

多視角取像與三維建模技術

陳文昭^{1,2}、陳稔¹

¹國立交通大學資訊工程學系

²工業技術研究院電子與光電研究所

E-mail: Chaody@itri.org.tw, zchen@cs.nctu.edu.tw

1. 簡介

在 2009 年由電影「阿凡達」帶起了一陣 3D 風潮，2010 年世界盃足球賽亦開始以 3D 方式進行轉播。現今的影像技術以及相關應用產業皆訴求 3D 化。從近年來立體影像技術相關的國際知名研討會(如 SIGGRAPH、CVPR)所發表的技術，可看出電腦圖學與電腦視覺兩大領域在技術上已開始進行整合，其目的即是希望提供觀看者更真實的立體影像。隨著近年來立體顯示器的技術日益成熟，互動式全視角 3D TV(interactive free-viewpoint 3D TV)將會是下一代家用電視的新規格，其播放的內容(content)將不只是目前的立體影像，而是具有高品質的 3D 模型。精緻的 3D 立體模型除了可滿足視覺上的享受之外，模型本身具有高度的互動性(interact)、且充滿可變性(flexible)，於各領域中皆可讓觀看者見到更接近人眼視覺的影像。收看此類可互動式全視角 3D TV 的觀眾，將可以自行決定欣賞電影、運動節目的觀看角度。

然而要能讓 3D 相關產業能有效的發展，除了需具備立體顯示器相關技術之外，最重要的還是能有效率的大量製作 3D 數位內容。因此本研究提出使用多視角取像系統(multi-view capturing system)來擷取物體不同視角的資訊，用以拍攝及製作含有 3D 模型的內容。本技術主要發展能即時(real-time)建構出物體三維模型的技術，以未來應用於即時轉播(live broadcasting)的 3DTV 為目標。利用八台攝影機，同步擷取動態場景的不同視角影像，對每張影像進行背景去除以得到輪廓線影像，再利用輪廓線影像進行影像式建模計算，最後經由多視角貼圖技術，即可建置出含有貼圖之動態 3D 模型。此外，本系統亦與擴增實境技術進行結合，除了可從各角度自由觀看物體外，亦能搭配各種 3D 影像的應用需求。

2. 相關研究

重建物體 3D 模型的演算法可以分為兩大類，一種是主動式(active)，另一種是被動式(passive)。傳統上，主動式的方式(如雷射掃描器)，是利用打主動光將圖案投影至物體身上，再經由光的反射，利用光圖案在影像中的變化或三角測量法則來計算出物體的三維幾何資訊。但是若物體的材質(如玻璃)無法有效反射圖案，主動式方法將會失敗。而另一種單純利用相機取像重建物體之三維模型的被動式視覺重建技術，除了可以避開此問題之外，亦節省了整個取像裝置的花費。

2.1. 多視角取像系統

目前多視角取像系統平台以國外發展較為成熟，其中 Stanford 大學架設了一套具有 100 台 320x240 的 HD multi-view camera 系統[1]，利用此系統除了可以得到 6900x3500 解析度的影像，亦可利用此系統拍攝高速影像，快門可內插至 1/100000000 秒。Chang 及 Chen [2]亦架設了一套 self-reconfigurable camera array 系統。利用 48 台可移動式相機平台，其取像解析頻率為 640x480x30fps，透過 100Mbps 的 Ethernet cables 連接至 HTTP 伺服器發送訊號與傳送 JPEG 影像資料，以約 4-10fps 的速度產生 640x480 的新視角影像，而其獨到之處是單單利用一台 PC 來控制所有的 cameras 的取像與運動。

另一方面，Microsoft Research 的 Lou 等人[3]發展一套 multi-view video 系統，包括即時取像、運算、及互動播放等功能。在實作上，他們提供了相關解決方法，包括系統校正、物件追蹤、影像壓縮與互動傳送。可以在 real-time 的環境下，顧客可以互動地選擇他們想要觀看的方向，來享受許多刺激的視覺體驗，諸如 view switching、freezing shot 以及 view sweeping。這套系統可同時應用於請求服務傳送(on-demand delivery)以及現場廣播(live broadcast)。該系統共包括 32 台 camera，各嵌在 32 個具有 pan/tilt 的支架，並連截至 16 個同步裝置(sync units)與 16 部個人電腦(control PCs)。此互動多視角視訊(Interactive Multi-View Video)架構包含了三個部分：(1)取像端：包含了 CCD camera，鏡頭，同步裝置，pan/tilt 支架，以及控制 PC。值得一提的是，一旦開始捕捉動態物體，video signals 就會在 control PCs 進行壓縮，再藉由 network backbone 傳送至 server。(2)伺服器端：Server 負責收集大量從控制 PC 傳送來的已壓縮 video streams，並對用戶端提供多視角視訊服務。此外，Server 亦扮演 camera 管理者的角色，包括產生 camera 的控制訊號，進而改變相機取像方向、相機取像參數等設定。(3)用戶端：用戶的部分是負責接

收由 server 傳來的多視角視訊資料，然後解碼與播放顯示 video 給 end-user 觀賞。藉由控制 client，顧客可以享受現場互動式多視角視訊。此系統能提供的視覺體驗包括「view switching：可任意切換觀看視角」、「frozen moment：使用者可以在同一個時間點來回瀏覽各不同角度拍攝的 camera 取像資料」、「view sweeping：當拍攝事件持續進行，使用者可以持續在連續的 view direction 觀賞」。

2.2. 三維建模技術

三維建模相關技術，可細分為三大類：(1) Volume-based approaches, (2) Surface-based approaches, (3) Depth map based approaches。

(1) Volume-based approaches

1974 年 Baumgart[4]提出了 volume intersection 的方法，此方法的輸入影像是 binary 的不同角度之 silhouette 影像，先用一個足夠大的正立方體包覆住要重建的物體，接著利用這些 silhouette 影像來“雕刻”此正立方體。因此也被稱為 shape-from-silhouette，此方法有以下的特點：(a)這種方法只能重建出 convex 物體，無法重建物體凹陷的部份。(b)在實作時，通常是利用 voxel 的方式來表示空間的最小單位，並依需要而設定 voxel 的大小，voxel 尺寸愈小，就能重建出愈精細的物體，不過相對需要大量的儲存空間，如下圖所示。或是利用大 voxel 來表示較粗大的區域，以及用小 voxel 來表示較精細的部份。(c) Voxel 常會搭配 octree [5-7] 來做為其表示方式。(d)重建物體的品質取決於輸入影像的數目。

