FeatureGC [107]	33.5	1.08 16	1.59 16	5.82 21	**0.08** 1	**0.16** 1	1.11 2	7.17 60	8.25 10	18.5 75	4.33 74	9.40 53	11.1 73	5.72
NonLocalFilter [131]	34.1	1.47 48	1.85 39	7.88 63	0.25 35	0.42 24	2.60 34	6.01 27	11.6 38	14.3 21	2.87 25	8.45 25	8.10 30	5.48
AdaptOvrSegBP [33]	34.2	1.69 55	2.04 50	5.64 14	0.14 10	0.20 2	1.47 6	7.04 57	11.1 30	16.4 47	3.60 51	8.96 42	8.84 46	5.59
GlobalGCP [104]	34.3	0.87 3	2.54 65	4.69 4	0.16 15	0.53 34	2.22 22	6.44 38	11.5 35	16.2 41	3.59 49	9.49 56	8.95 49	5.60
CSM [120]	35.3	**0.82** 1	1.20 2	**4.39** 1	0.34 47	0.61 39	2.55 32	7.67 68	12.4 55	17.2 60	3.33 42	9.35 51	7.96 25	5.65

圖5 Middlebury網站。

圖6 Middlebury網站上測試圖(Tsukuba)，左圖為左眼影像，右圖為右眼影像，由於只有負視差的情況，所以左影像中的物件在右影像中都是往左位移。

3.2 動態規劃方法

然而在這麼多方法之中，其中以動態規劃(Dynamic programming)的方法比較快速，更容易實現即時的效果[2-10]，大部分都是以動態規劃這個觀念以及架構來做改善，使準確率提高，並且速度又快。動態規劃的主要觀念是把一個大問題分割成很多小問題，順序找出這些小問題的答案後，再利用這些小問題的答案來組合出大問題的答案。在立體視覺的應用中，就是把影像分割成一條一條的掃描線。每一條掃描線各別利用公式與圖 7 求出最適合的像素點差值後，再把每一條掃描線的結果組合起來，就是最後的深度圖。

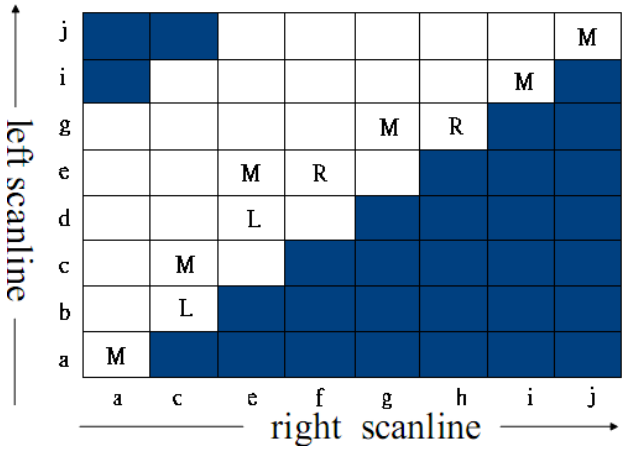


圖7 動態規劃的範例。在左右影像的每一對應列中，都會找到一條最適合的路徑。圖中 a~j 表示影像灰階；M 表示是對應到；L 表示只有左邊影像看得到（右邊影像被遮蔽）；R 表示只有右邊影像看得到（左邊影像被遮蔽）。

原始的動態規劃[11]主要分成兩個步驟，第一步驟計算動態規劃矩陣，將影像中每一列計算出一個動態規劃矩陣，計算的方法有很多種，其中最基本的就是以[9]公式(1)計算，d_{sad} 為絕對差值的總和來當比對的成本，P 為左影像的座標點，Q 為右影像中的座標點，d_R 和 d_G 和 d_B 為 RGB 色彩空間的元素(p_R 為 p 點紅色元素，q_R 為 q 點的紅色元素)，公式(2)中的 x 和 y 為影像中的座標位置，d 為視差的值。將左影像中每個像素點 p(x,y)去計算差值當成比對成本值，將可得到動態矩陣 C，C 為一個 H*X*N 的 3 維矩陣，H 維影像的高度，X 為影像的寬度，N 為允許的視差長度。

$$d_{sad}(P,Q)=d_R(P,Q)+d_G(P,Q)+d_B(P,Q) \quad (1)$$

$$\begin{cases} d_R(P,Q)=\left|p_R(x,y)-q_R(x-d,y)\right| \\ d_G(P,Q)=\left|p_G(x,y)-q_G(x-d,y)\right| \\ d_B(P,Q)=\left|p_B(x,y)-q_B(x-d,y)\right| \end{cases} \quad (2)$$

第二步驟是在上述找到的每一列動態矩陣中去找出最佳路徑，利用全域性最佳化(Global optimization)公式(3)，E_{data} 項為第一步驟計算出來的動態矩陣 C(左影像與右影像中像素點的顏色差值)。而 E_{smooth} 項主要目的是為了視差值能夠平滑，E_{smooth} 項限制只有參考鄰近點視差的差值，可以使得同一個物件的視差平滑減少雜訊，而 λ 是一個常數，d 為視差值，用於懲罰深度不連續，dis(x,y)為在影像中座標(x,y)的視差值。

$$E(d) = E_{data}(d) + \lambda E_{sooth}(d) \quad (3)$$

$$E_{smooth}(d) = \sum_{x} |dis(x,y) - dis(x-1,y)| + |dis(x,y) - dis(x,y-1)| + |dis(x,y) - dis(x,y+1)|,.... \quad (4)$$

不過動態規劃會有一個缺點，會造成水平斑紋狀(streaking artifacts)的錯誤結果。如圖 8 所示。很多人就加入了一些方法把這個錯誤改善掉了，使得正確率可以提升許多。

圖8 動態規劃方法做出的深度圖，會有斑紋狀的錯誤效果。

3.3 其他比對方法

另外還有許多其他方法，像是比較簡單的 SAD(Sum of Absolute Differences)與 SSD(sum of square difference)[11]的比對方法，這是兩個方法主要都是比對一個區域內(window)的色彩值，找尋最相近的色彩值當作對應點，這個概念在其他論文中還是常常被使用到。不過這個方法比較簡單，但是效果並不會很好，速度也不快。

另外還有常常被使用到的方法“可信度傳遞”（belief propagation）[12-18]，利用區塊(sad or ssd)比對的方式求視差，只是將雙影像中所有對應點的局部最小值求出，因此求得的視差圖準確率不高。而利用全域性最佳化視差求法，像是可信度傳遞（belief propagation)，利用所有視差提供可供信賴的程度，將整個視差圖經過疊代更新，直到視差圖收斂為止。利用可信度傳遞於深度不連續處（depth discontinuities)，由於信息會在此處停止傳遞，因此彼此不同深度的信息將不會互相影響。可信度傳遞的這個方法效果不錯，但是處理的時間會很長，比較難以達到即時的效果。

還有針對不同的光源的條件下的比對方法，如圖 9，由於左右眼視差的關係，不同視角所以可能導致某些區域會有光源不同的情形，而[19]這個演算法可以針對不同光源的情況做出處理，他主要是將 RGB 色彩空間轉換成 HSL 的色彩空間來比對，使得光源的問題得以改善。

(a) Left image (b) Right image

圖9 這兩張分別為左右影像，由於雙眼不同視角，導致有視差效果，因此使得兩張影像上方的光源不同。

另外就是還有許多方法有用到切割(segmentation)，事先將左眼影像做切割，把每個物件切割出來再做比對，加入切割後會使每個物件的深度值比較平滑，比較不會有雜訊，不過這就要看切割的方法好不好，如果能把每個物件精準的切割出來，那麼在比對上也會比較好比對。不過要切割的好，那可能也是要花比較多的時間。

4. 研究方法

於 2D 影像估測深度圖的實驗中，我們的方法也是將影像每一列分開來做比對，但主要是以線段比對為主，線段的意思代表者在影像水平線上，如果顏色相近的連續點，我們稱之為線段。因為我們假設一個物件每個像素點與鄰近像素點的顏色都很相近，所以這些視差變化是會很平滑的(smooth)，所以我們認定每一個線段的視差是很平滑的，所以我們將影像每一列分開來處理，每一列中又會有許多線段，將左影像的每一線段分別到右影像中找哪個線段的顏色最為相近，便可找到對應的線段，這樣就可以找到視差值。

圖 10 為流程圖，實驗一開始必須先輸入一對平行的左右影像，接下來分成四個主要部分，第一個部分是左影像的邊緣偵測，利用 RGB 色彩來判斷，此目的是為了求得線段出來。第二部分是將計算成本矩陣，計算左影像中每個像素點與右影像中每個像素點的 RGB 色彩差異值存入矩陣中，第三個部分是找出對應線段，在成本矩陣中找出像素點色彩差異值最小的，便認為它們是對應線段，便可找到視差。第四個部分是提出了三個方法來偵測遮蔽區域，並改善遮蔽區域的錯誤。

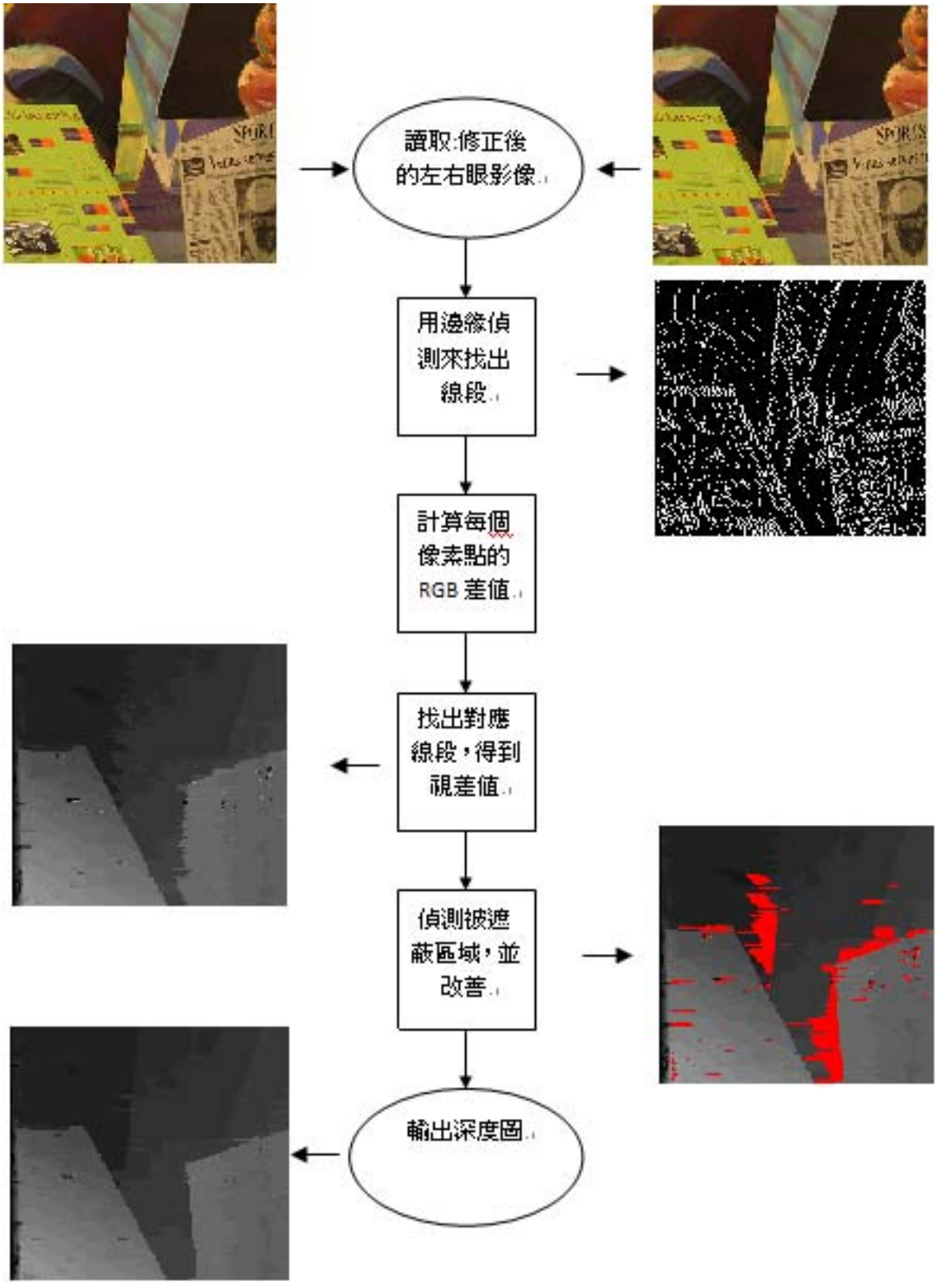

讀取:修正後的左右眼影像
用邊緣偵測來找出線段
計算每個像素點的RGB 差值
找出對應線段，得到視差值
偵測被遮蔽區域，並改善
輸出深度圖

圖10 流程圖

4.1 利用邊緣偵測找出線段

輸入左右影像後，我們開始針對左影像進行彩色影像邊緣偵測，彩色影像邊緣偵測是利用 RGB 色彩空間來找出邊緣，此偵測的目的找出線段在後面線段比對時需要用到。彩色影像邊緣偵測是從影像左上方開始，由左向右，由上至下的方向。一開始以第一個像素點為基準，和第二個像素點比較 RGB 顏色差異值是否相近，如果相近的話，認為此兩點為同一個線段，則將此點設為黑點，並繼續判斷下一個像素點的 RGB 顏色差異是否與此線段的所有像素點 RBG 顏色平均值是否相近，相近的話設為黑點，如果不相近，則認為此像素點是不同線段的，則將此點設為白點，以公式(5)做計算，這公式判斷線段中所有像素點的 R 與 G 與 B 元素平均與被判斷的像素點的 R 與 G 與 B 元素的個別差值是否小於門檻，小於的話則判斷成同一個線段，l_R 為線段中所有像素點的 R 元素值的平均，th 為臨界值，q_R 為判斷點的 R 元素值。

$$\begin{cases} |(l_R - q_R)| < th \\ |(l_G - q_G)| < th \\ |(l_B - q_B)| < th \end{cases} \qquad (5)$$

並且我們必須限制每個線段最大的長度為 30，限制長度的目的是為了避免每個線段太過平滑，因為有些場景的物件是有傾斜的現象，像是圖 11 中房屋的咖啡色屋簷。這是有傾斜的情況，假如這個屋簷都給予同一深度值的話，那就會有錯誤了。這屋簷的深度值應該是由上到下遞減；由右到左慢慢的變小，而不是同一個深度值。所以我們必須限定一個線段的長度，以免線段太長，如果線段太長而此物件又有傾斜的情況，那很有可能會有錯誤。以此方法一直做下去直到整張圖做完為止，結果如圖 12 所示。由此圖中可以看出，白色的像素點為邊緣的像素點；水平連續的黑色像素點為一個線段。如圖 13 說明如何判斷成一個線段，統計黑色像素點，如果遇到白色像素點則停止，統計此白色像素點到前一個線段內有幾個像素點，當成線段長度。然後再由此白色像素點當起始點開始計算下一個線段的長度。

圖11 圖中咖啡色的屋簷有傾斜的情況。

圖12 左圖為原始影像，右圖為偵測後的結果圖，白色像素點為邊緣。

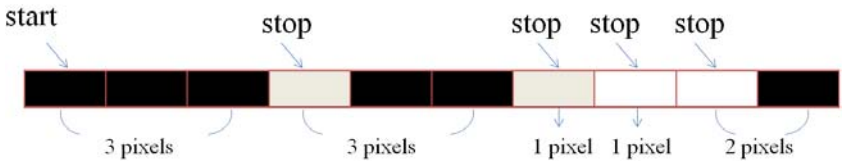


圖13 有十個像素點，白色像素為邊緣像素點。這十個像素點分成五個線段。

4.2 計算成本矩陣

接下來就是計算成本矩陣，將影像每一列分開計算，每一列就會有一個成本矩陣，我的方法是用 RGB 色彩差值當作成本利用公式(1)和公式(2)來計算，以最基本的 SAD 的方式來計算。如圖 14 說明，將每一列分開計算，將左影像的列與右影像的列的每一個像素點計算 RGB 色彩差值，然後存入矩陣中，接下來向左位移一個像素點後再將每個像素點取差值存入矩陣，依序作下去直到位移 N-1 像素為止。此目的是將所有像素點的顏色差值全部計算出來並存到矩陣中，避免重複計算。

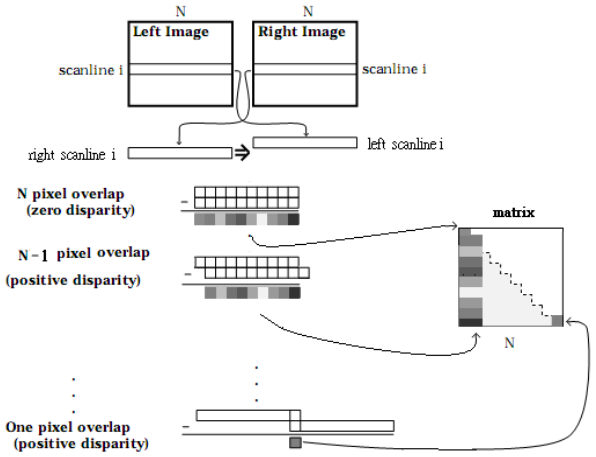


圖14 描述成本矩陣的如何產生。在左影像與右影像中取得第i列的像素點，將左影像第k個像素點的RGB值減去右影像第k個像素點的RGB值(k=0, 1, 2…N)並將其值存入矩陣中，然後位移一個像素點，再做相減並存入矩陣，直到位移N-1個像素點為止。

