國立台灣科技大學
資訊工程研究所

碩 士 論 文

3D 點雲與 SIFT 特徵點室內影像定位之研究

3D Point Cloud and SIFT Descriptor Indoor Localization Research

研 究 生：陳致良
學 號：M9915057

指 導 教 授：項天瑞博士

中 華 民 國 一 百 零 一 年 七 月 十 日

3D 點雲與 SIFT 特徵點室內影像定位之研究

學生： 陳致良　　指導教授： 項天瑞博士

國立台灣科技大學資訊工程研究所

摘　　要

本篇論文探討如何利用單張彩色影像來重建出三維人臉模型。我們的方法是利用彩色影像來取代灰階值去建立張量模型 (tensor model)，在人臉資料庫中，是用典型相關分析 (Canonical correlation analysis) 是來建立彩色影像與深度資訊的對應關係，一旦建立好屬於各自的對應關係後，在重建人臉的過程中，就只需要單張彩色影像即可利用典型相關分析的對應關係來推算出正確的深度資訊。實驗中，我們的方法可以在不同的光線環境跟人臉角度得到不錯的效果。

3D Point Cloud and SIFT Descriptor Indoor Localization Research

Student: ZZhi-Liang Chen Advisor: Dr. Tien-Ruey Hsiang

Submitted to Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University of Science and Technology

ABSTRACT

This paper develops a tensor-based 3D face reconstruction approach from a single color image. Instead of the grayscale image, we also consider additional color factors in constructing the tensor model. Canonical correlation analysis is applied to establish the relationship between the color image and the depth information in the face database. During the face reconstruction, given a single color face image, the depth estimation is computed from the CCA-based mapping between the tensor models. Experimental results show our approach is better suited under different lighting conditions and poses.

誌　　謝

本論文能夠完成，首先要感謝的是指導教授項天瑞老師。老師嚴謹的治學態度，讓我不但在學術研究上學習到更謹慎的思考，也在日常生活上獲益良多。

感謝實驗室的同學們，實驗室的生活有苦有樂，有你們才讓我能撐得下這三年漫長的時間。感謝建群、松翰學長給我的指導。感謝益偉、崇峰、承誌、誠儀、慶豪學長以及恩緯、盈樽、嘉駿、世寬學長給我的指導。感謝實驗室的學弟妹們，訓哲、立昂、致良、青緯、庭耀、宗博、雅筑，常常幫了我不少忙。更要感謝我的同梯們，貴彥、薇穎、冠佑，我們同甘共苦，一起奮鬥，特別是在弄計畫的過程中，冠佑幫助了我很多，有你們在我才能走到這一步。也感謝好朋友志強跟我一起修正文法與用詞。

最後我要把我最深的感謝留給我的家人。謝謝我的爸媽跟兩位姊姊，你們讓我沒有經濟壓力地讀完這這碩士學位，也常常給我很多鼓勵，今天我終於拿到這個學位，終於可以讓你們放下心上的一塊大石頭了。

Table of Contents

List of Tables

List of Figures

Chapter 1　相關研究

1.1 尺度不變特徵向量 (Scale Invariant Feature Transform)

尺度不變特徵向量 (SIFT) 一開始由 lowe [1] 所提出，目的是尋找兩張圖片中的相似特徵向量來比對兩張圖片的相對關係，主要分成四個階段：

- (1) 區域空間極值分布
- (2) 特徵點定位與篩選
- (3) 特徵點方向分配
- (4) 特徵點描述向量建立

第一階段 (1) 區域空間極值篩選，先利用不同尺度間的高斯金字塔選擇區域中的最大極值，其高斯分布式子如下：

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} exp(-(x^2 + y^2)/2\sigma^2) \tag{1.1}$$

不同尺度的高斯分布利用摺積 (Convolution) 將影像模糊化。$I(x, y)$ 代表原始影像，$G(x, y, \sigma)$ 代表高斯函數：

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1.2}$$

再利用每組影像相鄰的高絲模糊影像進行高斯差分 (Difference-Of-Gaussian)，目的用於在集合內 4 組高斯差分影像中找出極值，式子如 (1.3) 所示：

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{1.3}$$

在此 k 為高斯模糊的尺度比值，設為 $\sqrt{2}$，若某個像素的極值為 26 個相鄰的像素中最大或最小的話，則此像素的位址即為區域極值的所在。

第二階段 (2) 特徵點定位篩選，其主要的目的在於找出真正有用的特徵點，在此特徵點的精準度必須要達到次像素的精度。有的特徵點其極值為低對比度的點，這時候這些低對比度的特徵點就會不予採用，剩下的特徵點即可為下一階段所使用。作法首先將 (1.3) 利用泰勒展開得到 (1.4):

$$D(x) = D + \frac{\delta D^T}{\delta X}X + \frac{1}{2}X^T\frac{\delta^2 D}{\delta X^2}X \tag{1.4}$$