Seitz[8]為了突破 binary 的 voxel 分類方式，提出了 voxel coloring 的演算法，但位於前方的 voxel 在某些角度的影像中會擋到後方的 voxel，這也就是所謂的 visibility 問題。對於此問題，voxel coloring 採用 depth ordering 的方法，從離相機較近的位置往遠方檢查，也就是先計算遮蔽物的部份，如此可以用 one-pass plane sweep 演算法在一回合內計算完畢。然而此方法只適用於相機在物體的另一邊，或是其他可以依特定順序計算的相機擺放方式。為了解決 voxel coloring 對於相機位置的限制，Kutulakos[9,10]提出了一個較 general 的 space carving 演算法。Space carving 演算法是從 volume 的六個面做 sweeping，且只考慮與 sweep plane 同一邊的相機，此方法稱為 multi-pass plane sweep 演算法。此方法易於實作，且因為 orders of magnitude 較低，所以執行速度較快，對於具較細結構或是具有不連續表面的物體能有不錯的重建結果。

(2) Surface-based approaches

Plane sweeping 演算法常被使用於多視圖影像重建技術當中，其演算法是針對某一特定的影像，利用空間中相互平行的平面在 3D 空間中反覆的移動，位於空間中某深度的平面可以與其他影像之間形成 homography 的關係，如此即能將 key view 影像輕易地 mapping 到其他 reference 影像上。如果在某一深度的平面剛好通過欲重建物體的表面，key view 影像上的顏色值應該與 reference 影像所對應位置的顏色值一樣，反覆利用此原理進行比對，即能重建出物體的 3D 模型。

前述的 plane sweeping algorithm 是利用許多空間中的平面來逼近真實物體的表面，而這些平面皆是 fronto-parallel 於某參考影像。然而此種方法並無法呈現出物體真正的表面情形，Zabulis[11]則對此提出了一個具有 multiple sweeping direction 的 plane sweeping 方法，除了平面的深度之外，亦考慮到平面的方向，但是這將會導致必須要對無限種可能的平面方向做比對、檢查。對於此問題，Gallup[12]則提出了一個有效的方法來減少比對的次數，除了可以利用對於已知場景的 prior knowledge 來做 heuristic 假設外，亦可以透過檢查場景的大致結構以減少需要檢驗的平面方向。他提出含有兩項 penalty term 的 energy function，並透過 graph cuts 演算法求得最佳解。

然而在物體表面缺少 texture 的情況下，純用平面上的 texture 來做 correlation 會有問題，就算有許多影像也不一定能有更好的結果，但若加入 visual hull 的資訊，將有助於找出物體約略的空間位置。因此 Furukawa 與 Ponce [13]使用各個角度的 silhouette 影像，先計算出物體大致的形狀及空間位置，利用此資訊可以先建置出 visual hull，以得到物體較粗略的表面資訊。當獲得大部份表面資訊之後，再利用 graph cuts 針對物體表面的 photo-consistency 來做全域最佳化，以找出表面的主要特徵(main feature)，接著使用疊代的方式來針對表面細節的部份做更進一步的修正。Furukawa[14]則提出了另一種方式來重建稠密式的 3D 物體模型，此方法不需事先知道物體的 bounding volume，其主要步驟為：(1) Matching –利用 Harris 或是 Difference-of-Gaussians 來擷取影像上的特徵點，然後在各個角度的影像中計算特徵點對應關係，再以此做為 sparse 的對應區塊。接著再一直重覆下列兩步驟，直至建立起完整的 3D 模型。(2) Expansion –從前一步驟所計算出來的對應區塊開始由鄰近的 pixel 往外擴張，以得到較 dense 的對應區塊。(3) Filtering –利用 visibility 的限制來去除不正確的對應區塊。

對於此類技術，我們也進行相關的技術發展，利用圖 1(a)的多視角取像系統，拍攝如圖 1(b)的八張人臉影像，利用 surface-based approach 進行 photo consistency 的計算，可以得到圖 2(a)-(c)中人臉的三維重建結果，然而此方法需要大量的運算時間進行每個小區塊的比對。

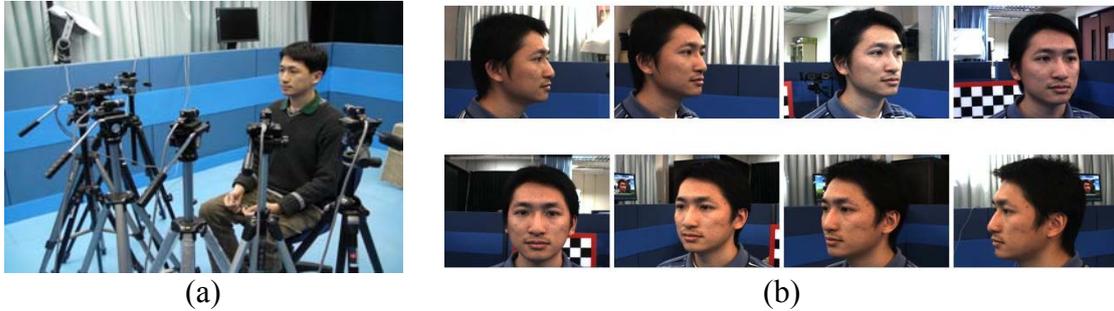


圖 1. (a)利用多視角擷取系統拍攝人臉、(b)不同視角的拍攝影像

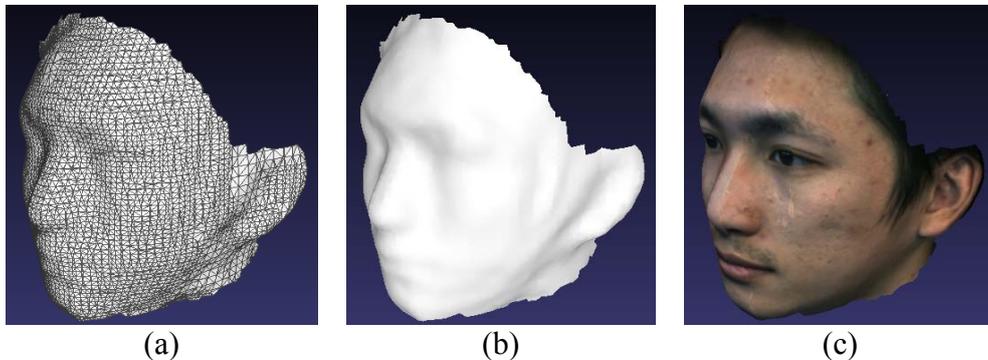


圖 2. (a)重建出的人臉三維網格模型、(b) 人臉三維模型表面、
(c)貼上材質後的三維模型

(3) Depth map based approaches

Hernandez and Schmitt [16]先計算每張影像的 depth map，再將其結果進行合併成一個 cost volume。之後再從初始的 visual hull 開始對 mesh 進行疊代變形，以此找到最接近物體的表面，因此可以定義一個 optimal surface 來對 global energy 做最佳化。Goesele[17]也是將不同角度所重建的 depth map 合併成一完整的模型，其演算法分為兩步驟：(1) Depth map generation – 對每一角度的影像計算其 depth map，對於一深度值，計算一個 correlation 值當做所有影像的 NCC 平均值。(2) Merging depth maps – 利用 VRIP[18]來合併所有的影像。