4.3 找出對應點

接下來的步驟就是找出每個線段的視差值，利用邊緣偵測圖來找出左影像中每一列的線段，每一列會有很多線段，將這些線段 l 依序做比對，我們比對的順序是從最左上開始，由影像上到下，由左到右順序比對，但在處理比對前必須先判斷此線段是哪種情況，第一種情況就是此線段中的像素點超過一個點的情形，另一種情況就是線段只有一個點的情形，我們將這兩種情形分開處理。一開始我們的想法是利用線段的顏色資訊來比對，找出每個線段的視差值。用公式(6)來計算， n 為目前比對的線段 l 中所有的像素點的個數。E_{data} 項是將線段 l 中所有像素點與右影像中同樣座標位置的像素點的 RGB 色彩差值，如公式(7)， p_R 為 p 點紅色元素，q_R 為 q 點的紅色元素，$p(x_i, y)$ 為左影像中比對線段的座標位置，q 點為右影像中 (x_i, y) 座標的像素點，d 為視差值(位移量)，$d=0, 1, 2\cdots d_{max}$。當 d=0 時是將左影像線段 l 中的像素點與右影像中同樣座標位置的像素點作 RGB 色彩差值的計算，當 d=1 時是將左影像線段中的像素點與右影像中同樣座標位置向左位移一個像素後作色彩 RGB 差值的計算。我們找出最小的 E(d)值出來($d=0, 1, 2\cdots d_{max}$)，把此最小的 E(d)的 d 值當成線段的視差值。結果如圖 16(c)，從圖中可知做出來的效果並不好，雜訊很多。我們只是用很簡單的彩色影像邊緣偵測來找線段，所以找出來的線段並不是很好，有些線段太短，甚至一個線段只有一個像素點，導致比對上容易出錯。

$$\begin{cases} E(d)=\sum_{i=1}^{n} E_{data}(d) & \text{if } n>1 \\ E(d)=E_{data}(d) & \text{if } n=1 \end{cases} \qquad (6)$$

$$E_{data}(d)=\left|p_R(x_i,y)-q_R(x_i-d,y)\right|+\left|p_G(x_i,y)-q_G(x_i-d,y)\right| \\ +\left|p_B(x_i,y)-q_B(x_i-d,y)\right| \quad (7)$$

所以我們另外加入了另一個方法，利用全域性最佳化(Global optimization)公式(3)來計算。我們將公式(3)改成公式(8)，E_{smooth} 項主要是考慮鄰近像素點的視差值，如果與鄰近視差值相距越大，E_{smooth} 項的值會越大，此目的是為了讓深度變化平滑。E_{smooth} 項我們將公式(4)裡的 E_{smooth} 改成公式(9)，假如此線段的像素點超過一點的情況，我們參考的鄰近點是上方及左方像素點的視差值；假如此線段只有一個點的情況，我們只參考上方像素點的視差值。x 為影像中像素點的水平座標位置，y 為影像中像垂直座標位置。dis(x,y)為此座標像素點的視差值。E_{smooth} 也是將線段中所有像素點去做計算，我們將找出此線段 E(d)值最小的當作對應位置，d 值就是位移量也就是視差值。結果圖如圖 16(d)，從結果圖看出效果有改善很多，結果比較平滑了，但是錯誤的地方還是很多，所以又另外想了一個方法來改善。

$$\begin{cases} E(d)=\sum_{i=1}^{n}E_{data}(d)+\sum_{i=1}^{n}\lambda E_{smooth}(d) & \text{if n}>1 \\ E(d)=E_{data}(d)+\lambda E_{smooth}(d) & \text{if n}=1 \end{cases} \quad (8)$$

$$\begin{cases} E_{smooth}(d)=\begin{bmatrix}\left|d-dis(x_i-1,y)\right| \\ +\left|d-dis(x_i,y-1)\right|\end{bmatrix} & \text{if n}>1 \\ E_{smooth}(d)=\left|d-dis(x_i,y-1)\right| & \text{if n}=1 \end{cases} \quad (9)$$

我們另再試了一個方法來改善，加入鄰近像素點的色彩資訊進去，把週遭的色彩資訊一起加入比對，我們使用了 11*1 的區塊(window)把色彩資訊累加起來，用公式(10)計算，此公式是累加線段中所有像素點的垂直色彩資訊，E_{data} 項由公式(7)改成公式(11)，p_R 為左影像比對線段中的像素點的紅色元素，q_R 為右影像中像素點的紅色元素。結果圖如圖 16(e)，從結果圖發現與 16(c)有了明顯的改善，但效果也還不是很好。此時我們發現分別加入了這兩個方法做出來的結果都有很大的改善，而且這兩個方法錯誤的地方都不同。如圖 15，上方為原始只用線段色彩資訊來做比對的結果，左圖為加入 E_{smooth} 後的結果，右圖為加入 11*1 垂直色彩資訊的結果，下圖為合併 E_{smooth} 與 11*1 垂直色彩資訊一起使用。我們將這兩個方法一起使用，合併成公式(12)，d 為視差值(d=0, 1, 2…d_{max})，n 為目前比對的線段 l 中所有的像素點的個數，E_{data} 為公式(11)來做計算，E_{smooth} 為公式(9)計算。結果如圖 16(f)，由結果圖發現合併後的方法效果最好，效果很平滑，雜訊也變少很多了。

$$\begin{cases} E(d)=\sum_{i=1}^{n}\sum_{k=-5}^{5}E_{data}(d) & \text{if n}>1 \\ E(d)=\sum_{k=-5}^{5}E_{data}(d) & \text{if n}=1 \end{cases} \quad (10)$$

$$E_{data}(d)=\begin{pmatrix}\left|p_R(x_i,y+k)-q_R(x_i-d,y+k)\right|\\+\left|p_G(x_i,y+k)-q_G(x_i-d,y+k)\right|\\+\left|p_B(x_i,y+k)-q_B(x_i-d,y+k)\right|\end{pmatrix} \tag{11}$$

$$\begin{cases}E(d)=\sum_{i=1}^{n}\sum_{k=-5}^{5}E_{data}(d)+\sum_{i=1}^{n}\lambda E_{smooth}(d) & \text{if n}>1\\ E(d)=\sum_{k=1}^{5}E_{data}(d)+\lambda E_{smooth}(d) & \text{if n}=1\end{cases} \tag{12}$$

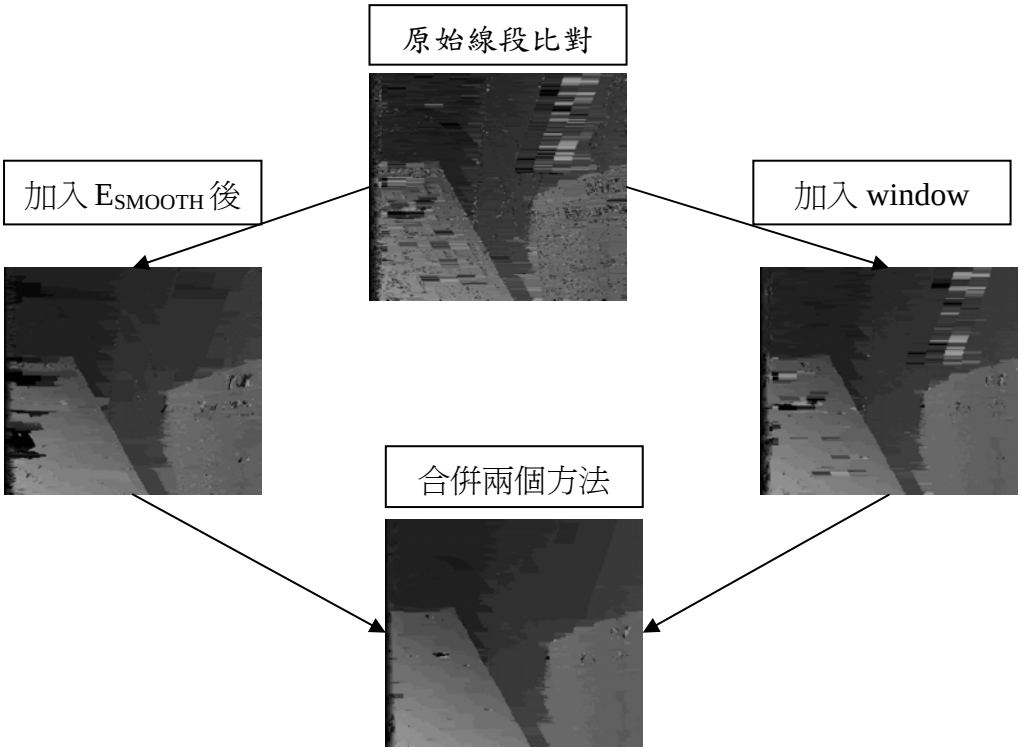
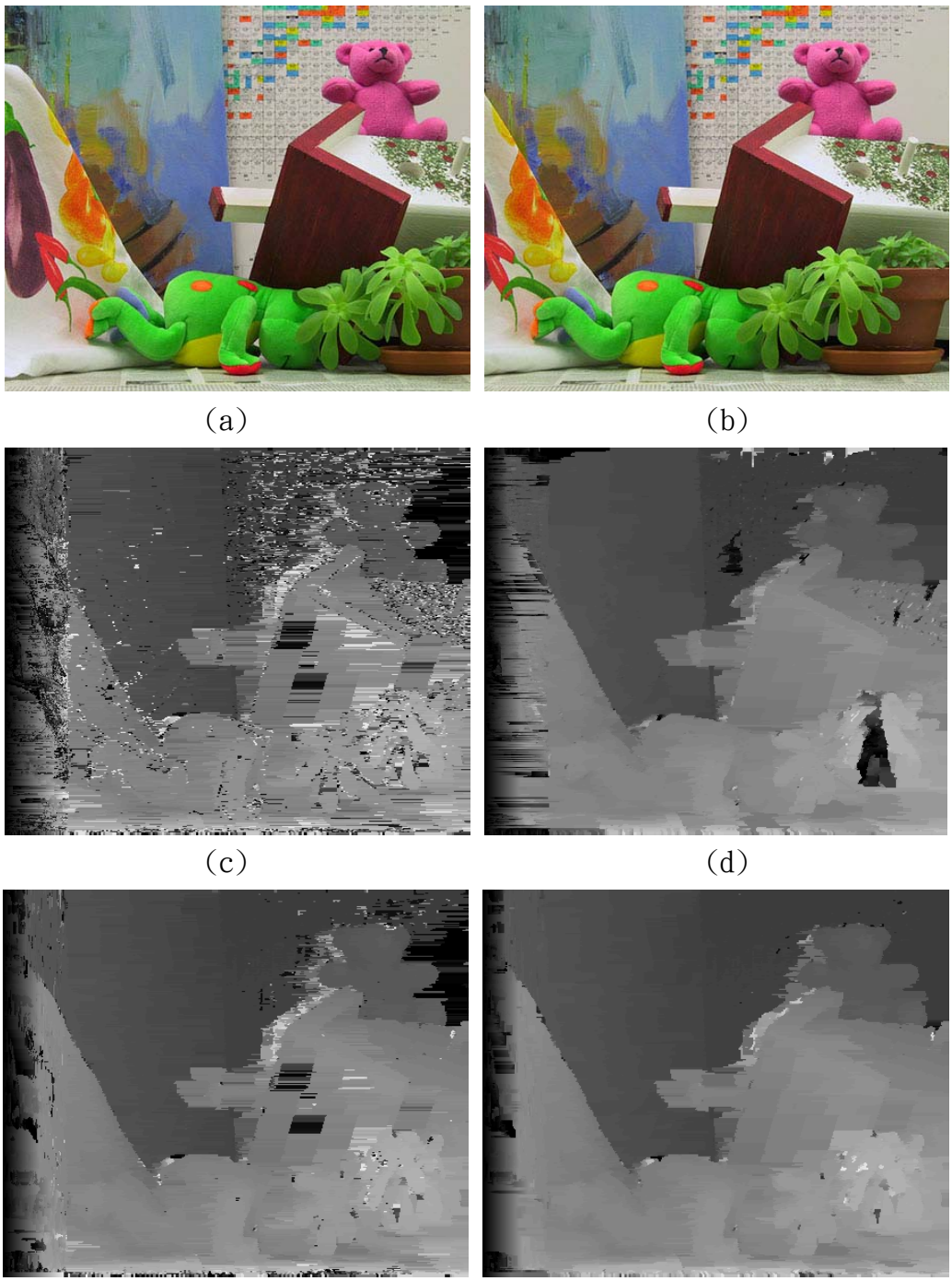


圖15 上方的影像為一開始沒加入任何改善前的結果，左邊的影像為加入E_{smooth}項後的結果，右邊的影像為加入鄰近11*1區塊顏色後的結果，下方的影像為加入E_{smooth}項跟鄰近11*1區塊顏色後的結果。

由此可見，如果只用線段的色彩資訊比對的話是不夠的，雜訊非常的多，因為線段太短的情況，而且周圍像素點的色彩又很相近時，非常容易比對錯誤。所以加入 E_{smooth} 項進去，它的目的是考慮周圍深度值，如果與周圍的深度值差距愈大，懲罰會愈大。以及加入鄰近像素點的色彩資訊進去，把周遭的色彩資訊一起加入比對，我們使用了 11*1 的區塊(window)把色彩資訊累加起來。可以看出雜訊都可以消除掉了，並且錯誤的區塊也大大改善了，不過被遮蔽的區域錯誤無法改善到，所以我們下一個章節提出了三個方法來改善被遮蔽區域的錯誤。



(a) (b)

(c) (d)

(e) (f)

圖16 Teddy(a)為左影像，(b)為右影像，(c)沒有用11*1的區塊累加以及沒使用E_{smooth}的結果，(d)沒有使用11*1的區塊累加的結果。(e)沒有使用E_{smooth}的結果，(f)使用11*1的區塊累加以及使用E_{smooth}的結果。

4.4 偵測遮蔽區域並且改善

由圖 16 可得知我們線段比對方法在被遮蔽區域的地方錯誤率非常高。被遮蔽區域會發生在前景和背景交錯的地方，由於前景的位移量(視差)大，所以前景會遮蔽到背景。我們無法得知被遮蔽區域的色彩資訊，所以無法準確的估測出此區域的視差值。因此我們提出了三個偵測遮蔽區域的方法，並且改善它。第一個方法是針對被遮蔽區域視差差異劇烈的地方，此區域為前景與背景視差值差異大。第二個方法是偵測一些少數的錯誤視差值，我們利用九個區塊來偵測，統計每個區塊的視差值，然後過濾出出現機率較少的視差值。第三個方法主要針對深度變化較平緩的區域，此區域為前景與背景視差差異不會太大的區域，不過依然還是有錯誤的可能。

4.4.1 偵測被遮蔽區域視差差異劇烈的地方

首先我們先去偵測被遮蔽區域並且視差變化大的地方，由於前景與背景視差變化大的話，那麼被遮蔽的區域就被比較大，由於無法得知被遮蔽區域的色彩資訊，所以此區域的錯誤率很高。所以我們先改善掉視差變化大的區域。

由於 Middlebury 網站的測試圖只有負視差與零視差的情況，負視差情況是左影像中的物件出現在右影像中會往左位移，所以比對上左影像中的像素點去右影像找對應點時，只須往左邊的方向找尋。因此只有一種情況會被遮蔽到，如圖 17 說明圖，左圖為左影像，右圖為右影像，縱軸為深度變化，橫軸為影像水平座標位置，A 和 B 分別表示不同的物件，紅色區域為被遮蔽的區域。左圖中 A 物件並沒有遮蔽到 B 物件，但在右圖中 A 物件遮蔽到 B 物件，因為 A 物件的深度值比 B 物件的深度值還要大(深度值越大代表距離攝影機距離較近，所以位移量大)。所以被遮蔽的情況只有一種，當右邊物件的深度值大於左邊物件的深度值時，這種情況才會有遮蔽的可能。

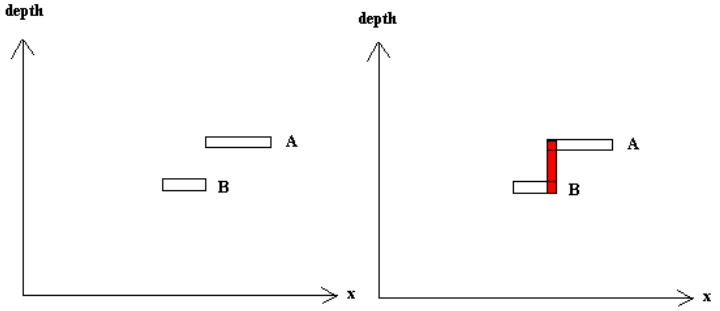


圖17 說明被遮蔽的情況:左圖為左影像，右圖為右影像，縱軸為深度變化，橫軸為影像水平座標位置，A 和 B 分別表示不同的物件，紅色區域為被遮蔽的區域。

我們偵測的方法是去判斷每一個線段的左右兩個線段，如果右邊的線段的視差值大於左邊線段的視差值時，可能就有遮蔽的情況。因此由公式(13)來表示，假如右邊線段的視差值比左邊線段的視差值大超過 1 以上的話，那麼此線段被遮蔽的可能性非常大，我們將此被偵測的線段標記起來，Q 表示右邊線段的座標點，S 表示左邊線段的座標點， $D_L(Q)$為左線段的視差值，$D_L(S)$為右線段的視差值。假如右線段的深度值比左線段的深度值大超過 1 的話，那

麼 Line(p)的點將被標記起來。由圖 18 說明，圖中有綠色跟紅色跟藍色連續三個線段，綠色線段表示被偵測線段的左邊線段，紅色表示被偵測是否被遮蔽的線段，藍色表示被偵測線段的右邊線段，S 表示綠色線段的座標點，P 表示紅色線段的座標點，Q 表示藍色線段的座標點。假設右邊線段(藍色)會比左邊線段(綠色)的視差值大超過 1 以上的話，那麼被偵測線段(紅色)就有可能被遮蔽到，因此我們將被偵測線段標記起來。結果如圖 19 所示，紅色區域為我們標記成被遮蔽的區域。找出錯誤點後，我們接下來利用 4.2 章節改善的方法來改善，改善後的結果如圖 19(d)所示，我們將紅色區域(被遮蔽的區域)改善後的結果，可從圖 19(b)與 19(d)作比較我們的方法將被遮蔽區域做了不錯的改善。

$$D_L(Q) - D_L(S) \geq 2 \qquad (13)$$

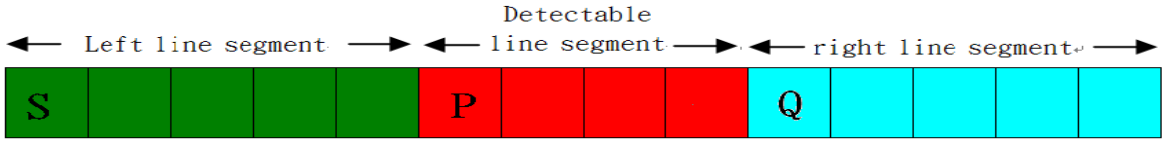
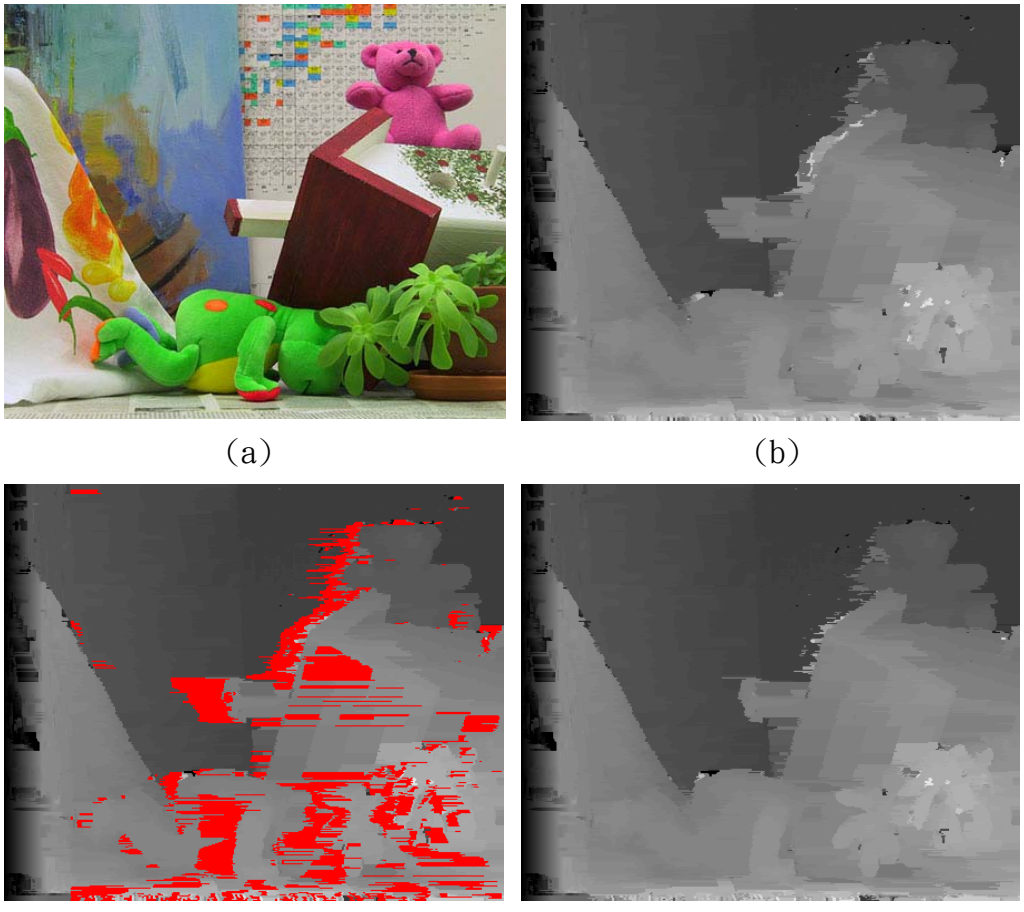


圖18 圖中有綠色跟紅色跟藍色三個線段，假如右線段(藍色)的視差值比左線段(綠色)的深度值大超過1的話，那麼被偵測線段(紅色)就會被標記。



(a) (b)

(c) (d)

圖19 Teddy(a)為左影像(b)為改善前的影像(c)為我們將被遮蔽的區域標記成紅色，(d)改善後的結果。

4.4.2 改善的方法

接下來針對被標記的線段做改善，被標記的線段都是被偵測為此線段有被遮蔽到，所以此線段的視差值可能都有錯。由於被遮蔽的區域的色彩資訊無法得知，所以不管如何比對都很難比對出準確的視差值。所以我們想了一個方法，在被改善的像素點設定一個搜尋範圍，這個範圍內的每像素點與被改善的像素點色彩差異小於門檻時，我們將此點的視差值加入統計。統計出這個範圍內哪個視差值出現機率最高，那麼就將出現出率最高的視差值取代被改善的像素點的視差值。不過範圍內有被標記的像素點，此像素點我們不作統計，像素點被改善後，我們會將此點的標記取消。

利用統計表 H 統計這個範圍內所有像素點的視差值，針對被標記線段 Line(p)的每一個 p 像素點的視差值作 p±(0,N)和 p±(N,0)範圍統計。用公式(14)表示，假設這範圍內的像素點為 q 點，假如範圍內的 q 點與 p 點的色彩差值小於門檻 th 的話，那麼 $H(D_L(q))$將加 1，$D_L(q)$表示左影像中 q 點像素點的視差值。

$$D_L(P) = \arg\max_d H(D_L(q)) \qquad (14)$$

將統計完成的Ｈ統計表後，在統計表中找出 d(視差值)=0, 1, 2…max 中出現機率最大的視差值，此視差值用來取代 P 點像素點的視差值。

4.4.3 偵測一些少數的錯誤視差值

接下來我們要過濾掉一些少數的錯誤視差值，此目的是為了消除雜訊點。我們的方法是統計一個大範圍全部像素點的視差值，統計完後我每會去判斷個視差值，如果某一個視差值小於這個大範圍的 2%的話，那將被我們認為是雜訊，把這個視差值的位置找出來並且標記起來。利用統計表 H 統計一個 h/2*w/2 範圍的每個像素點視差值(h=影像的高度，w=影像的寬度)，利用這個統計的方法統計左影像中的九個範圍，這九個範圍依序是圖 20 中的[1,2,3,4]，[5,6,7,8]，[9,10,11,12],[13,14,15,16]，[3,4,9,10]，[2,5,4,7]，[7,8,13,14]，[10,13,12,15]，[4,7,10,13]。這九個範圍分別統計，並判斷每個視差值個數小於 h/2*w/2*0.02 的話(h:影像高度，w:影像寬度)，便把此視差值的像素點標記起來成 1，其他沒有錯誤的像素點標記成 0。每個範圍統計完成後，便可找出錯誤的像素點。如圖 4.9(c)所示，紅色區域為被標記成錯誤的像素點。我們接下來也是利用 4.2 章節改善的方法來改善，改善後的結果如圖 4.9(d)。由圖 4.9(b)改善前的影像與圖 4.9(d)改善後的影像比較。可以看得出來被標記的區域有了改善。

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

圖20 上圖表示將左影像分成16個等分。

至於我們為什麼要用這九個區塊來找呢?那是因為假如一個物件剛好在中心的位置如下圖 21，而我們只用四個區塊來做統計的話，那麼那中心的那個物件將會被標記成錯誤的像素點。因為我們必須避免一個物件分屬在不同範圍的情況，所以用了這九個區塊來做統計。

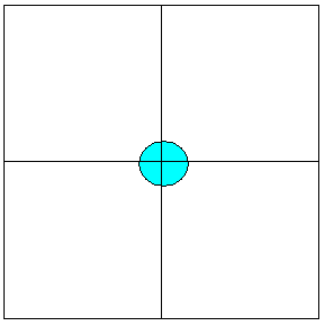
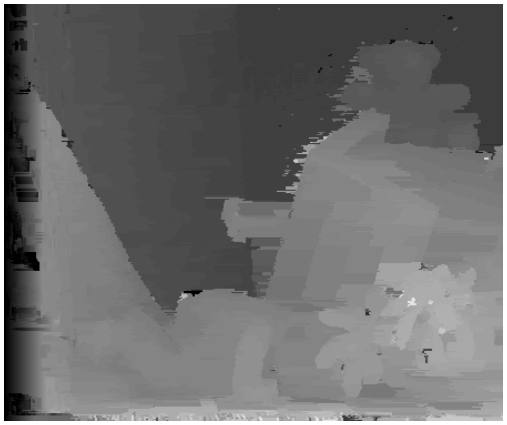
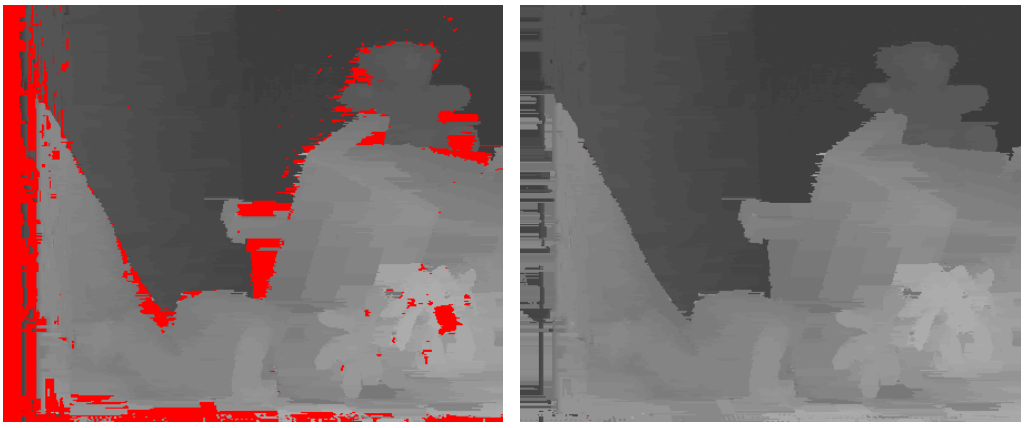


圖21 中心圓圈的物件剛好分屬在這四個範圍，假如我們只用這四個區塊來統計的話，那麼這個圓圈的物件會被標記成錯誤的情況。為了避免這種情況，所以用了九個區塊來統計。

(a)



(b)



(c) (d)

圖22 Teddy(a)為左影像(b)為改善前的影像(c)為我們將被遮蔽的區域標記成紅色，(d)改善後的結果。

4.4.4 偵測被遮蔽區域視差差異較平緩的區域

第三個偵測的方法與 4.4.1 的方法一樣，由於已經把視差變化劇烈的地方做改善了，不過還有視差變化平緩的區域沒有做改善。視差平緩的區域就是前景視差值比背景視差值大於等於 1 的情況，雖然只有視差值的差異只有 1，但是這個情況還是有被遮蔽到，依然會影響到比對上的準確性，所以這個地方我們還是要去做偵測及改善。我們去判斷每一個線段的左右兩個線段，假如右邊的線段是視差值比左邊線段的視差值大於等於 1 的話，並且被偵測的線段與右邊的線段視差值相等的話，我們便把此線段標記起來。利用公式(15)來做判斷，$D_L(Q)$為左線段的視差值，$D_L(S)$為右線段的視差值，$D_L(P)$為被偵測線段的視差值，Q 為右線段的像素點，S 為左線段的像素點，P 為被偵測線段的像素點。假如右邊的線段是視差值比左邊線段的視差值大於等於 1 的話，並且被偵測的線段與右邊的線段視差值相等的話，那麼被偵測線段將被標記起來。由圖 23 說明，圖中有綠色跟紅色跟藍色連續三個線段，綠色線段表示被偵測線段的左邊線段，紅色線段表示被偵測是否被遮蔽的線段，藍色線段表示被偵測線段的右邊線段，S 表示綠色線段的座標點，P 表示紅色線段的座標點，Q 表示藍色線段的座標點。假設右邊線段(藍色)會比左邊線段(綠色)的視差值大於等於 1，並且被偵測線段(紅色)與右邊線段(藍色)的視差值相等的話，那麼被偵測線段(紅色)就有可能被遮蔽到，因此我們將被偵測線段標記起來。結果如圖 24(c)所示，將被遮蔽的線段標記起來，接下來也是利用 4.2 章節改善的方法來改善，改善後的結果如圖 24(d)。由圖 24(b)改善前的影像與圖 4.24(d)改善後的影像作比較，由此可以看出我們可以把被標記的區域做了明顯改善。

$$\begin{cases} D_L(Q) - D_L(S) \geq 1 \\ D_L(P) = D_L(Q) \end{cases} \tag{15}$$

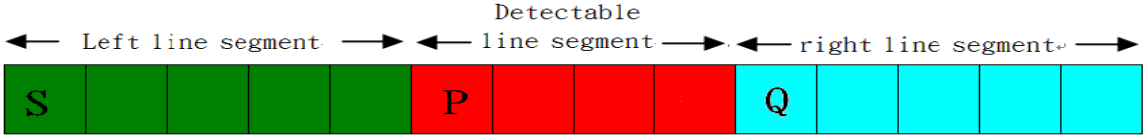
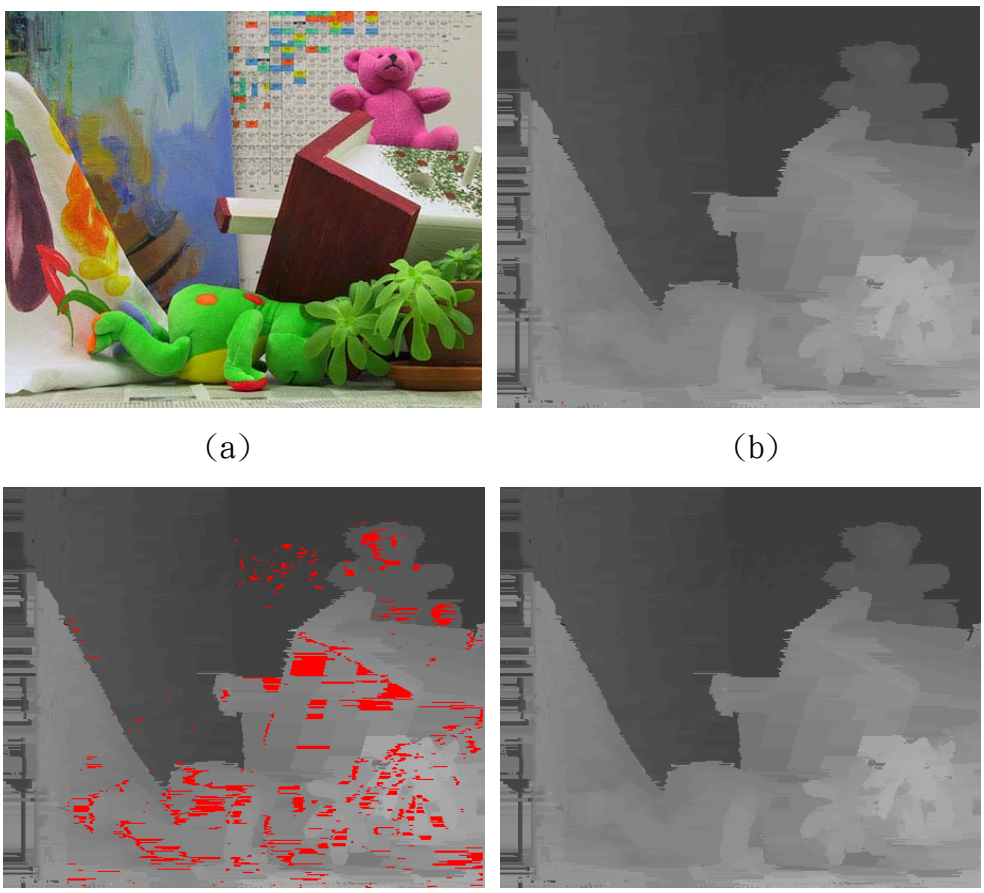