3. 多視角取像系統

為了滿足全視角 3DTV 相關應用的需求，我們提出了一套多視角取像系統，其整體架構及流程如圖 1 所示。在此系統中的前處理模組(preprocessing module)

包含了相機校正、背景估測，以及不同相機間的色彩校正。當開始拍攝一段同步的多視角影像後，此系統即進行背景去除的動作，並基於 octree 結構進行三維模型建置，最後以多視角貼圖技術來真實呈現重建的三維物體。在此架構中，只有 octree 結構及多視角影像會從伺服器端傳送至客戶端。本研究主要改善傳統利用 octree 結構重建三維物體的技術，並結合 exact marching cube (ExMC) [19]來進行重建物體表面的描述，以此方法達到即時(real-time)建模的結果。

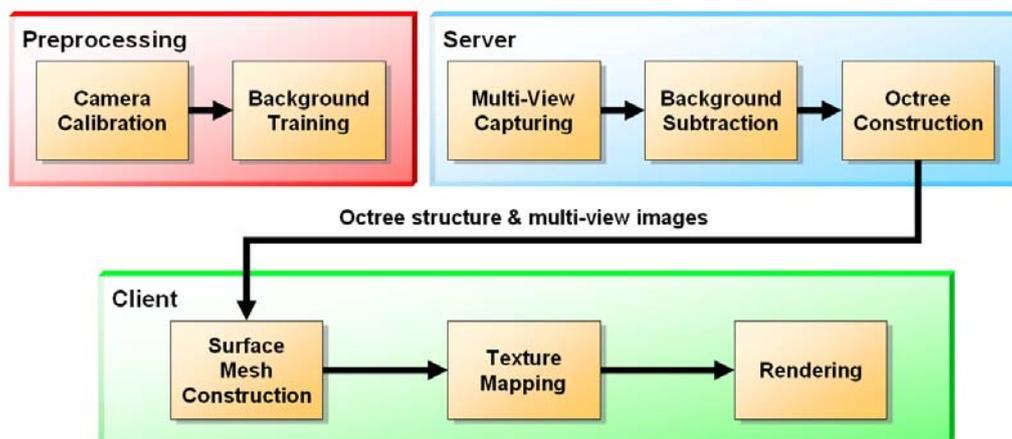


圖 1. 系統流程圖

3.1. 取像系統硬體架構

本系統主要訴求在於即時(30 fps)捕捉擷取高解析度(1024 x 768)的未壓縮影像(16 bit)，其資料量相當龐大，於 45MB/sec，對系統效能的要求甚高。整套系統架構如圖 2 所示。每一部 Camera 連接到一台 PC，各 PC 以 1394a 訊號線相串接，並由網路連接形成一獨立區域網路。我們架設了一個類似藍幕的小型攝影棚，其可用的拍攝空間約 3 m x 6 m x 2 m，共有八台取像鏡頭架設在此攝影棚四周，多視角取像的結果如圖 3 所示。

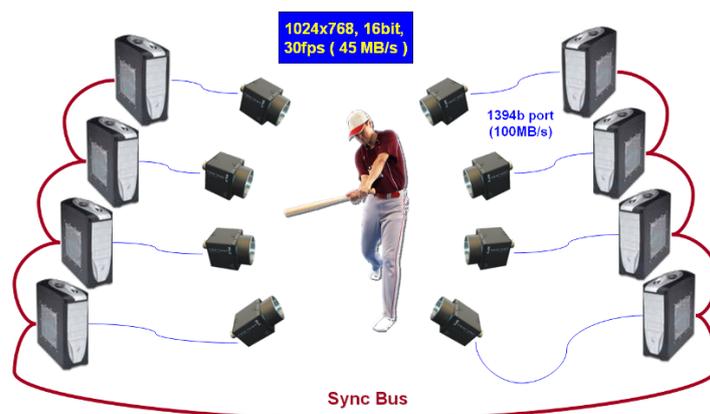


圖 2. 多視角同步取像系統



圖 3. 八個不同角度的拍攝影像

本系統利用色彩校正板來對八台相機進行顏色上的校正，此外亦利用 Bouquet 的方法[20]，以棋盤格校正板來進行相機投影矩陣的估算。雖然物體在一個藍幕的環境下進行拍攝，但仍會因陰影的關係無法完整擷取出前景物體，故本系統採用[21-22]所提及的方法，並以人工的方式選定一個 threshold 來正確擷取出前景物體的輪廓線影像(silhouette image)，並用此輪廓線影像進行後續的三維建模。

3.2. 多視角三維建模技術

本研究以 octree 結構利用物體輪廓線影像重建三維模型，在維持重建三維模型精確度的前提下，可以在較早的 octant 切割階段，即能預測並判斷每個 octant 的屬性，以減少需儲存的 octant 數量，進而減少記憶體消耗以及加快處理的速度。傳統利用 octree 來重建物體時，需要先指定初始(root) octant 的體積大小，如此可以計算出 octant 在三度空間的位置。針對初始 octant 做一次切割，會得到八個 sub-octant，如圖 4 所示。此時即能將 sub-octant 經由投影矩陣投影到不同角度的輪廓線影像。投影後便能比較各 sub-octant 的 2D 投影和物體之輪廓線影像是否有交集(intersection)，並進行 sub-octant 的分類。在依交集的結果分類時，若一個 octant 完全處於物體內部，則稱其為「黑色」節點，反之，若一個 octant 完全處於物體外部，則稱其為「白色」節點，而「灰色」節點則是僅有部份處在物體的外部。其架構如圖 4 所示，如此地分割及分類直至沒有任何「灰色」節點為止。