圖23 圖中有綠色跟紅色跟藍色三個線段，假如右線段(藍色)的深度值比左線段(綠色)的視差值大於等於1，並且被偵測線段(紅色)與右線段(藍色)的視差值相等的話，那麼被偵測線段(紅

色)就會被標記。



(a) (b)

(c) (d)

圖24 Teddy(a)為左影像(b)為改善前的影像(c)為我們將被遮蔽的區域標記成紅色，(d)改善後的結果。

5. 結果與討論

我們用我們提出的方法來評估 Middlebury 網站的資料 [1]。測試的電腦平台是 CPU Core i5 2.67GHz 和 2GB 記憶體。測試四組影像結果顯示在圖 25，26，27，28，這四組影像都是使用 RGB 色彩空間來做估測。圖 25(a)，26(a)，27(a)，28(a)為左影像。圖 25(b)，26(b)，27(b)，28(b)為左影像的真實的深度影像(ground truth)。圖 25(c)，26(c)，27(c)，28(c)為改善遮蔽區域錯誤前的影像。圖 25(d)，26(d)，27(d)，28(d)為改善遮蔽區域錯誤後的影像。圖 25(e)，26(e)，27(e)，28(e)為改善遮蔽區域錯誤前錯誤的像素點，黑色的像素點為非遮蔽區域的錯誤點，灰色的像素點為遮蔽區域的錯誤點。圖 25(f)，26(f)，27(f)，28(f)為改善遮蔽區域錯誤後錯誤的像素點，黑色的像素點為非遮蔽區域的錯誤點，灰色的像素點為遮蔽區域的錯誤點。由圖 25(c)，.26(c)，27(c)，28(c)與圖 25(d)，26(d)，27(d)，28(d)比較可看出我們改善的方法大大的減少被遮蔽區域的錯誤。第一個改善的區域是偵測視差值差距大的區域，由於右線段視差值比左線段視差值大到超過 1 以上的區域可能造成遮蔽區域大，這個地方的錯誤率偏高。第二個改善的區域主要針對一個大區域內錯誤的視差值。第三個改善的區域主要針對視差值較平緩的區域，由於右線段視差值比左線段視差值大於等於 1 的區

域，這個區域的被遮蔽的範圍較小，不過還是可能導致計算出錯誤的視差值。而我們三個改善的方法都是一樣，都是由錯誤像素點附近範圍找尋顏色相近像素點的視差值作統計，統計出哪一個視差值出現機率最大，將取代錯誤像素點的視差值。

表一由 Middlebury 網站評估出來改善遮蔽區域錯誤前的錯誤率，每組影像會評估出三項分數，第一項 nonocc 表示非遮蔽區域的錯誤率，第二項 all 表示整張影像的錯誤率，第三項 discontinue 表示非連續區域的錯誤率，最後還有一項分數為四組測試圖的平均錯誤率。改善遮蔽區域錯誤前的所估測出來的這四組影像深度資訊的平均準確率已經達到 83%，但是從結果圖 25(e)，26(e)，27(e)，28(e)看出被遮蔽的區域很容易估測錯誤(灰色的像素點很多)。表二為改善遮蔽區域錯誤後的錯誤率，平均準確率已達到 87.9%，從結果圖 25(e)，26(e)，27(e)，28(e)與圖 25(f)，26(f)，27(f)，28(f)做比較，遮蔽區域改善了很多(灰色的像素點少了很多)。

另外我們將做另一個實驗，將 RGB 色彩空間轉換到 YCrCb 色彩空間來估測深度圖。YCrCb 色彩空間中的 Y 為顏色的亮度，CrCb 都代表色彩，Cb 代表藍色資訊，Cr 代表紅色資訊。由於 YCrCb 色彩空間通常運用於影片或是數位攝影系統中，而一對左右眼影像估測深度資訊的研究中，輸入的左右影像也是由影片或是數位攝影系統中取得。因此我們實驗將 RGB 色彩空間轉換到 YCrCb 色彩空間來估測深度圖，看結果是否會有改善。公式(16)來做轉換，R 代表紅色元素，B 代表藍色元素，G 代表綠色元素。YCrCb 色彩空間估測出來的深度圖在圖 29(c)。圖 29(d)為 Middlebury 網站上估測出來錯誤的像素點(YCrCb 色彩空間估測出來的深度圖)，黑色的像素點代表非遮蔽區域的錯誤，灰色的像素點代表遮蔽區域的錯誤。圖 29(a)為使用 RGB 色彩空間估測出來的深度圖，圖 29(b)為 Middlebury 網站上估測出來錯誤的像素點(RGB 色彩空間估測出來的深度圖)。從圖 29(b)跟圖 29(d)做比較，可以看得出來前面三組 Tsukuba 跟 Venus 跟 Teddy 影像，使用 RGB 色彩空間估測出來的效果比較好一點，而最後面 Cones 影像，RGB 色彩空間跟 YCrCb 色彩空間估測出來的效果差不多。

$$\begin{cases} Y = 0.299*R + 0.587*G + 0.114*B \\ Cb = -0.1687*R - 0.3313*G + 0.5*B + 128 \\ Cr = 0.5*R - 0.4187*G - 0.0813*B + 128 \end{cases} \quad (16)$$

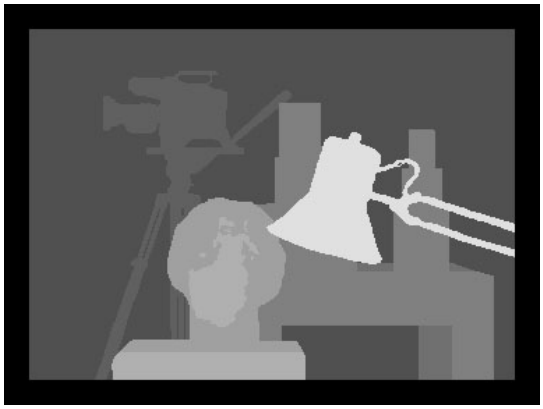
表三為 Middlebury 網站計算出四組影像的錯誤率(使用 YCrCb 色彩空間估測出來的深度圖)，第一項 nonocc 表示非遮蔽區域的錯誤率，第二項 all 表示整張影像的錯誤率，第三項 discontinue 表示非連續區域的錯誤率，最後還有一項分數為四組測試圖的平均錯誤率。可以從表二(使用 RGB 色彩空間估測出來的深度圖)跟表三(使用 YCrCb 色彩空間估測出來的深度圖)做比較，前三組影像(Tsukuba 跟 Venus 跟 Teddy)是 RGB 色彩空間的效果比較好，三項分數(非遮蔽區域，整張影像，非連續區域)用 RGB 色彩空間估測的錯誤率比較低，而第四組影像(Cones)用 YCrCb 色彩空間估測的錯誤率，只有非遮蔽區域跟非連續區域比較低一點，但是 YCrCb 色彩空間估測與 RGB 色彩空間估測的整張影像錯誤率一樣，都是 16.4%的錯誤率。所以以這四組測試影像可以得知使用 RGB 色彩空間即可。

表四為跟別人的方法做比較，測試影像跟我們實驗的影像一樣，評分的方式也是都由 Middlebury 網站做評分。第一個 RealTimeGPU[4] 的方法是用動態規劃為架構來估測的，不過他的方法加入很多計算做考量，所以他的方法估測的時間要比較久一點，如果估測 320*240

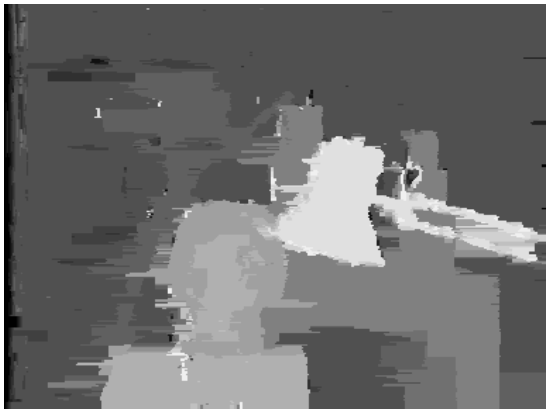
大小的影像大約需要 3.61~9.63 秒，不過他有使用 GPU(圖形處理器)來幫忙計算，使用 GPU 與 CPU 平行運算，減少估測時間。而我們的方法評估出來的前兩組影像(Tsukuba 跟 Venus)的整張錯誤率比 RealTimeGPU 方法還低，不過另外兩組影像的錯誤率就比 RealTimeGPU 方法的高了。而第二個 TreeDP[5]的方法是以原始動態規劃做改良的，使準確度更高，不過這的方法就沒有交代他的估測速度有多快了。而我們的方法評估出來的後兩組影像(Teddy 跟 Cones)的整張錯誤率比 RealTimeGPU 方法還低，但是前兩組影像錯誤率就比他的高了。第四個 DP 的方法就是原始的動態規劃的方法，我們的方法評估出來的四組影像的整張錯誤率都比 DP 的還要低，而且四組平均錯誤率比 DP 的方法將近要低了 2%。而最後兩個方法一個是使用 SSD 的方法架構來估測，另一個是以 SAD 的方法架構來估測，這兩組的估測的結果並不好，正確率不高。

我們演算法的主要優點是不需要執行任何的切割運算，只需要簡單的邊緣偵測，可以大量的減少運算時間。而我們直行時間大部分都花在線段的比對上面，我們執行四組影像所花的時間分別是 0.359 秒，0.625 秒，1.281 秒，1.297 秒(Tsukuba，Venus，Teddy，Cones)，改善的部分花不到 0.1 秒。而我們另一個特點，雖然我們的方法與 DP 相似也是以每一列分開來比對，但是，我們是以線段比對方式做比對，線段比對的方式可以抑止條紋狀(streaking)效果，但是以線段比對的方式會造成遮蔽區域的錯誤，所以我們用了三個方法將遮蔽的區域大幅做了改善，所以將遮蔽區域的錯誤大量的減少。

(a)



(b)



(c)

(d)

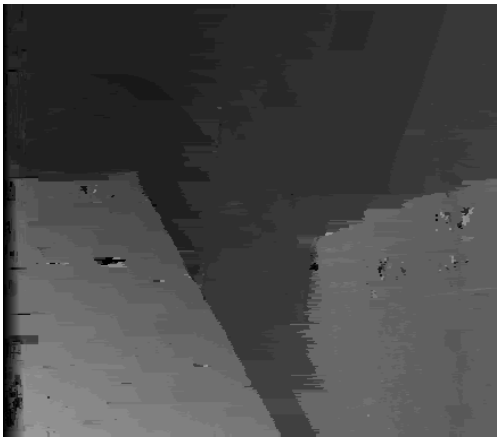
(e)

(f)

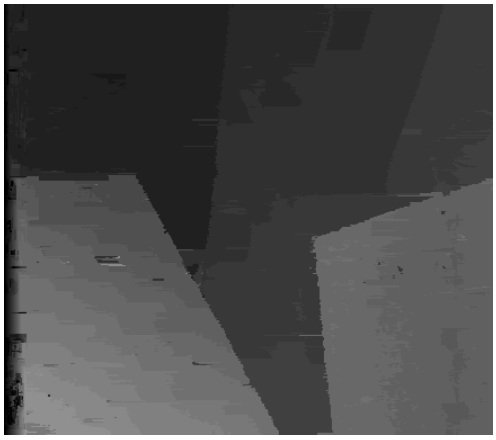
圖25 Tsukuba (a)左影像，(b)影像為真實深度資訊影像，(c)為改善前估測出來的深度影像，(d)改善後估測出來的深度影像，(e)為改善前估測錯誤的像素點(錯誤視差值大於1)，(f) 改善後估測錯誤的像素點(錯誤視差值大於1)。

(a)

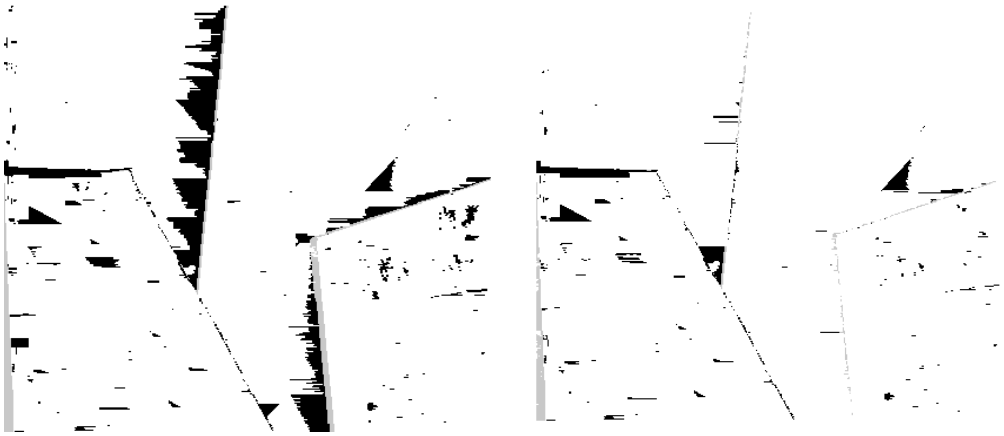
(b)



(c)

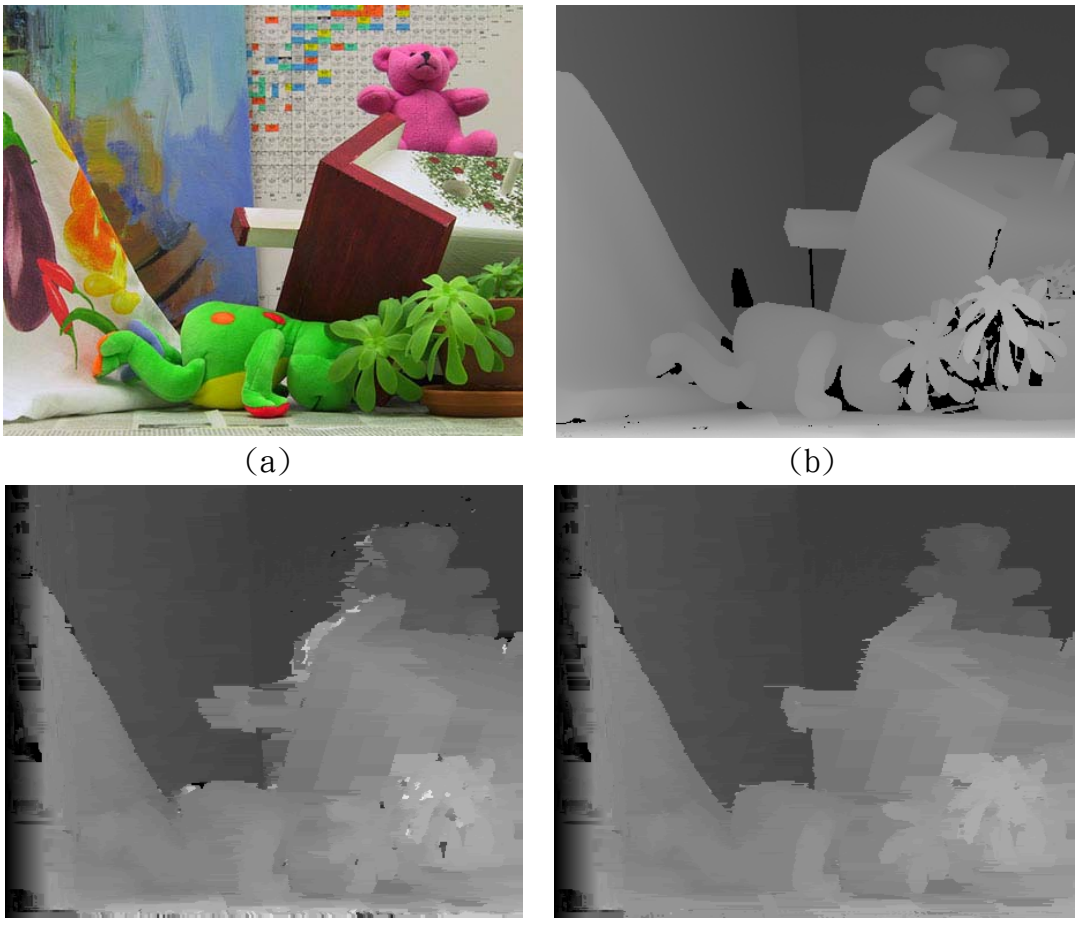


(d)



(e) (f)

圖26 Venus (a)為左影像，(b)影像為真實深度資訊影像，(c)為改善前估測出來的深度影像，(d)為改善後估測出來的深度影像，(e)為改善前估測錯誤的像素點(錯誤的視差值大於1以上)，(f) 為改善後估測錯誤的像素點(錯誤的視差值大於1以上)。



(a) (b)

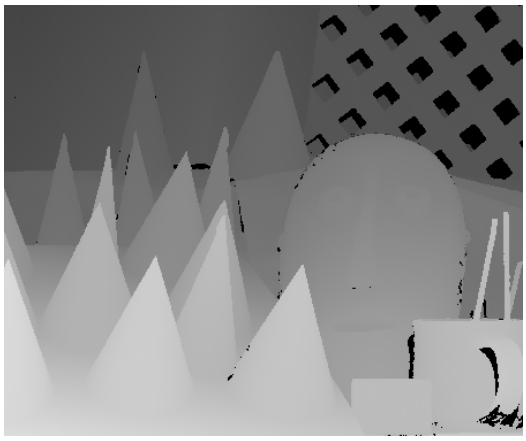
(c) (d)

(e)

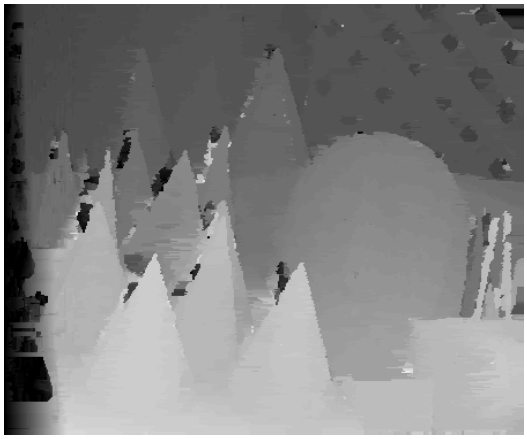
(f)

圖27 Teddy (a)為左影像，(b)影像為真實深度資訊影像，(c)為改善前估測出來的深度影像，(d)為改善後估測出來的深度影像，(e)為改善前估測錯誤的像素點(錯誤的視差值大於1以上)，(f) 為改善後估測錯誤的像素點(錯誤的視差值大於1以上)。

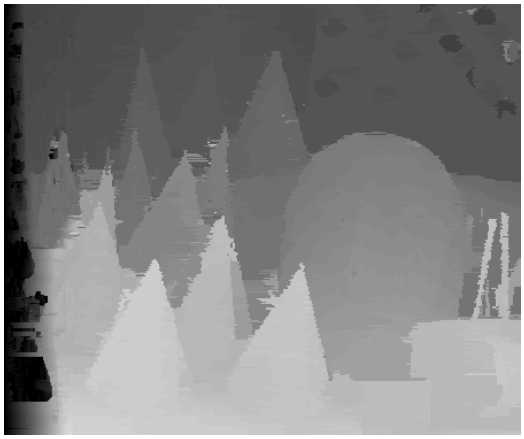
(a)



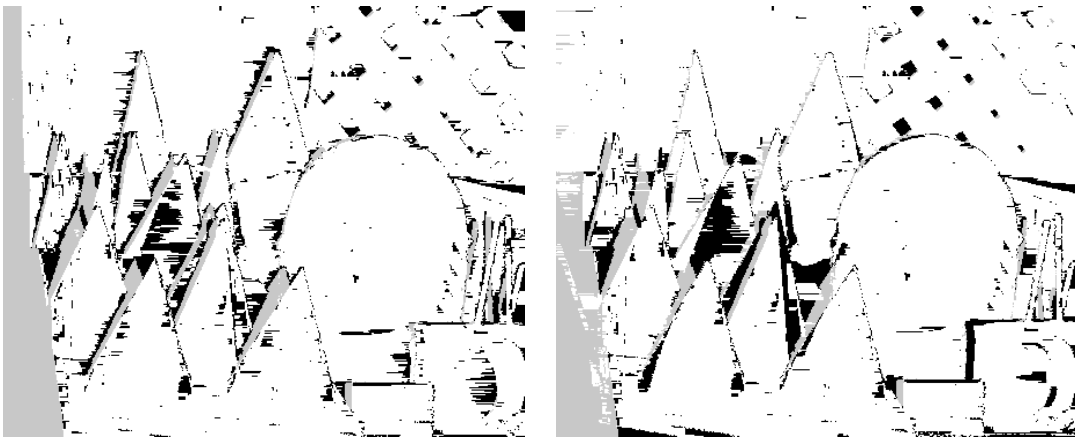
(b)



(c)

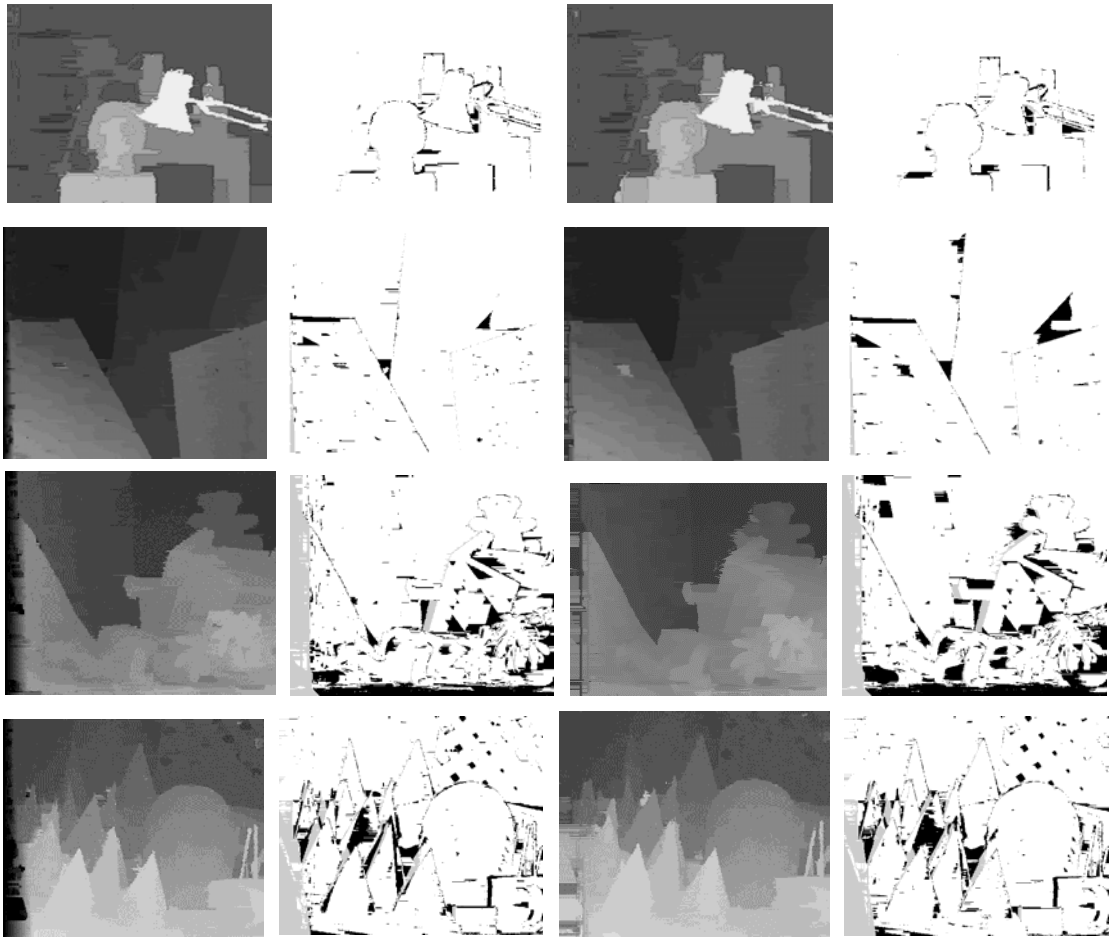


(d)



(e) (f)

圖28 Cones (a)為左影像，(b)影像為真實深度資訊影像，(c)為改善前估測出來的深度影像，(d)為改善後估測出來的深度影像，(e)為改善前估測錯誤的像素點(錯誤的視差值大於1以上)，(f) 為改善後估測錯誤的像素點(錯誤的視差值大於1以上)。



(a) (b) (c) (d)

圖 29 (a)為RGB色彩空間估測出的深度圖，(b)為RGB估測出來深度圖的錯誤像素點(錯誤的視差值大於1以上)，(c)為YCrCb色彩空間估測出的深度圖，(d)為YCrCb色彩空間估測出來深度圖的錯誤像素點(錯誤的視差值大於1以上)。

表一. 我們由 Middlebury 資料庫上評分了改善前的四組圖，第一項分數為非遮蔽區域的錯誤率，第二項分數為整張影像的錯誤率，第三項分數為非連續性的地方的錯誤率。

	nonocc	all	discontinue	Average percent of bad pixels
Tsukuba	5.75	7.88	22.2	17.0
Venus	6.45	7.98	29.6	
Teddy	15.3	24.0	32.5	
cones	9.88	19.9	22.3	

表二. 我們由 Middlebury 資料庫上評分了改善後的四組圖，第一項分數為非遮蔽區域的錯誤率，第二項分數為整張影像的錯誤率，第三項分數為非連續性的地方的錯誤率。

	nonocc	all	discontinue	Average percent of bad pixels
Tsukuba	2.54	3.22	12.7	12.1
Venus	1.98	2.83	11.1	
Teddy	13.8	20.6	29.3	
cones	9.61	16.4	21.2	

表三. 由 Middlebury 資料庫上評分將 RGB 色彩空間轉換到 YCrCb 色彩空間後估測的結果，第一項分數為非遮蔽區域的錯誤率，第二項分數為整張影像的錯誤率，第三項分數為非連續性的地方的錯誤率。

	nonocc	all	discontinue	Average percent of bad pixels
Tsukuba	3.03	3.89	14.5	13.9
Venus	3.67	4.62	17.1	
Teddy	17.3	24.6	32.8	
cones	9.30	16.4	19.2	

表四. 跟別人的方法做比較，第一項分數為非遮蔽區域的錯誤率，第二項分數為整張影像的錯誤率，第三項分數為非連續性的地方的錯誤率。

	Tsukuba			Venus			Teddy			Cones			Avg. percent bad pixels
	nocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
RealTimeGPU[4]	2.05	4.22	10.6	1.92	2.98	20.3	7.23	14.4	17.6	6.41	13.7	16.5	9.82
TreeDP[5]	1.99	2.84	9.96	1.41	2.10	7.74	15.9	23.9	27.1	10.0	18.3	18.9	11.7
Our method	**2.54**	**3.22**	**12.7**	**1.98**	**2.83**	**11.1**	**13.8**	**20.9**	**29.3**	**9.61**	**16.4**	**21.2**	**12.1**

DP[11]	4.12	5.04	12.0	10.1	11.0	21.0	14.0	21.6	20.6	10.5	19.1	21.1	14.2
SSD+MF[11]	5.23	7.07	24.1	3.74	5.16	11.9	16.5	24.8	32.9	10.6	19.8	26.3	15.7
LCDM+Adaptwat[19]	5.98	7.84	22.2	14.5	15.4	35.9	20.8	27.3	38.3	8.90	17.2	20.0	19.5

我們提出一個即時線段比對的方法從一對左右影像來評估深度資訊。在影像中每一列顏色相近的連續像素點，我們稱這些顏色相近的像素為線段，利用簡單的邊緣偵測方法來找出影像中所有的線段，再以線段比對的方法來計算出影像的視差資訊。不過線段比對的缺點是在被遮蔽區域的比對錯誤率很高。因此我們提出了三個改善的方法大幅的改善遮蔽區域的錯誤。由實驗結果中圖 25-28 顯示出我們提出的方法可以準確且即時的評估出深度資訊。

實驗結果證明了我們的方法是即時且準確的估出深度資訊。估測了 Middlebury 網站的四組圖，我們的方法準確率可高達 87.9%的正確性。而我們改善的方法可以大幅改善被遮蔽區域的錯誤，處理速度最快可到 0.359 秒。

未來的方向可以考慮犧牲一下速度，加入一些方法來提升準確，像是我們現在找線段的方法是用很粗略的邊緣偵測,找出來的線段並不好,可以加入切割方法來找出比較好的線段，使正確率能提高。以及加入一些其它的方法或特徵來增強比對上的準確性。以及可以考慮使用 GPU(圖形處理器)來分工幫忙計算，使我們的方法可以達到即時的效能。

參考文獻

[1]http://vision.middlebury.edu/stereo/eval/

[2] Ohta, Y., Kanade, T, “Stereo by intra- and inter-scanline search using dynamic programming.”IEEE Transactions on Pattern Analysis and Machine Intelligence ,7(2),pp.139–154,1985.

[3] Forstmann, S., Kanou, Y., Ohya, J., Thuering, S., Schmitt, A, “Real-time stereo by using dynamic programming.” IEEE Conference on Computer Vision and Pattern Recognition Workshop, Volume 3, pp.29, 2004.

[4] Wang, L., Liao, M., Gong, M., Yang, R., Nister, D, “High-quality real-time stereo using adaptive cost aggregation and dynamic programming.” International Symposium on 3D Data Processing, Visualization and Transmission,pp.798–805,2006.

[5] Veksler, O, “Stereo correspondence by dynamic programming on a tree.”IEEE Conference on Computer Vision and Pattern Recognition, Volume 2, pp.384–390, 2005.

[6] Hirschmuller, H., “Improvements in real-time correlation-based stereo vision.” Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). In Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision, pp.141-148,2001.

[7] M.L. Gong and Y.H. Yang, “Fast stereo matching using reliability-based dynamic programming and consistency constraints”, Ninth IEEE International Conference on Computer Vision, vol.1, pp.610 – 617, 2003.

[8] Kim, J.C., Lee, K.M., Choi, B.T., Lee, S.U, “A dense stereo matching using two-pass dynamic programming with generalized ground control points.” IEEE Conference on Computer Vision and Pattern Recognition, Volume 2, pp. 1075–1082,2005.
[9] J. Salmen, M. Schlipsing, J. Edelbrunner, S. Hegemann, and S. Lueke. “Real-time stereo vision: making more out of dynamic programming.” CAIP 2009: COMPUTER ANALYSIS OF IMAGES AND PATTERNS, volume 5702, pp.1096– 1103, 2009.
[10] Y. Deng and X. Lin. “A fast line segment based dense stereo algorithm using tree dynamic programming.” ECCV 2006: 9th European Conference on Computer Vision, partIII, pp.201-212, 2006.
[11] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, pp.7–42, 2002.
[12] M. F. Tappen and W. T. Freeman, “Comparison graph cuts with belief propagation for stereo, using identical MRF parameters,” International
Conference on Computer Vision, 2003.
[13] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Understanding belief propagation and its generalizations,” Mitsubishi Electric Research Laboratories, Jan. 2002.
[14] S.-B. Lee, K.-J. Oh, and Y.-S. Ho, “Segment-based multi-view depth map estimation using belief propagation from dense multi-view video,” 3DTV
Conference, May 2008.
[15] S. Larsen, P. Mordohai, M. Pollefeys, and H. Fuchs. “Temporally consistent reconstruction from multiple video streams using enhanced belief propagation,” Eleventh International Conference on Computer Vision (ICCV), pp.1-8, 2007.
[16] A. Klaus, M. Sormann and K. Karner. “Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure**,”** 18th International Conference on Pattern Recognition (ICPR), pp.15-18, 2006.
[17] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. “Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.492-504, 2008.
[18] J. Sun, Y. Li, S.B. Kang, and H.-Y. Shum. “Symmetric stereo matching for occlusion handling,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2 , pp.399-406, 2005.
[19] L. Nalpantidis and A. Gasteratos. “Stereo vision for robotic applications in the presence of non-ideal lighting conditions,” Image and Vision Computing, pp.940-951, 2010.