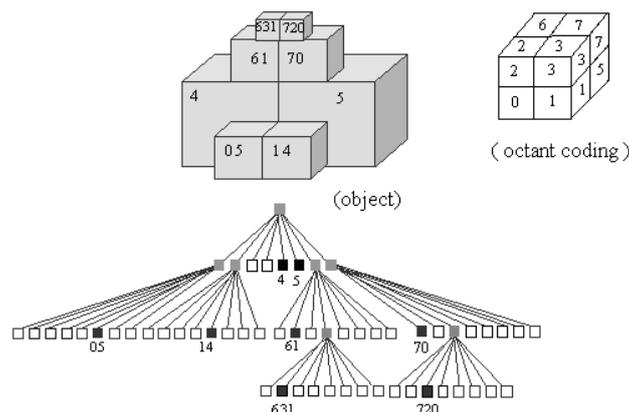


圖 4. 以 octree 重建物體

在傳統利用 octree 重建技術中，octant 以前述方法進行分類時，在實作上通常會以一個外接圓來表示一個 octant，並利用含有正負號的距離地圖(distance maps)[23]來加速判斷交集的結果，如圖 5 所示。此距離地圖記錄了影像中每個像素到物體成像輪廓線之最短距離。並以正負來區分此像素是落在物體的成像內或外面。而計算 Chebyshev 距離地圖時，可以利用 Hoff 等人[24]所提的方法，結合顯示卡本身功能來達到高速運算。先將 octant 的八個頂點投影在 2D 影像中，利用此八個頂點之 2D 投影點計算出一個圓心為 c ，半徑為 r 之外接圓包住此 octant 投影。而圓心 c 的距離值(distance value)，在此稱為 $DistMap(c)$ 。利用外接圓的半徑 r 與 $DistMap(c)$ 之間的關係可得知做交集的結果，若 $DistMap(c)$ 值為正，則表示圓心在物體輪廓線的內部，反正則是在物體外部。當 $DistMap(c)$ 值為零時，則表示正好在物體的邊緣上，對每個視角影像都做此空間關係的檢查後，則能得到如表 1 所列出的各種關係結果。

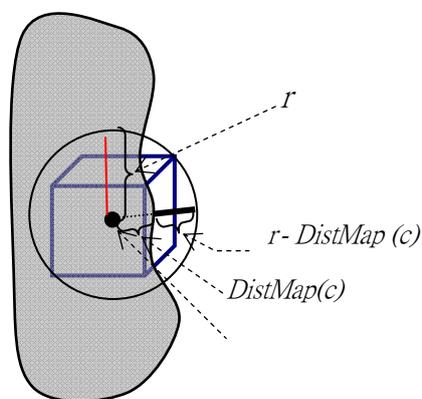


圖 5. 利用距離地圖來描述 octant 與輪廓線影像之間的空間關係

表 1. 判定 octant 投影與各角度輪廓線影像交集之關係

$DistMap(c) \geq 0$	在任一影像中 $r > DistMap(c)$	Octant 投影與輪廓線相交 (令其為 grey octant)
	在所有的影像中 $r \leq DistMap(c)$	Octant 投影在輪廓線之內 (令其為 black octant)
$DistMap(c) < 0$	在任一影像中 $r > -DistMap(c)$	Octant 投影與輪廓線相交 (令其為 grey octant)
	在任一影像中 $r < -DistMap(c)$	Octant 投影在輪廓線之外 (令其為 white octant)

為了能夠量測重建出的物體 O 與物體 T 本身之間的差異，我們使用輪廓線影像(需轉換成 binary 影像) S_v^O 與 S_v^T ，其中 $v=1, \dots, M$ 代表不同的視角。依下式進行平均的 XOR 運算來計算誤差：

$$XOR\ error = \frac{1}{M} \sum_{v=1}^M S_v^O \otimes S_v^T \quad (1)$$

上式中的 $S_v^O \otimes S_v^T$ 代表對所有視角影像，兩張 binary 輪廓線影像全部的 XOR 誤差數量(單位為像素)。

另一種量測重建物體準確度的方法是利用下式計算誤差像素佔的百分比：

$$Err(O, T) = \sum_{v=1}^M S_v^O \otimes S_v^T / \sum_{v=1}^M Area(S_v^T) \quad (2)$$

其中 $Area(S_v^T)$ 是真實物體輪廓線內的區域面積(或像素總數)，換句話說，若重建結果愈好，其值會愈小於 1。

本研究提出一種新的 octree 重建方法，需事先定義一容許 XOR 誤差上限值 P 。在每一次切割時，位於階層(level) l 時，會會分類成五種 octant： $B_{l,N}$ (黑)、 $W_{l,N}$ (白)、 $GB_{l,N}$ (灰黑)、 $GW_{l,N}$ (灰白)以及 $GG_{l,N}$ (灰灰)。其中黑與白 octant 與前述傳統的定義一致，而其他三種灰色 octant 則定義如下：

傳統方法中在 l 階層下的灰色 octant $G_{l,c}$ ，如果其外接圓圓心 c 的距離值 $DistMap(c)$ 小於 0，且此 octant 的在輪廓線以內的(黑色)部份在任何視角下都滿足 $0 < r + DistMap(c) \leq P$ ，則將其定義成灰白($GW_{l,N}$) octant。

傳統方法中在 l 階層下的灰色 octant $G_{l,C}$ ，如果其外接圓圓心 c 的距離值 $DistMap(c)$ 非負，且此 octant 的在輪廓線以外的(白色)部份在任何視角下都滿足 $0 < r - DistMap(c) \leq P$ ，則將其定義成灰黑($GB_{l,N}$) octant。

傳統方法中在 l 階層下的灰色 octant $G_{l,C}$ 不能被上述兩種方法所分類，則將其定義成灰灰($GG_{l,N}$) octant。也就是說此 octant 在輪廓線以外(白色)及以內(黑色)的部份皆未達到原先設定的誤差上限 P 。

灰白 octant 意即有 octant 本身有大部份位在物體輪廓線之外，反正灰黑 octant 則是 octant 本身大部份位在物體輪廓線之內。我們僅針對灰灰($GG_{l,N}$) octant 進行細分割，直至沒有新的 $GG_{l,N}$ 產生。在這裡亦定義了一個值 $L_{N,P}$ ，代表在誤差上限值 P 時，不再有灰灰($GG_{l,N}$) octant 產生時的階層(level)數。

3.3. 動態建構與漸進式傳輸物體模型

當利用前述的 octree 結構方法，以及誤差上限值 P 重建出物體模型後，若想要得到一個以更小 P 值的模型，可以動態的更新產生而不需完全重頭進行重建。也就是說，對於一個新的較小的 P 值，只有那些有著比 P 值還大的灰黑及灰白 octant 需要進行細分割，依此方法，octree 模型即能動態的進行更新且變得更加細緻，而此建構及 octant 分割的過程可以利用雙向的指標來進行記錄。

在建置 octree 時，每一個產生 octant 的 XOR 投影誤差都會被計算，此投影誤差值不止能用於上述的動態建構之外，更能用於漸進式的傳輸。當要傳輸一個重建出的物體模型，可利用一個「優先序列」(priority queue)來儲存依 XOR 投影誤差由大至小排過序的 octant 列表。每一次在進行切割時，皆針對優先序列中第一個 octant 做細分割的動作，因為該 octant 有著最大的投影誤差，並且將其標示為是在誤差上限值等於 P 時所切割的結果。而細切割出的新的八個子 octant，也依照各自的 XOR 投影誤差值的大小插入優先序列當中，此漸進式的重建及 octant 分割的過程亦可以利用雙向的指標來進行記錄。在傳輸模型時，僅有灰黑及灰白的 octant 需要由伺服器傳輸至用戶端進行描繪(rendering)。當優先序列中第一個 octant 的誤差值不大於原先設定的誤差上限值 P 時，即終止漸進式傳輸。

3.4. 物體表面網格表示方法與材質貼圖

在進行漸進式傳輸的同時，除了重建出的octree模型之外，亦將會由伺服器端傳輸一至多張多視角影像用戶端。接著即可利用exact marching cubes (ExMC)演算法來得到octant投影的各個邊，與輪廓線影像之間「確切相交點」(exact intersection)。Octant投影的邊可能會與輪廓線影像有多個相交點，此時則選擇其中一個在各個視角中離輪廓線最近的一個點做為確切相交點。如此即能將octant結構的體積式表示方法，轉換成以網格型式表示的物體表面。

在轉換成網格時，需要注意是否會因此在網格上產生破洞(crack)。傳統octree重建方法則是所有在最後一階層的灰色octant都是同樣階層產生的，因此octant的大小皆相同，如此在轉換成網格時，並不會有破洞產生。本研究所提出的方法，因為相鄰的octant可能是不同階層所產生的，因此octant的大小也不同，在轉換成網格時，會因此而產生破洞。為了要避免破洞的產生，我們目前採用的方法是將在較低階層產生的較大octant切割成最後階層octant的大小，如此在使用marching cubes轉換成網格時即不會產生破洞。另一方面，Kazhdan等人[25]提出了Poisson surface reconstruction演算法(PSR)，可以針對具有方向性的3D點進行平滑表面的估測，然而此方法非常耗時，並無法應用於3DTV即時播放的應用當中。

對一個僅有一個使用者觀看的用戶端而言，其實僅需依使用者所想要觀看的角度描繪出單一視角結果即可。因此依此需求的視角角度，選擇一至兩張最接近此視角的原始拍攝影像即足以進行材質貼圖的計算。當給定一個視角時，利用其投影矩陣即可計算出UV貼圖座標，利用目前市面上的顯示卡，即能在30毫秒內完成材質貼圖的動作。

4. 實驗結果

4.1. 以公用多視角影像資料為基準進行演算法比較

在此章節，我們從網路上收集較為測試比較使用的資料集，包含下列影像序列資料：David(19張影像)、dinosaur(34張影像)、以及dancer(20張影像)[27]。我們比較三種不同的物體重建方法，包含：(1)傳統octree重建方法結合marching cube演算法[26]轉換成網格表示方法(以下簡稱Conv + MC)、(2)傳統octree重建方法結合Exact marching cube演算法[19]轉換成網格表示方法(以下簡稱Conv + ExMC)、以及(3)本研究所提出的octree重建方法結合ExMC演算法。表2-4中列出這三種不

同方法，在重建物體的XOR誤差值、三角網格數量、以及執行時間的比較結果。從這些表格中可以看到本研究所提出的方法有最佳的表現，Conv + ExMC其次，而傳統的Conv + MC最後，尤其是當物體愈算愈細緻時，此差異愈明顯。

從圖6-10中可以看出這三種不同方法所重建的三維模型之間的差異。在圖7(c)與圖8(c)中與其他方法在局部有較不同的地方。如表2中所示，本研究所重建的dinosaur結果，較其他兩演算法有較小的XOR誤差值結果，因此我們的方法可以得到較佳的重建品質。一般來說，本研究所提出的方法允許灰黑及灰白的octant的存在，因此會有部份的重建結果較物體突出或凹入。而另兩種方法基本上並不會有任何凹入輪廓線的現象發生。此外，我們也可以透過控制 P 值來產生較粗或較細的重建結果。

表 2. 三種不同演算法在約略同樣網格數下的重建 XOR 誤差比較表

	David (19 張影像)			Dinosaur (36 張影像)			Dancer (20 張影像)		
	網格 數目	XOR error	$Err(O,T)$	網格 數目	XOR error	$Err(O,T)$	網格 數目	XOR error	$Err(O,T)$
Conv+MC	27,368	21,561	6.525 %	62,616	147,671	7.139 %	10,198	90,665	6.632%
Conv+ExMC	27,368	21,244	6.481 %	62,616	120,586	5.830 %	10,198	75,594	5.530%
本研究	26,602	15,786	4.888 %	69,242	87,891	4.249 %	10,897	73,897	5.406%

表 3. 三種不同演算法在約略準確度下的三角網格數目比較表

	David			Dinosaur			Dancer		
	$Err(O,T)$	XOR error	網格 數目	$Err(O,T)$	XOR error	網格 數目	$Err(O,T)$	XOR error	網格 數目
Conv+MC	8.007 %	26,787	6,658	7.139 %	147,671	62,616	5.572%	76,178	42,200
Conv+ExMC	7.367 %	24,100	6,658	6.677 %	138,103	15,192	5.530%	75,594	10,198
本研究	7.180 %	23,189	6,350	6.089 %	125,946	16,734	5.406%	73,897	10,897

表 4. 三種不同演算法在約略同樣網格數下的計算速度比較表

	David	Dinosaur	Dancer
網格數目	27k	62k	40k
Conv+MC	1,360	4,078	1,765
Conv+ExMC	1,453	4,204	1,728
本研究	750	1,390	624