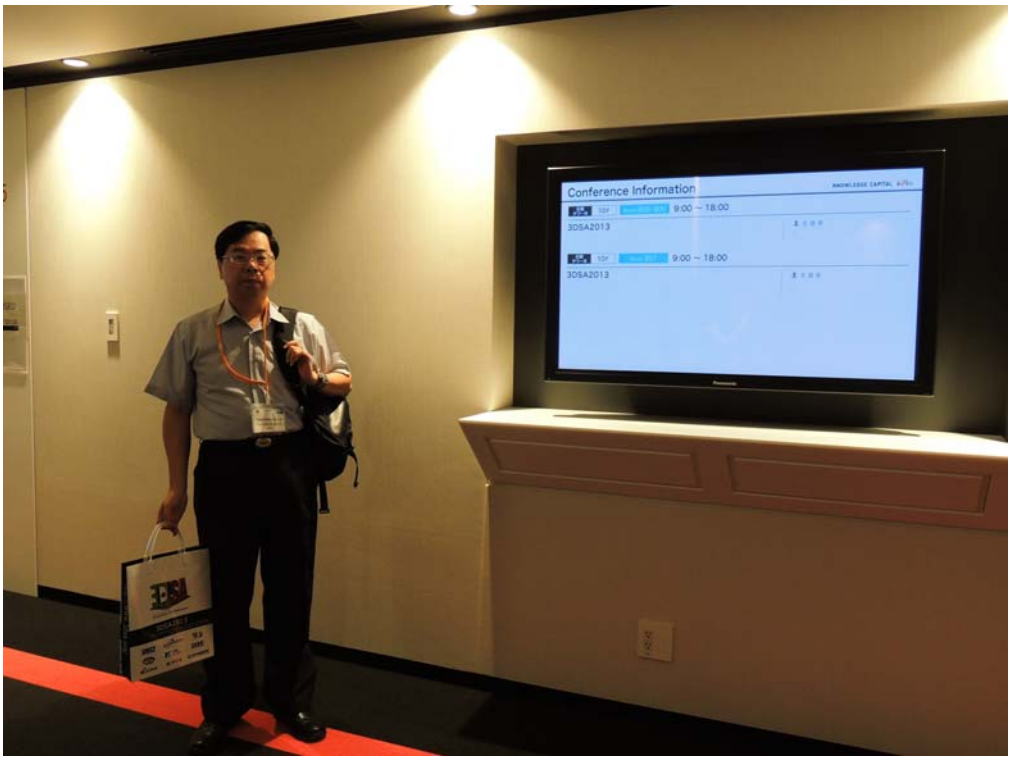
中華大學補助教師出席國際性及大陸地區海峽兩岸學術會議報告

102 年 7 月 8 日

報告人姓名	鄭芳炫	系所職稱	中華大學資工系教授
時間 會議 地點	2012/6/26~2013/6/28 日本大阪	本會核定 補助文號	(101)中華研國字第 043 號 NSC101-2221-E-216-034
會議名稱	(中文) 2013 三維系統及應用國際研討會 (英文) 2013 International Conference on Three Dimensional Systems and Applications		
發表論文題目	（中文） （英文）Depth Correction by Minimizing Energy of Neighboring Regions		

一、參加會議經過

本次會議在日本大阪舉行，由於接近暑假期間已無本國直航的班機，因此為配合會議時間，只好搭乘日本樂桃航空之班機。本人於 6 月 23 日搭日本樂桃航空上午的班機經飛行大約三小時抵達日本關西機場，接著搭日本 JR 火車經大阪直達京都，於下午 3 點半抵達京都，進行私人參訪行程。6 月 25 日下午從京都車站搭日本 JR 火車約 30 分即直達大阪車站，隨即入住飯店休息，並準備明天起一連三天的研討會。26 日一早便前往會議地點 GrandFront Osaka，此區域為日本建築大師安藤忠雄設計，前後共有四幢大樓連成一氣，建築十分雄偉確有大師的風範。由於主辦單位並沒有清楚的指示標誌，竟然花了將近一小時才找到會議的地點，隨即註冊(見圖一)。本人在領取了會議資料完成註冊手續後隨即進入會場參加會議。



圖一

會議的開幕典禮由主辦單位與會議的委員會主席簡單的致歡迎詞後，隨即展開。由於本會議為一個專業的中小型研討會，因此會議議程的安排有別於一般的研討會。會議第一天主要以專題演講及邀請演講為主，第二天則是安排論文的口頭發表(Oral)，分二個場地進行，第三天則是張貼論文的發表及實驗室的參觀。本屆會議參加人數為 130 人，共有超過 5 個國家 122 篇論文投稿, 每篇論文都經過二位評審審查後, 最後通過 99 篇論文，包括 49 篇口頭發表論文及 50 篇張貼論文。此外會議於第一天安排了一場主題專題演講(Keynote Speech)及四篇邀請演講(Invited talk)，第一天之主題專題演講由日本東京大學的教授 Prof. Hirose 主講。第三天也安排了一場主題專題演講，由韓國 Aachen 大學的教授 Prof. Ohm 主講。這二位教授在各自領域均學有所成，演講內容亦十分精彩，因此可說是收獲良多。

本人之論文『Depth Correction by Minimizing Energy of Neighboring Regions』被安排在第二天的上午Section 5之場次發表，同時也主持了前一場次Section 3的議程，如圖二所示。本次會議尚有許多台灣之其他論文發表，經過三天完整會議研討，與會者均有豐富的收穫。詳細論文報告場次規劃可參考議程表。此外會議主辦單位特別在第二天晚上安排一個晚宴，使與會者之間有互相交流的機會，見圖三。

圖二

圖三

二、與會心得

此次會議的主辦單位為 NIST，會議地點就在大阪 JR 車站旁新開放的 GrandFront Osaka，這樣的安排應該是考慮到來自世界各國與會者的方便，不過若是能安排在校園內，應該更能感受學術交流的氣息。本會議是中型的研討會, 但定位上仍是以專業精緻之研討會自許，與一般大雜燴式之大型研討會不同。主要目的是讓與會之學者能真正達到充份的學術交流，而不是走馬看花。三天的會議安排得十分緊湊，每天都是從上午 9:00 至下午 18:00 止。本會議在第二天晚上安排的晚宴中除了報告此次會議的相關數據外也頒發了最佳論文獎，同時亦宣佈下一屆在韓國首爾舉辦。

國際研討會是學術研究交流很好的場合，可結合全世界相同研究領域的學者互相切磋。主辦單位除了安排專題演講及論文發表的議程外，若能安排半天的行程到會議主辦大學做參訪，應該更能達到交流的目的。

三、考察參觀活動(無是項活動者省略)

本會議的定位是專業精緻之研討會，三天的會議安排得十分緊湊，每天都是從上午 9:00 至下午 18:00 止。為了達到進一步交流的目的，因此主辦單位在下午安排了一場實驗室參訪之旅(Lab tour)，主要是參觀全世界最大尺寸的 200 吋 3D 裸視顯示器。

四、建議

每次參加研討會常常會在會場碰到許多台灣去的學者教授，若在出國前就可以互相聯繫一起出席，不僅在費用上可以比較節省，在會議上也可以整合力量為台灣之學術界出聲讓國際能充份了解台灣在學術領域之實力。本次會議台灣也是主辦單位之一，共有超過 5 個國家的研究學者參加，台灣大概有十幾位教授及學生參加，除本校中華大學外，尚有台北科技大學、成功大學、雲科大、工研院等。也許國科會可以在現有之網站上另闢一個出席國際會議之交流園地，讓國內之研究學者可以互通訊息，不僅可以整合大家的力量，也可知道國內在國際學術界之活動能量。

五、攜回資料名稱及內容

本次會議攜回一本紙本的會議導引手冊，另在環保與預算的考量下，本次研討會並不印製紙本的論文集，不過大會提供了隨身碟，內容為本次會議的所有論文集。

六、其他

Depth Correction by Minimizing Energy of Neighboring Regions

Fang-Hsuan Cheng & Yu-Pang Chang
Dept. of Computer Science & Information Engineering
Chung Hua University
707, Sec. 2, Wu-Fu Rd., Hsinchu 300, Taiwan
fhcheng@chu.edu.tw

***Abstract*—**It has been proposed in this paper an idea of correcting depth map obtained according to local stereo matching. In this paper, energy was calculated based on the entire image, meanwhile, energy minimization concept was adopted, and the area obtained according to color segmentation algorithm was adopted too. The color feature and depth value among different regions and their neighboring regions are used to define the relation between the smooth and occluded regions in the energy function. Then the region energy was calculated repeatedly until the change was insignificant or the number of iterations was reached. From the experimental result, it is proved that the depth map after correction showed better object shape and depth dense sense.

Keywords*-Stereo matching; Color segmentation; Plane fitting; 3D vision*

I. INTRODUCTION

In recent years, depth estimation based on stereo matching was a hot research topic in image processing. Lots of scholars had proposed estimation methods to achieve more accurate depth estimation result. In stereo matching method, it can be mainly divided into two categories of local matching and global matching.

Local matching method was a stereo matching method that was proposed earliest. It was a fast method that takes low cost. When a pair of left and right view is entered, they must be calibrated by the coordinate of the camera. By means of pixel basis, the same row of the left and right view is searched with the most similar point, and the reference point can be located at the left or right view. The disparity can be found from the shift pixels between reference point and the most similar point, then after normalize on the disparity, the depth value can be obtained. The method of searching similar point is the key part of local stereo matching method. The search of similar pixel is to calculate the reference pixel in the left image and the similar pixel in the right image is from the same horizontal line that has the smallest error to this reference pixel using a window based SSD (Sum of Squared Difference) or SAD (Sum of Absolute Difference). The basis of error calculation can be based on color spaces (RGB, Lab, YCbCr and HSV) or grey values, etc. The window size could be NxN, or window of uncertain shape. Therefore, if there are very similar but not expected similar points in the reference image, matching error will then be generated. Therefore, local matching result could easily generate lots of noises, meanwhile, the occluded region will be difficult to be processed, and the pixel point in the occluded region does not exist in the reference image. In [1], global matching architecture was used, better processing effect will be obtained on noise point and occluded region, however for image with more complicated texture, good depth estimation was still difficult to be obtained.

Global stereo matching [2-3] showed in recent years higher enhancement on the depth quality of occluded region and non-texture region, among them, segmentation based method [4-6] showed better result on non-texture region. In these methods, first, the image was performed with color segmentation, then faster local matching method was used to get rough initial disparity. The plane model was done on the segmented region and plane fitting conducted too, hence, the estimation reliability on the non-texture region can be enhanced. However, in the plane fitting process, it could easily be affected by mistaken estimation or noise point, hence, effective release method was proposed in this paper.

II. DEPTH CORRECTION METHOD

The system architecture of this paper was depicted in Figure 1, and the image segmentation used was Mean-Shift image segmentation, meanwhile, energy calculation must be referred to the right view at the same time so as to get reasonability on the depth matching.

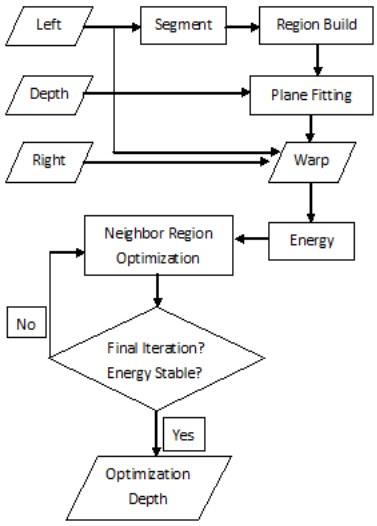


Figure 1. Flow chart of the proposed algorithm

A. ***Mean-Shift Image Segmentation***

In this paper, mean-shift [7] image segmentation was adopted, and the color distance and space widow size parameter values of color image were entered to perform pixel cluster classification and to achieve the goal of segmenting the image. Since stereo matching method based on region hypothesized that the boundaries of regions of different colors will be consistent with boundaries of discontinuous depths, hence, in order to satisfy this hypothesis as much as possible, we have segmented as many regions as possible. However, the region cannot be too small so as not to affect the calculation efficiency and reliability. Too many regions will lower the calculation efficiency and will let too many stray pixels regions affect the neighboring regions; however, if the number of region is too small, different region of similar color will be merged into the same region, which will result in boundary consistency of larger error.

B. ***Least Square Plane Fitting***

Refer to [8], a disparity plane model of the segmented region is represented by function as follows:

$$d(x,y) = ax + by + c \tag{1}$$

(x,y) is image coordinate, a,b,c are plane parameters, $d(x,y)$ is the disparity value calculated according to plane parameters, hence, in order to fit all the points in the region, the least square method (see Eq.(2)) as proposed in [8] was used to fit the plane in each region:

$$\begin{bmatrix} \sum_{i=0}^{N} x_i^2 & \sum_{i=0}^{N} x_i\, y_i & \sum_{i=0}^{N} x_i\, d_i \\ \sum_{i=0}^{N} x_i\, y_i & \sum_{i=0}^{N} y_i^2 & \sum_{i=0}^{N} y_i\, d_i \\ \sum_{i=0}^{N} x_i\, d_i & \sum_{i=0}^{N} y_i\, d_i & \sum_{i=0}^{N} d_i^2 \end{bmatrix} = 0 \tag{2}$$

N is the number of points within the region, and x_i, y_i are pixel image coordinates within the region, and d_i is the corresponding disparity value. The parameter values obtained according to least square method approximate a plane for all the disparity values within the region, and this matches the hypothesis that all the disparity values within the region must be smooth.

Since least square plane fitting method is to construct the plane model on all the depth values of the region, it can be obviously seen that in the original depth estimation, there are some stray points (noise point or wrong estimation depth) which will affect the entire result after fitting the plane. If there are lots of stray points in the region, it will strongly affect the reliability of the plane model made by plane fitting. Therefore, a mechanism to detect and remove stray points has been proposed to avoid such issue and to enhance the reliability of the least square method. Make occurrences frequency statistics on the initial depth value within the region as below:

$$stray(x,y) = \begin{cases} 1, & If \ \ (|d(p) - d_{MAX}|)/d_{MAX} \geq Ts \\ 0, & Otherwise \end{cases} \tag{3}$$

$d(p)$ is the occurrences frequency of the depth value within the region, d_{MAX} is the highest occurrences frequency of the depth value within the region, $0 < Ts < 1$ is the threshold value to control the stray point. Hence, if it is set up too high, the stray points will be difficult to be removed, but if it is set up too low, lots of reliable points will be removed. If $d(p)$ and d_{MAX} are too divergent, then that point will be treated as stray point, Ts is set up as 0.2 according to experimental result.

C. ***Energy Minimization based on Neighboring Region***

During the process of performing neighboring region correction, in order to evaluate whether the change of disparity plane parameter of the current region i is reasonable or not, it is needed to define an energy function E_i as below:

$$E_i = E_{data} + E_{occ} + E_{smooth} \tag{4}$$

E_{data} is data energy, which is used to represent the similarity of the mapping of left image to the right image according to depth value, and is defined as below:

$$E_{data} = \sum_{P \in V_W, Q \in R} |G(P_x, P_y) - G(Q_x, Q_y)| \tag{5}$$

$P \in V_W$ represents, when the left image is converted into disparity value according to the current depth value, the pixel set for mapping the pixel point to the right image according to disparity value, and visible constraint should be matched [5]. That is, when the pixel point of the left image is mapped into the right image and if the target location is mapped repeatedly, then the pixel point with higher disparity value is visible, and others are invisible; $Q \in R$ is the pixel set of the right image; $G(x)$ is the grey value of that point. E_{occ} is occluded energy defined in Eq.(6), which makes reasonable performance possible when left image is mapped to the occluded region on the right image according to depth map.

$$E_{occ} = \left(\left|Occ_L\right| + \left|Occ_R\right|\right)\lambda_{occ} \tag{6}$$

When left image is mapped to the right image, if pixel is mapped repeatedly, the pixel number of that situation will be represented by Occ_L, which is because the region the pixel is located is covered by the right region; on the contrary, if there is hole between two regions, the pixels within the hole is represented as Occ_R, and here λ_{occ} represents occluded penalty parameter.

E_{smooth} is smooth energy defined in Eq.(7) which represents boundary energy similarity between regions.

$$E_{smooth} = \sum_{P\in B, Q\in N} \begin{cases} \lambda_{smooth}, & If \ |d(P)-d(Q)|>2 \\ 0, & Otherwise \end{cases} \quad (7)$$

B represents boundary pixel set of the region in left image, N represents the boundary pixel set on other regions that are neighboring to B, and $P \in B$ and $Q \in N$ are four-connected two neighboring pixel points. $d(P)$ and $d(Q)$ are the disparities of pixel P and Q, λ_{smooth} is smooth penalty parameter, $|d(P)-d(Q)|>2$ is used to judge if certain boundary point in the current region is discontinuous disparity point, and what is important is, pixel P should not be occluded pixel.

It is supposed that after plane fitting, each region still needs further merging. As shown in Figure 2, region A_{10} has seven neighboring regions, and we have used A_{10}, according to the plane parameters of seven neighboring regions: $A_{10,n}$, $n=\{1,2,3,4,5,6,9\}$ to calculate respectively depth values and the energy change as well. Among seven neighboring regions, those which can make minimized energy on A_{10} and entire image is targeted for correction. After visiting all the regions, each region and the correction target is combined and the energies of all the regions are calculated, then the next iteration is continued until energy does not change or the iteration number is reached.

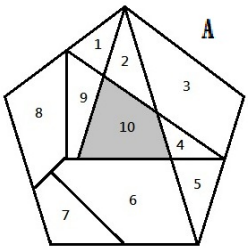


Figure 2. Neighboring Region Correction

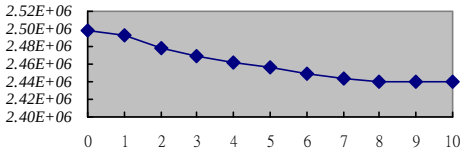
III. EXPERIMENTAL RESULTS

The proposed method is realized through Borland C++ 6.0, and the Middlebury stereo image that is widely adopted as assessment standard is used. For the parameter setup aspect, the color distance and space widow size for color image segmentation are set as 5 and 7, λ_{occ} and λ_{smooth} are set as 50.

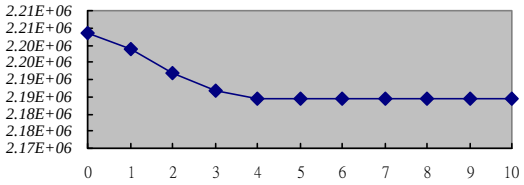
Figure 3 is energy descending curve, wherein only Teddy and Cones are performed with iteration method because the depth distribution and complexity of Tsukuba and Cones are relatively few, and there is no significant improvement on the result after iteration. For Teddy's iteration, it can be found that after four iterations, energy has descended to stable value, for Cones' iteration, stability can be obtained only after eight iterations. The reason is because in Cones, the region and depth change is relatively divergent, and stability can only be achieved after more iteration.

Figure 4 is the error matching between the estimated depth and real depth, wherein black and grey pixel represents the case with depth error and occlude pixel error larger than 1. At the edge of depth change, image segmentation treatment can bring very good improvement to object edge and the edge of depth change, and higher consistency can then be achieved, which brings more natural performance to the result after DIBR (Depth-image-based-rendering). The depth in the error region can be corrected according to correct depth in neighboring region, and eventually, the error rate in the smooth region can be improved.