* 計算平台為：Intel Quad Core 2.83GHz 處理器、具有3GB Ram

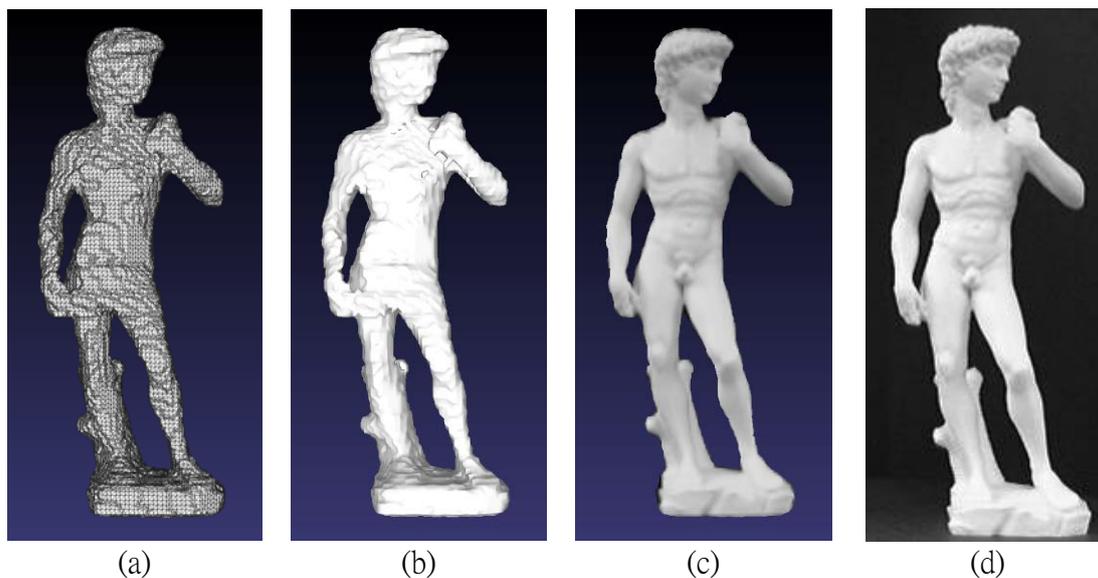


圖 6. 利用 David 序列影像所重建的模型結果，(a)以網格表示、(b)3D 呈像結果、(c)貼質貼圖結果、(d)原始序列的其中一張影像。

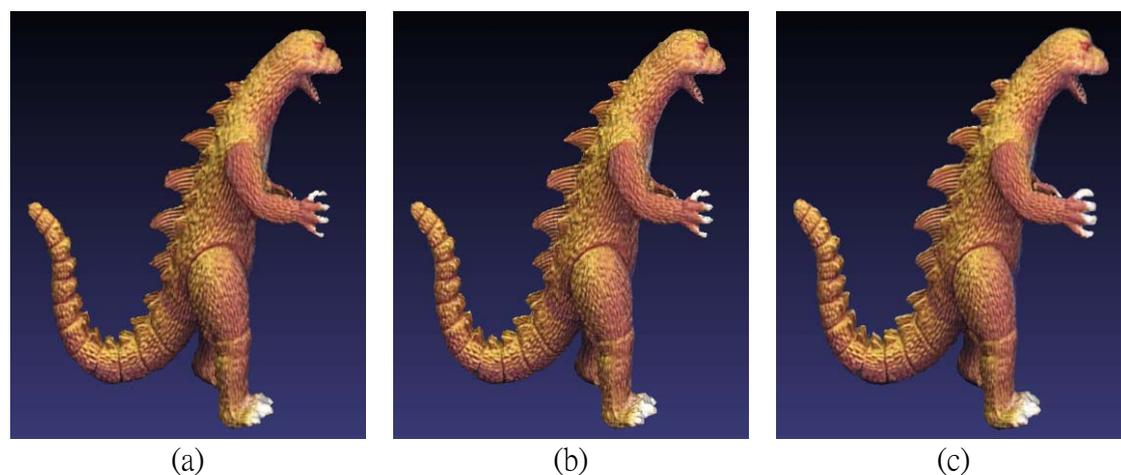


圖 7. 利用 dinosaur 影像序列，不同方法在約略相同三角網格數量下所重建出三維模型(含材質貼圖)結果：(a) Conv + MC、(b) Conv + ExMC、(c)本研究結果。

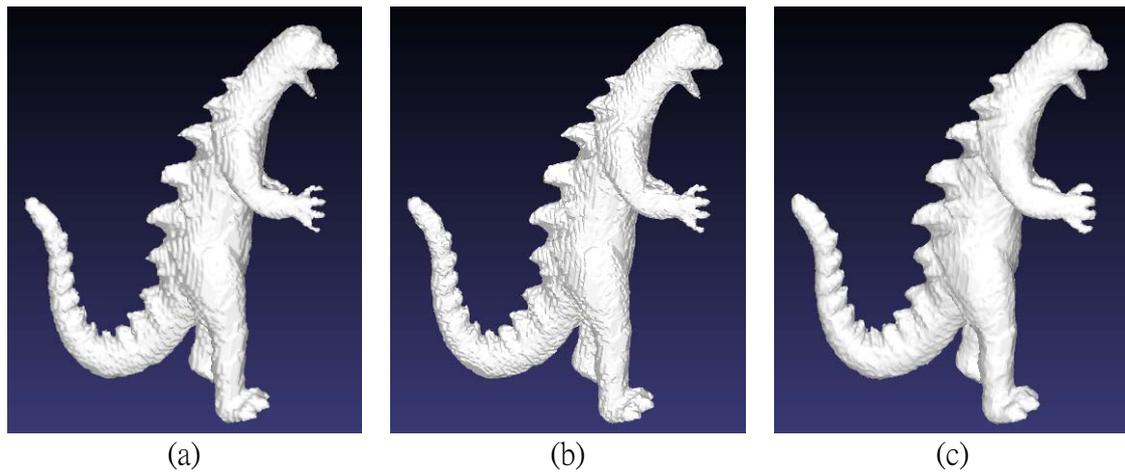


圖 8. 利用 dinosaur 影像序列，不同方法在約略相同三角網格數量下所重建出三維模型（不含材質貼圖）結果：(a) Conv + MC、(b) Conv + ExMC、(c)本研究結果。

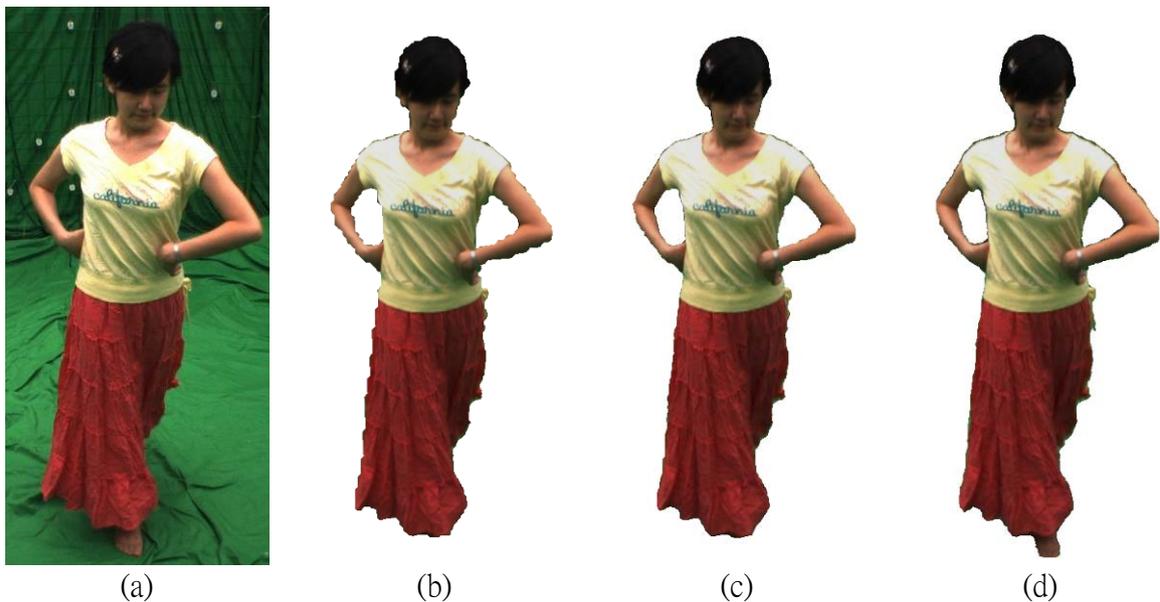


圖 9. 利用 dancer 影像序列，不同方法在約略相同三角網格數量下所重建出三維模型(含材質貼圖)結果：(a)原始序列的其中一張影像、(b) Conv + MC、(c) Conv + ExMC、(d)本研究結果。

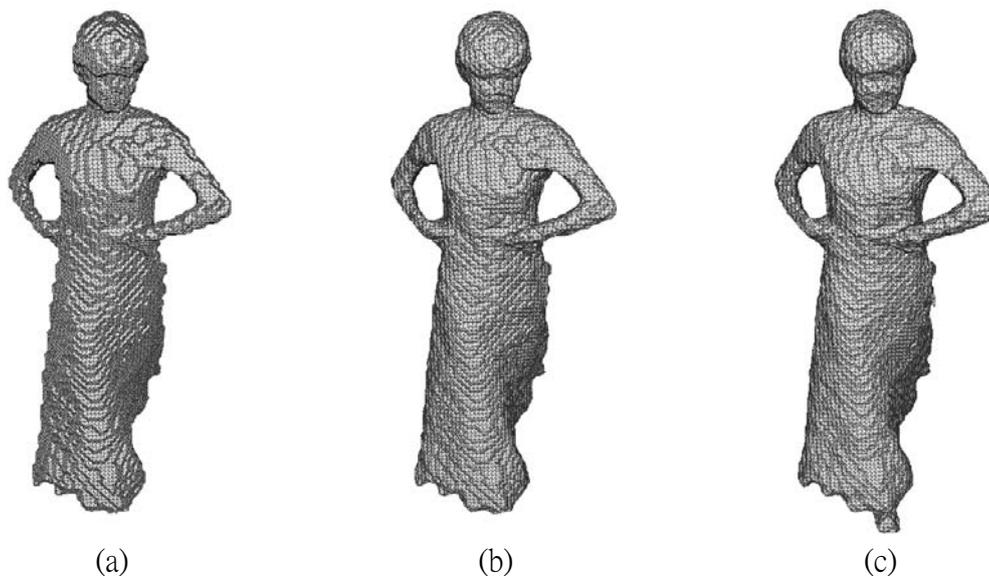


圖 10. 利用 dancer 影像序列，不同方法在約略相同三角網格數量下所重建出三維模型 (不含材質貼圖) 結果：(a) Conv + MC、(b) Conv + ExMC、(c)本研究結果。

4.2. 漸進式人物三維模型重建結果

針對本研究所提出的漸進式傳輸的三維模型建構方法，即利用於第三章中提及的多視角取像系統所拍攝的多視角影像進行實驗。圖11(a)-(c)為使用3.3節所提及的漸進式重建octree結構的結果。圖12為經ExMC轉換成網格表示方法後的三角網格模型；圖13則是貼上材質後的結果以及相關的效能資訊。

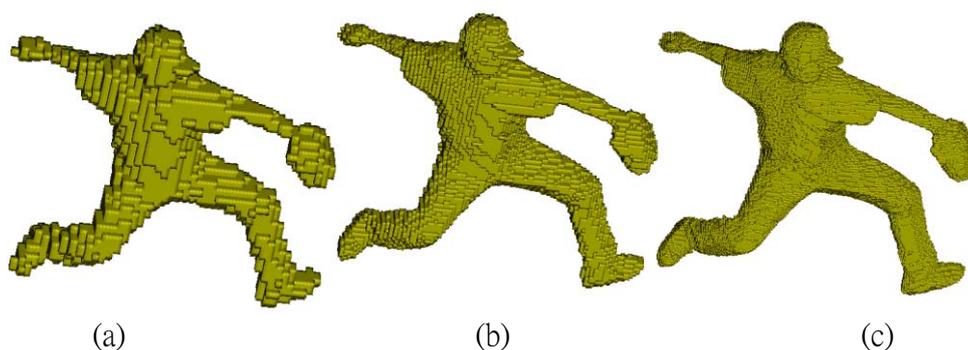


圖 11. 分別在不同投影誤差上限值的漸進式重建 octree 結構模型結果：(a) $P=13$ 、(b) $P=7$ 、(c) $P=4$ 。

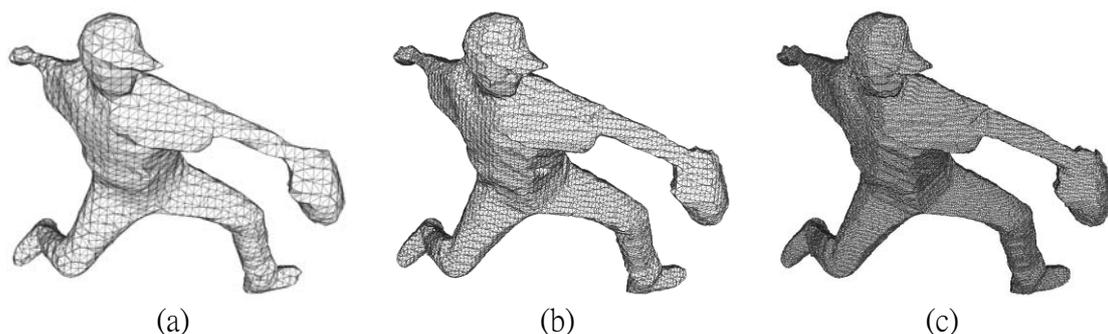


圖 12. 對圖 11(a)-(c)的重建結果進行 ExMC 演算法轉換成網格的結果：

(a) $P=13$ 、(b) $P=7$ 、(c) $P=4$ 。



(a)



(b)

Octree Reconstruction time	47
Exact Marching Cubes	437
Texture mapping	< 1
Total time (ms)	485

Total number of vertices	3862
Total number of triangles	7720

*Intel Quad Core 2.83GHz, 3GB Ram

(c)

圖 13. (a)多視角系統拍攝的原始影像之一、(b)對圖 12(a)的三維模型進行材質貼圖計算後的結果、(c)系統處理速度以及三維模型相關資料。

4.3. 擴增實境應用

本系統利用如ARTookKit[28]的擴增實境(augmented reality, AR)軟體，將重建的動態模型以擴增實境的方式呈現。此互動AR的應用，主要偵測攝影機拍攝的視訊中是否有出現已知的二維條碼圖型。利用此偵測出的圖型，可將重建出的動態人物3D模型放置於正確的空間位置。如此就能如圖14中的使用者以手旋轉此二維條碼進行不同角度的觀看。