Table 1 shows the error rate of estimated depth compared to ground truth between initial depth and improved one by the proposed method. Entirely, very good improvement can be obtained on the error rate of the edge of the depth change and smooth region. Hence, the treated depth image will have performance closer to human eye in the result after DIBR. Some experimental results of the proposed correction method for the standard images from Middlebury database are shown in Figure 5.

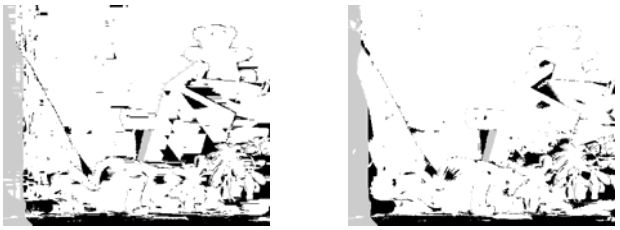


(a) Cones energy curve



(b) Teddy energy curve

Figure 3. Energy curve



(a) Initial depth error (b) Depth error by the proposed method

Fig. 4 Depth error Comparison between initial and the proposed method.

IV. CONCLUSIONS

This paper has presented iterative correction method according to neighboring region, which can get better depth reference from the neighboring region and in turn improve higher region error in the original depth estimation, meanwhile, it let the result screen have higher reliability of depth change edge and better smooth region density as compared to that of the initial depth map. Besides, the proposed stray point removal can raise the reliability of plane fitting.

This proposed method is constructed based on color image segmentation, which is quite sensitive to segmentation region

boundary, hence, in the future, the reliance on the regional boundary should be further reduced.

ACKNOWLEDGMENT

We would like to thank the National Science Council of Taiwan for their kind support to this research with grand no. NSC101-2221-E-216-034.

REFERENCES

[1] M. Gerrits, P. Bekaert, "Local Stereo Matching with Segmentation-based Outlier Rejection," The 3rd Canadian Conference on Computer and Robot Vision, 2006, pp.66.

[2] Jian Sun, Nan-Ning Zheng, Heung-Yeung Shum, "Stereo matching using belief propagation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, pp.787 - 800.

[3] M. Gerrits, P. Bekaert, "Local Stereo Matching with Segmentation-based Outlier Rejection," The 3rd Canadian Conference on Computer and Robot Vision, 2006, pp.66.

[4] Jian Sun, Nan-Ning Zheng, Heung-Yeung Shum, "Stereo matching using belief propagation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, pp.787 - 800.

[5] P. F. Felzenszwalb, D. R. Huttenlocher, "Efficient belief propagation for early vision," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, Vol. 1, pp. I-261 – I-268.

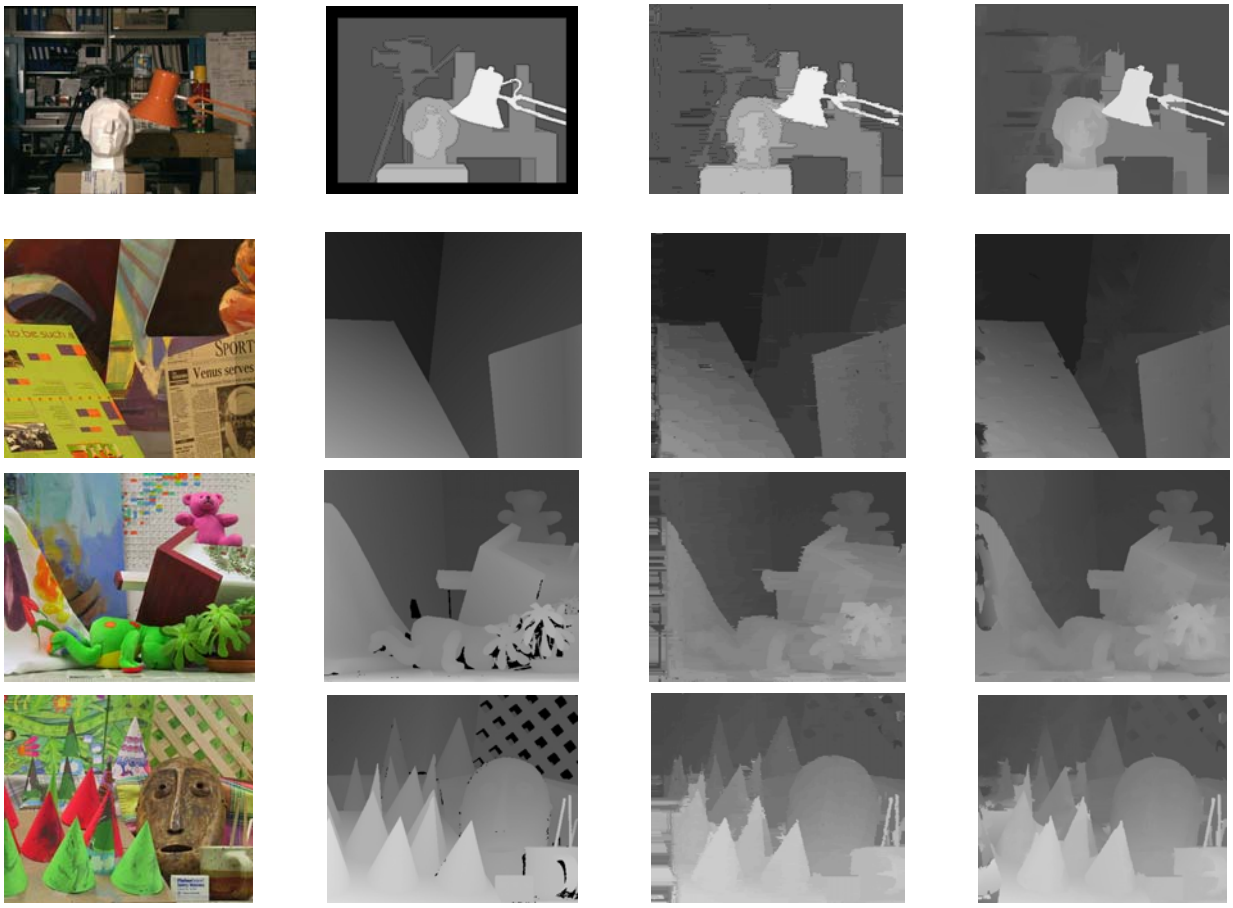
[6] Li Hong, G. Chen, "Segment-based stereo matching using graph cuts," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, Vol. 1, pp. I-74 – I-81.

[7] M. Bleyer, M. Gelautz, "A layered stereo algorithm using image segmentation and global visibility constraints," International Conference on Image Processing, 2004, vol. 5, pp.2997–3000.

[8] Zeng-Fu Wang, Zhi-Gang Zheng, "A region based stereo matching algorithm using cooperative optimization," IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp.1–8.

TABLE 1. Error rate of estimated depth compared to ground truth from Middlebury database

	Tsukuba			Venus			Teddy			Cones			Avg. Bad
	Non-occ	All	Disc	Non-occ	All	Disc	Non-occ	All	Disc	Non-occ	All	Disc	
Proposed	2.32%	2.58%	10.9%	0.59%	1.01%	5.98%	11.6%	18.4%	27.1%	8.63%	14.9%	20.2%	10.4%
Initial	2.54%	3.22%	12.7%	1.98%	2.83%	11.1%	13.8%	20.9%	29.3%	9.61%	16.4%	21.2%	12.1%



(a) Reference figure (b) Actual depth (c) Initial depth (d) Depth correction result

Figure 5. Experimental results of the proposed correction method for the standard images from Middlebury database.

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

102 年 9 月 24 日

報告人姓名	鄭芳炫	系所 職稱	中華大學資工系教授
時間 會議 地點	2012/9/14~2013/9/17 日本熊本	本會核定 補助文號	NSC101-2221-E-216-034
會議 名稱	(中文) 參加第八屆創新計算資訊暨控制國際研討會 (英文) The Eighth International Conference on Innovative Computing, Information and Control		
發表 論文 題目	（中文） （英文）A NOVEL ALGORITHM FOR REAL-TIME HAND TRACKING		

一、參加會議經過

本次會議在日本九州之熊本舉行，由於本國航空並無直飛熊本之班機，因此搭乘華航直飛日本福岡之班機，再搭乘日本鐵路(JR)由福岡至熊本。本人於 9 月 13 日搭中華航空下午的班機經飛行大約二小時抵達日本福岡機場。由於日本時差早台灣一小時，到達福岡時已是下午五點多，因此先在福岡博多車站附近的旅館住宿休息一晚。由於日本交通費用很高，為了節省長途運輸之交通費，便購買了日本為外籍遊客發行之週遊卷，可連續五日在北九州地區無限次數使用日本鐵路，費用只要九仟日幣。9 月 14 日下午從福岡博多車站搭日本 JR 新幹線約 40 分即直達熊本車站，隨即再搭市區電車到達會議地點熊本市國際交流會館(Kumamoto City International Center)註冊，見圖一。註冊後隨即登記入住飯店休息，並準備明天起一連三天的研討會。

圖一

會議的開幕典禮由主辦單位與會議的委員會主席簡單的致歡迎詞後，隨即展開。由於本會議為一個專業的中型研討會。會議第一天主要是註冊及歡迎酒會(Welcome Party)，第二、三天則是安排專題演講及論文的口頭發表(Oral)，分六個場地同時進行，第四天則是安排參訪行程。本屆會議共有超過 10 個國家參加，註冊人數為 283 人，每篇發表之論文都經過二位評審審查。會議中之二個專題演講分別由美國華盛頓大學教授 Prof. Ramesh Agarwal 主講，講題為“Feedback Control of Simple Model Systems of Climate Dynamics”及日本 Kyushu Institute of Technology 的教授 Prof. Takeshi Yamakawa 主講，講題為“Glimpse of Fussy through a Billow of Traditional Digital Systems – can Professional Medical Doctors be Replaced with Digital Computers?”。這二位教授在各自領域均學有所成，演講內容亦十分精彩，因此可說是收獲良多。

本次會議規畫的內容十分豐富，在同一時間有6個track同時舉行，研討主題包含如下:

Mechanics Analysis between Human Injuries and Signals
Management and Process Science in Service Computing
Optimization and Model
Control System and Applications
Innovative Computation and Its Applications
Advanced Multimedia Information and Human Interaction Technology
Intelligence Systems and Applications
Neural/Fuzzy Control
Management Science
Frontier Information Technology
Soft Computing
Industrial Systems
Data Processing
Recognition/Classification
Image Processing
Signal Processing
Natural Language Processing
Knowledge Discovery
Circuit System

本人之論文『A NOVEL ALGORITHM FOR REAL-TIME HAND TRACKING』被安排在第二天的下午之場次發表，主要探討如何在高度自由甚至手和臉或其它類似膚色的物件重疊的環境下進行手勢追蹤，如圖二所示。本次會議尚有許多台灣之其他論文發表，經過四天完整會議研討，與會者均有豐富的收穫。詳細論文報告場次規劃可參考議程表。此外會議主辦單位特別用心，分別在第一天晚上安排一個歡迎酒會，第二天晚上安排一個晚宴，第三天晚上則安排了一個歡送酒會，第四天則安排參訪活動行程，使與會者之間有更多互相交流的機會。

圖二

二、與會心得

此次會議的主辦單位為日本東海大學 Tokai University，會議地點就在熊本公車總站旁的國際交流中心，這樣的安排應該是考慮到來自世界各國與會者的方便，不過若是能安排在校園內，應該更能感受學術交流的氣息。本會議是中型的研討會, 但定位上仍是以專業精緻之研討會自許，與一般大雜燴式之大型研討會不同。主要目的是讓與會之學者能真正達到充份的學術交流，而不是走馬看花。四天的會議安排得十分緊湊，每天都是從上午 9:00 至下午 18:00 止。為了讓參與研討會之各國學者有更多的交流，本會議在第一天晚上舉辦一個歡迎酒會，第二天晚上安排了一個正式的晚宴，第三天晚上則舉辦一個歡送酒會，同時宣佈下一屆在韓國舉辦。本次研討會台灣也有許多教授及學生參加，除本校中華大學外，尚有高雄應用科技大學、建國科技大學、中央大學、海洋大學、暨南國際大學等。

三、考察參觀活動(無是項活動者省略)

本會議的定位是專業精緻之研討會，四天的會議安排得十分緊湊，每天都是從上午 9:00 至下午 18:00 止。為了達到進一步交流的目的，因此主辦單位特別在第四天安排了一個參訪行程，讓與會者除了參觀熊本附近之風景名勝外，也讓與會之學者在輕鬆的氣氛下彼此交換研究心得。由於熊本市政府也大力贊助此次研討會，因此參訪行程中市政府特別派了一位隨行攝影記者全程記錄過程，也顯示熊本市政府對此次研討會重視的程度。

四、建議

國際研討會是學術研究交流很好的場合，可結合全世界相同研究領域的學者互相切磋。主辦單位除了安排專題演講及論文發表的議程外，若能安排半天的行程到會議主辦大學做參訪，應該更能達到交流的目的。

五、攜回資料名稱及內容

在環保與預算的考量下，本次研討會只提供一片會議論文集光碟，內容為本次會議的所有論文集。

六、其他

ICIC Express Letters
Part B: Applications
Volume **5**, Number **1**, February **2014**

ICIC International ©2014 ISSN 2185-2766
pp. 95–101

A NOVEL ALGORITHM FOR REAL-TIME HAND TRACKING

YEA-SHUAN HUANG, YU-CHUNG CHEN AND FANG-HSUAN CHENG

Department of Computer Science and Information Engineering
Chung-Hua University
No. 707, Section 2, Wufu Road, Xiangshan District, Hsinchu City 300, Taiwan
{ yeashuan; gibobo831521; fhcheng }@chu.edu.tw

Received March 2013; accepted June 2013

ABSTRACT. *This paper proposes a novel algorithm for real-time hand tracking, and this algorithm can successfully track a hand even when it is overlapped with other objects during tracking. Three situations (separation, proximity and overlap) between tracked objects are defined. A separation template image of the tracking hand is created in the state of proximity, and a feature-point-based matching comparison is conducted in the state of overlap. The experimental results show the proposed algorithm has highly accurate detection results and is robust to overlapped objects. In the running stage, the proposed algorithm reaches at 30-45 frames per second in real time.*
Keywords: Hand detection, Hand tracking, Skin color learning, Edge difference image, State detection, Separation template

1. **Introduction.** Because hand gesture is a commonly used communication method among people, it is essentially friendly and important if a user can interact with devices through detecting and tracking his/her hand gestures directly. Therefore, a lot of researches have been proposed, which mainly fall into two categories, i.e., glove-based and vision-based methods. Glove-based methods require a user to wear electronic gloves so that a device can sense the glove positions and perform appropriate response. This kind of methods is quite mature but is inconvenient and unnatural because of requesting extra accessories. In the vision-based methods, Lin [1] used 7400 images containing skin color data to generate a Gaussian-based skin-color probability model. This model identified fingers, knuckles, and finger rifts in hand images with simple background by means of feature comparison and then generated hand information through geometry comparison. Han [2] proposed a method which used a generic skin color model to generate training data and an SVM-based skin color classifier which, when used in conjunction with a JSEG [3] area analysis algorithm, could identify the contour of skin color objects; and Kalman filter [4] was then used to estimate the position of object contour. He also proposed setting up different search areas to solve the problem of hand disappearance at boundary or the problem of object overlapping. Pan [5] used a Bayesian probability model for skin color training. In his study, off-line learning of skin color samples was used in conjunction with online skin color learning to control the reliability ratio. Gesture tracking was carried out after having detected an initiating gesture; KLT [6] feature tracker was adopted to judge the moving behaviors of the feature points of a tracked hand object; and then hand position was determined according to feature point clustering. Sudderth et al. [7] developed probabilistic methods for visual tracking of a three-dimensional geometric hand model from monocular image sequences. A prior model was defined to enforce the kinematic constraints implied by the model's joints which has a local structure and is a pairwise Markov random field. Given a graphical model of hand kinematics, the hand's motion was tracked by using a nonparametric belief propagation (NBP) algorithm

which approximates the posterior distribution over hand configurations as a collection of samples.

However, it is still very difficult when the tracked hand overlaps with other skin-color objects. In order to resolve this issue, a novel algorithm is proposed by taking initiative to separate the tracked hand object through the application of feature point extraction and separation template designs. In this paper, the initial hand position of each video clip is given manually and only the method of hand tracking is discussed in detail.

2. Hand Tracking.

For hand tracking, a Kalman filter is used to estimate the user's hand position of the incoming frame according to the previous stored hand object information (position, size and skin color model), and the estimated position is used to establish a proper hand search area for skin color detection. Then, the connected component analysis operation is carried out on the detected skin color binary image to identify the positions and ranges of skin color objects. Afterwards, the state of the tracked hand object is determined and the object is subject to different processing approaches according to its state.

2.1. States detection.

A user's hand movement may give rise to overlap between his/her hand and other skin color objects and this in turn will cause tracking difficulties. However, if the interrelationship between the hand and other skin color objects can be known in advance and proper preparation has been made before the occurrence of overlapping, then the tracked hand object and other skin color objects still can be separated effectively when they do overlap. Through skin color detection, the positional relationship between the tracking hand object and other skin color objects can be obtained. Generally speaking, positional relationship between two objects is one of the three states, separation, proximity and overlap. The separation state means the tracked hand is away from other skin-color objects, and the proximity state means the hand object is close to at least one of other skin-color objects but is not really overlapped by others. And the overlap state means the tracked hand region is overlapped with other skin-color objects.

Suppose R_t^i is the range of the ith skin color object in I_t, O_t is the range of the currently tracked hand object, and C_t is the range of a non-tracking skin color object. During the tracking process, the overlap ratio B_i of the range of the ith object R_t^i in I_t and the range of the tracked hand object O_{t-1} in I_{t-1} are calculated one by one. The skin color object i^* with the highest overlap ratio is taken as the current hand tracking object. In order to detect the interrelationship between $R_t^{i^*}$ and the ith non-tracking skin-color object R_t^i $(i \neq i^*)$, an inspection range D_t^i from R_t^i is established according to the following relation.

$$D_t^i = \left(R_t^i \cdot x - \frac{R_t^i \cdot w}{2}, R_t^i \cdot y - \frac{R_t^i \cdot h}{2}, 2 \times R_t^i \cdot w, 2 \times R_t^i \cdot h \right) \tag{1}$$

The first two elements of D_t^i form the top-left coordinate of D_t^i and the last two elements of D_t^i are the width and the height of D_t^i respectively.

As shown in Figure 1, the interrelationship between two objects can be determined by checking whether the inspection range $D_t^{i^*}$ overlaps with the inspection range of the D_t^i or not. If $D_t^{i^*}$ does not overlap with any D_t^i $(\forall\, i \neq i^*)$, then it represents the tracking object is apart from all the non-tracking objects and can be easily tracked in the next image. If $D_t^{i^*}$ overlaps with a certain D_t^i, then it means that the tracking object is in a proximity state with the ith non-tracking object. This indicates that the tracking object is easy to overlap with some non-tracking objects in subsequent frames. When two objects are close to each other, we will take the range of the i^* object rectangle as the range of current hand tracking object rectangle (i.e., $O_t = R_t^{i^*}$), and the range of the ith object rectangle as the range of the approaching non-tracking object rectangle (i.e., $C_t = R_t^i$), and will use the image of C_t to establish a separation template, which is used to record the image texture information of the approaching non-tracking object. If O_t

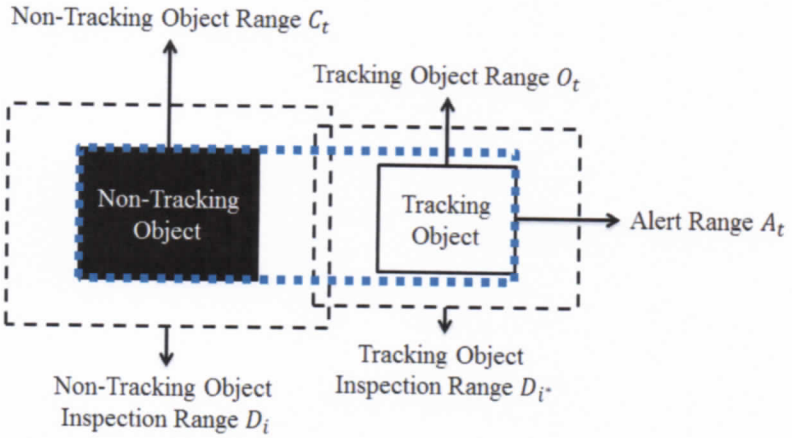


FIGURE 1. A schematic diagram of defined ranges between objects at the proximity state

and C_t do overlap in the future, the pre-established separation template can be used for feature point separation. Furthermore, in order to judge whether the state is changed from proximity to overlap or separation, we design an alert range A_t, which is used to record the minimum circumscribed rectangle between hand tracking object O_t and the approaching non-tracking object C_t. Let A_{left}, A_{top}, A_{right} and A_{bottom} be the coordinate values of the leftmost, topmost, rightmost, and bottommost positions of the alert range A_t. The criterion for determination of overlapping state is the ratio OA of the rectangle area $Area(O_t)$ of the tracking hand object to the rectangle area $Area(A_{t-1})$ of the alert range in the previous image. When overlap happens, the area of the tracking object O_t is identical to the area of the overlapped objects; therefore, the overlapping ratio OA of O_t to the alert range A_{t-1} will be a large one; on the contrary, if no overlap happens, then the OA value will be relatively small. Based on experiment results, 0.7 is chosen to determine the overlap state. If OA is greater than or equal to 0.7, then it can be judged the tracking object is in an overlap state. When the two objects are in the overlap state, the range of the overlapping object will be taken to be A_t.

2.2. **Separation template establishment.** If a tracking object is in approaching state, the system will continuously establish a separation template for each successive frame until its state is changed. The separation template contains two information items, i.e., separation range R_* and comparison image I_*. Separation range is the non-tracking object range C_t to which the tracking object is approaching; a comparison image is the image in the separation range at current moment, which is designed to retain the complete features of the non-tracking object range C_t to which the tracking object is approaching before overlapping. When a true overlap happens, image comparison can serve as the basis for the calculation of feature point matching operations.

2.3. **Feature point separation.** When the tracking object enters an overlap state, the system will, within the rectangle area O_t of the currently tracking object, perform feature point extraction and carry out block matching operation on each extracted feature points sequentially using the separation template established previously when the tracking object was in the proximity state. In fact, in addition to the optimal corresponding position of every feature point, the optimal matching score can also be identified from the separation template image. The matching score can also serve as the basis for feature point classification. The feature point classification herein may yield two results, i.e., a feature point either belong to or does not belong to the separation template established by a non-tracking object. If a feature point does come from a non-tracking object, it will have a relatively high matching score because the recorded separation template takes

the non-tracking object before overlapping as the recording target. On the contrary, if a feature point comes from the tracking object, its matching score in general is relatively low because no information on the tracking object is recorded in the separation range R_*. Therefore, so long as an appropriate matching score threshold T_D has been selected, feature points with their matching scores higher than T_D can be judged to be of a non-tracking object forming the separation template while those with matching scores lower than T_D can be judged to be of the tracking object. Figure 2 is an example of feature point separation results, where the red triangles are the feature points of the tracking object while the blue circles are the feature points of non-tracking objects.

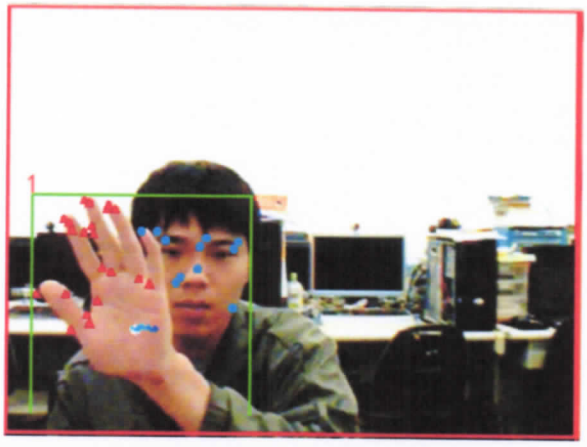


FIGURE 2. Result of feature point separation

3. **Refining Palm Center.** A distance transform (DT) image of the tracking hand object is used to locate the palm center position, which is the point with the maximum distance to the object's edge. However, when the hand object overlaps with other non-tracking skin color objects, it is quite probable that the identified palm center point may be a wrong one because the overlapped non-tracking object may have a large area such as a human face. In order to overcome this shortcoming, the feature points of non-tracking objects are identified from the overlapping objects by using the previously introduced feature point separation method and their values are further changed from skin color (255) to background (0). In this way, the skin color structure of the non-tracking object will be destroyed, then re-apply the DT operation and the maximum DT value will appear in the palm center position correctly.

However, a hand object usually contains wrist information, if the palm center experiences deformation, the position with the maximum DT value may appear on the wrist or the arm. As a result, the determined palm center is incorrect which will be inevitable to increase the tracking error. It is learnt from observation that images of wrist and arm are smoother than those of the palm; therefore, the number of feature points contained in the wrist and arm would be small, while the number of feature points contained in the palm center would be numerous because of finger rifts, knuckles, and palm creases. In light of this characteristic, better palm center position can be identified from DT images via feature point information. Figure 3 shows an example of wrist and arm exclusion in which A is the position which, having the maximum DT value of D_A within the hand area, is obtained through detection or tracking mode; and outside the circle centered at A with a radius of D_A, B is the position which has a maximum DT value of D_B. Then calculate the feature point numbers F_A and F_B inside the two circles centered at A and B with a radius of 1.5 times of D_A and D_B respectively. Finally, F_A and F_B are compared and the point corresponding to the larger one will be taken as the palm center position.

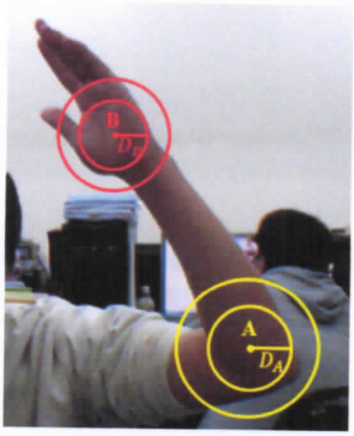


FIGURE 3. An example of feature point statistics for exclusion of arm interference

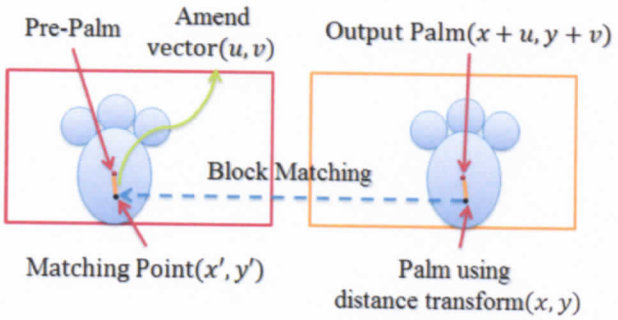


FIGURE 4. Palm center point compensation design

In order to improve the precision of the identified palm center position, we refer to the previous palm center position and carry out a palm center compensation operation. Let (x, y) be the identified palm center position through distance transform in I_t, (a, b) be the palm center position in I_{t-1}. First, block comparison is performed on an image block in I_t centered at (x, y) to find out the corresponding image block in I_{t-1} centered at (x', y'); then the deviation (u, v) between (x', y') and the previous palm center (a, b) is calculated, i.e., $u = a - x'$ and $v = b - y'$. If the deviation is low, it represents the currently identified palm center (x, y) is consistent to the previous palm center; but, if the deviation is high, it means the currently identified palm center point position (x, y) may have drifted from the correct palm center position considerably. In order to solve this problem, we perform position correction by adding the deviation (u, v) to the current palm center position (x, y); thus, the final output palm center point position becomes $(x + u, y + v)$ as shown in Figure 4.

When the hand object is not in overlapping state, we can according to the range of hand object, clearly define the rectangle area of the hand. However, when it is in overlapping state, the range of overlapping objects is usually greater than the range of hand object. Therefore, we proposed a hypothetic rectangle for tracked hand in overlapping state which can indicate the hand area in the overlapping object based on the palm center position and its DT value. Generally, a palm center has a larger distance to the tip of middle finger than to the bottom of the palm. Therefore, if the coordinate of palm center is (x, y) and its DT value is DT, then the hypothetic hand area HR is designed as

$$HR = (x - 2DT, y - 2.72DT, 4DT, 4DT) \tag{2}$$

4. **Experiment.** In the hand tracking experiment, a tracking stability comparison with the proposed tracking method, CamShift [8] and MIL-Boost [9] was carried out on 3788

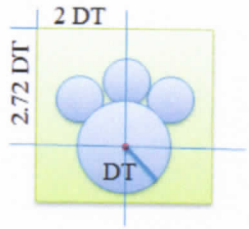


FIGURE 5. Hand rectangle range (DT is the distance transform value of palm center.)

FIGURE 6. Material images used in tracking experiment

TABLE 1. Experiment results of the proposed method

	Tracking success frames	Average tracking error	Average processing time	Frame lost
Proposed method	3668	4.542363	21.91411	7
CamShift	2496	25.88276	7.652735	490
MILBoost	3572	19.88961	98.33336	53

frames of 8 video clips recorded by different subjects under different environments and lighting sources. The palm centers of all frames were marked manually and Figure 6 shows some of them.

Before tracking, an Adaboost initiating gesture detector was used to initialize the tracking gesture position, and then three tracking methods were employed independently on the following video frames and the number of successful tracked frames (ST), the number of failed tracked frames (FT), average error distance and average processing time were recorded. ST refers to the number of frames in which gestures are successfully tracked. When tracking is successful, the distance between the detected palm center and its marked palm center is calculated and it is referred to as error distance. If an error distance exceeded 50 pixels, the tracking would be considered as tracking failure and FT would be increased by 1. Whenever a tracking failure happens, the initiating gesture detection is performed again to reset the tracking position for subsequent tracking. When all experiment images have been processed, the accumulated error distance and processing time would be divided by the number of tracking images to get the average error distance and average processing time. The experiment results are shown in Table 1. An observation on the mean error item reveals that the proposed method has a better accuracy than the other two methods. In addition, its average error is within 5 pixels. Furthermore, CamShift is vulnerable to background interference within the tracking range and at time of updating it is apt to accumulate errors, leading to large distance error and tracking

failure. As for MILBoost, in spite of its superior effectiveness to CamShift, it is still vulnerable to detection incompetence caused by large hand deformation, and this leads to tracking failure. With regard to average processing time, the proposed method is second only to CamShift but is still capable of providing real-time applications with average processing efficacy of 3045 images per second. Our method has far less tracking failures than the other methods, and its main error results from rather blurred images caused by speedy gesture movement.

5. **Conclusion.** In this paper, a fast and effective hand tracking technology which can correctly and steadily track the user's hand positions is proposed based on computer vision theory. A separation template image of the tracking hand is created in the state of proximity, and a feature-point-based block matching comparison is conducted in the state of overlap. By discriminating the feature points of the non-tracking object from the tracking object, the palm center of the tracked hand can be determined correctly. From experiments, it has been demonstrated that the proposed method can effectively reduce the deviation of palm center position, and the average processing time for each frame is just about 22 milliseconds. This makes the proposed method is concrete and feasible to various human-computer interactive applications.

Acknowledgment. This work is supported by the National Science Council of Taiwan with grant no. NCS101-2221-E-216-037-MY2-2. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] E. Lin, Hand tracking using spatial gesture modeling and visual feedback for a virtual DJ system, *Proc. of the 4th IEEE International Conference on Multimodal Interfaces*, pp.197-202, 2002.

[2] J. Han, Automatic skin segmentation and tracking in sign language recognition, *IET Computer Vision*, vol.3, no.1, pp.24-35, 2009.

[3] Y. Deng, B. S. Manjunath and H. Shin, Color image segmentation, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, pp.446-451, 1999.

[4] R. E. Kalman, A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, no.82, pp.35-45, 1960.

[5] Z. G. Pan, A real-time multi-cue hand tracking algorithm based on computer vision, *IEEE Virtual Reality Conference (VR)*, pp.219-222, 2010.

[6] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, *International Joint Conference on Artificial Intelligence*, pp.674-679, 1981.

[7] E. B. Sudderth, M. I. Mandel, W. T. Freeman and A. S. Willsky, Visual hand tracking using nonparametric belief propagation, *Proc. of the Conference on Computer Vision and Pattern Recognition Workshop*, pp.189-197, 2004.

[8] G. R. Bradski, Computer video face tracking for use in a perceptual user interface, *Intel Technology Journal*, vol.2, no.2, pp.1-15, 1998.

[9] B. Babenko, M. H. Yang and S. Belongie, Robust object tracking with online multiple instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.33, no.8, pp.1619-1632, 2011.

國科會補助計畫衍生研發成果推廣資料表

日期:2013/12/09

國科會補助計畫	計畫名稱: 快速又準確的立體影像深度估測法
	計畫主持人: 鄭芳炫
	計畫編號: 101-2221-E-216-034- 學門領域: 影像/視訊處理與電腦視覺

無研發成果推廣資料

101 年度專題研究計畫研究成果彙整表

計畫主持人：鄭芳炫	計畫編號：101-2221-E-216-034-
計畫名稱：快速又準確的立體影像深度估測法	

成果項目			量化			單位	備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等）
			實際已達成數（被接受或已發表）	預期總達成數(含實際已達成數)	本計畫實際貢獻百分比		
國內	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	1	0	100%		
		研討會論文	0	0	100%		
		專書	0	0	100%		
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（本國籍）	碩士生	3	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		
國外	論文著作	期刊論文	1	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	1	0	100%		
		專書	0	0	100%	章/本	
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（外國籍）	碩士生	0	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		

其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)	無

	成果項目	量化	名稱或內容性質簡述
科教處計畫加填項目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與（閱聽）人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

■達成目標

□未達成目標（請說明，以 100 字為限）

□實驗失敗

□因故實驗中斷

□其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文：■已發表 □未發表之文稿 □撰寫中 □無

專利：□已獲得 □申請中 ■無

技轉：□已技轉 □洽談中 ■無

其他：(以 100 字為限)

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）(以 500 字為限)

在 3D 立體顯示技術研究上，已完成從 2D 影像擷取 3D 深度資訊之技術及雙眼立體影像對之生成技術，為國內在 3D 立體顯示技術奠定了基礎。