圖 14. 利用擴增實境的方式進行不同視角的觀看。

圖15為我們所設計的不同動作結果，每個不同的二維圖型各代表其中一種動作，如投球或是揮棒的動作。所有動作皆以本研究所提的多視角影像擷取系統拍攝，並可如圖16般的在切換不同時間點的動作，以動態三維模型的方式呈現。

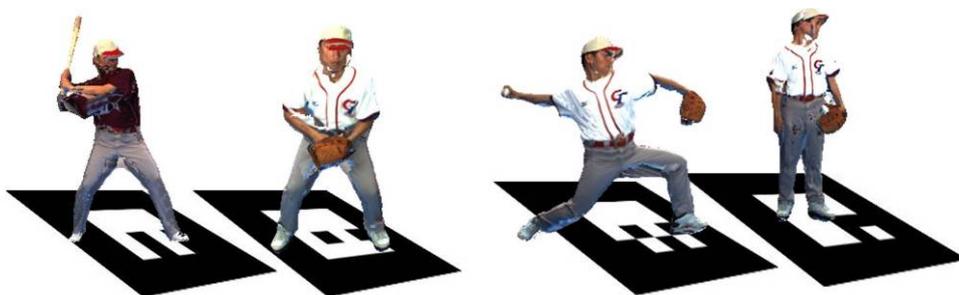


圖 15. 不同的二維圖型各自對應於一種動態三維模型。



圖 16. 在不同時間點會呈現不同的動作三維模型。

5. 結論

本研究提出了一個多視角影像擷取與三維建模系統，其中利用了新型的octree結構以多視角影像來重建物體三維模型。改善傳統octree重建技術，同時考慮octant

與物體輪廓的內外部之間關係來進行加速分類。重建出的三維模型除了可以真實還原物體的外型及外觀之外，亦能應用於漸進式的傳輸，使用者可以快速得到初期重建結果，且此三維模型會漸進式的變精緻。最重要的，由實驗結果可以看出，利用少數幾台PC或cluster進行平行化運算後，本系統即可達到即時計算的結果，並可應用於互動式3DTV上。未來本研究將繼續朝向重建出密集式的三維模型，以改善本研究無法重建內凹型的物體。

參考文獻

- [1] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Levoy, Mark H., "High Performance Imaging Using Large Camera Arrays", in *Proc. SIGGRAPH*, 2005.
- [2] C. Zhang and T. Chen, "A Self-Reconfigurable Camera Array", in *Eurographics Symposium on Rendering*, 2004.
- [3] J.-G. Lou, H. Cai, and J. Li, "A Real-Time Interactive Multi-View Video System", in *ACM Multimedia*, 2005.
- [4] B. G. Baumgart, "Geometric modeling for computer vision," *Technical Report Artificial Intelligence Laboratory Memo AIM-249*, Stanford University, 1974.
- [5] M. Potmesil, "Generating octree models of 3D objects from their silhouettes in a sequence of images," *Computer Vision, Graphics and Image Processing*, 40:1–20, 1987.
- [6] R. Szeliski, "Rapid octree construction from image sequences," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 58, no.1, 1993, pp.23– 32.
- [7] N. Ahuja and J. Veenstra, "Generating octrees from object silhouettes in orthographic views," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no.2, 1989, pp.137–149.
- [8] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Int. J. Computer Vision*, vol. 35, no.2, 1999, pp.151-173.
- [9] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *Int. J. Computer Vision*, vol. 38, no.3, 2000, pp.199-218.
- [10] K. Kutulakos, "Approximate N-view stereo," in *Proc. European Conf. on Computer Vision*, 2000, vol.1, pp.67-83.
- [11] X. Zabulis and K. Daniilidis, "Multi-camera reconstruction based on surface

- normal estimation and best viewpoint selection,” in *Int. Symp. 3D Data Processing, Visualization and Transmission*, 2004, pp.733-740.
- [12] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, “Real-time plane sweeping stereo with multiple sweeping directions,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.
- [13] Y. Furukawa and J. Ponce, “High-fidelity image-based modeling,” *Technical Report 2006-02*, UIUC, 2006
- [14] Y Furukawa, and J. Ponce, “Accurate, Dense, and Robust Multi-view Stereopsis,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.
- [15] M. Habbecke and L. Kobbelt, “A surface-growing approach to multi-view stereo reconstruction,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.
- [16] C.Hernandez and F. Schmitt, “Silhouette and stereo fusion for 3D object modeling,” *Computer Vision and Image Understanding*, vol.96, no.3, 2004, pp.367-392.
- [17] M. Goesele, B. Curless, and S. Seitz, “Multi-view stereo revisited,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, vol. 2, pp.2402-2409.
- [18] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proc. ACM SIGGRAPH*, 1996, pp. 303-312.
- [19] C. Liang, K. -Y. K. Wong, “Exact visual hull from marching cubes”, in *Proc. International Conference on Computer Vision Theory and Applications*, 2008, pp.597-604.
- [20] J. Y. Bouguet, Camera calibration toolbox for Matlab.
http://www.vision.caltech.edu/bouguet/calib_doc/
- [21] R. Zhang, S. Zhang, S. Yu, “Moving objects detection method based on brightness distortion and chromaticity distortion”, *IEEE Transactions on Consumer Electronics*, 53 (3) (2007) 1177-1185.
- [22] T. Horprasert, D. Harwood, L. S. Davis, “A statistical approach for real-time robust background subtraction and shadow detection”, in *Proc. IEEE International Conference on Computer Vision*, Greece, 1999, pp.1-19.
- [23] R. Szeliski, “Rapid octree reconstruction from image sequences, Computer Vision”, *Graphics and Image Processing: Image Understanding*, 58 (1) (1993) 23-32.
- [24] K. E. Hoff, J. Keyser, M. Lin, D. Manocha, and T. Culver, “Fast computation of

- generalized Voronoi diagrams using graphics hardware”, in *Proc. International Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, 1999, pp. 277-286.
- [25] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction”, in *Symposium on Geometry Processing*, Sardinia, Italy, 2006, pp.61-70.
- [26] W. E. Lorensen and H.E. Cline, “Marching cubes- A high resolution 3D surface reconstruction algorithm”, in *Proc. SIGGRAPH*, 1987, pp.163-169.
- [27] Yebin Liu, Qionghai Dai, and Wenli Xu, “Free viewpoint video data sets,”
<http://media.au.tsinghua.edu.cn/fvv.jsp>.
- [28] H. Kato, ARToolKit, <http://www.hitl.washington.edu/artoolkit/>