式中 X 為極值 $(x, y, \delta)^T$，D 為高斯差分後的結果，再將 (1.4) 對 X 作偏微分可得 $\vec{X}$ 算出 X 為極值點的的偏移量。

$$\vec{X} = -\frac{\delta^2 D}{\delta X^2}^{-1}\frac{\delta D}{\delta X} \tag{1.5}$$

若是 $\vec{X} >= 0.5$，或是 $\sigma > k/2$，表示此區域極值點較靠近相鄰的點位，則需要再將此點移至相鄰的極值再經 (1.5) 計算後得到最佳的位置。若將 $\vec{X}$ 帶入 (1.4) 中，可得 (1.6) 我們所用來篩選的式子:

$$D(\vec{X}) = D + \frac{1}{2}\frac{\sigma D^T}{\sigma X}\vec{X} \tag{1.6}$$

利用 (1.6) 將求出的絕對值與其他絕對值相比，可將對比度小的特徵點刪除以達到過濾的效果。

第三階段 (3) 特徵點方向分配，目的在於當對比的圖片有旋轉或者是尺度上的變化，相同的特徵點為了保有相同方向的特性，必須賦予每個特徵點一組特定的方向。其做法則是利用統計的方式，將所有的梯度值以角度每 10 個單位做方位直方圖記錄，並且記錄每個梯度的強度，以 (1.7)(1.8) 表示：

$$\theta(x, y) = tan^{-1}(\frac{\delta L}{\delta y}/\frac{\delta L}{\delta x})) \tag{1.7}$$

$$m(x,y) = \sqrt{(\frac{\delta L}{\delta x})^2 + (\frac{\delta L}{\delta y})^2} \tag{1.8}$$

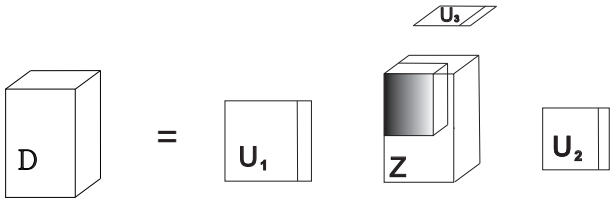


Figure 1.1: An N-mode SVD orthogonalizes the N vector spaces associated with an order-N order (the case $N = 3$).

Chapter 2　虛擬環境建置與定位方法描述

2.1　方法大綱

利用 3D 點雲環境座內定位，目的主要是為了能夠藉由模擬當時的重建出來的環境，來取得比一般影像定位更多的特徵點資訊。在之前的影像定位研究顯示，只要特徵點的數量越多，代表跟要定位中的照片越多的相似處，定位就能夠越準確。整個定位過程分成三大步驟：(1). 建置 3D 點雲環境 (2). 虛擬照相機設置 (3). 虛擬影像定位。基於這樣的理由，我們利用 Kinect 紅外深度攝影機這種能夠取得深度的資訊儀器幫助我們做出 3D 的點雲環境。有了 3D 的點雲環境，透過點雲中的坐標系來作均勻分布 (Uniform Distribution)，藉由這些有規律的分布區域來決定虛擬照相機的位置。因為每個照相機間距相同，所代表觀察到的區域都有固定大小與角度，藉由角度上的調整就可以取得比一般影像定位所照出包涵更廣的角度與更大的覆蓋空間，表示可以取出更好的照片。有了更好的照片，就可以減低之後定位所造成的誤差。圖 2-1 為整體 3D 點雲環境座內定位的過程，這些流程會在之後章節逐一解釋步驟與這些步驟的目的。

2.2　建立 3D 點雲環境

3D 環境的建置分成下列五個步驟：

- (1) 取得 Kinect 照片
- (2) 將偵測到照片中的特徵點作隨機抽樣一致演算法 (RANSAC)
- (3) 將組好的點雲作 Graph SLAM
- (4) 設置點雲涵蓋範圍的界線作界限範圍 (Bounding Box)
- (5) 調整坐標系

前面三個步驟的目的是建置初步的環境，環境有了完整虛擬實境的樣貌後，後面的步驟是為了微調之後撒上虛擬照相機的限制範圍。設限制的區域使虛擬照

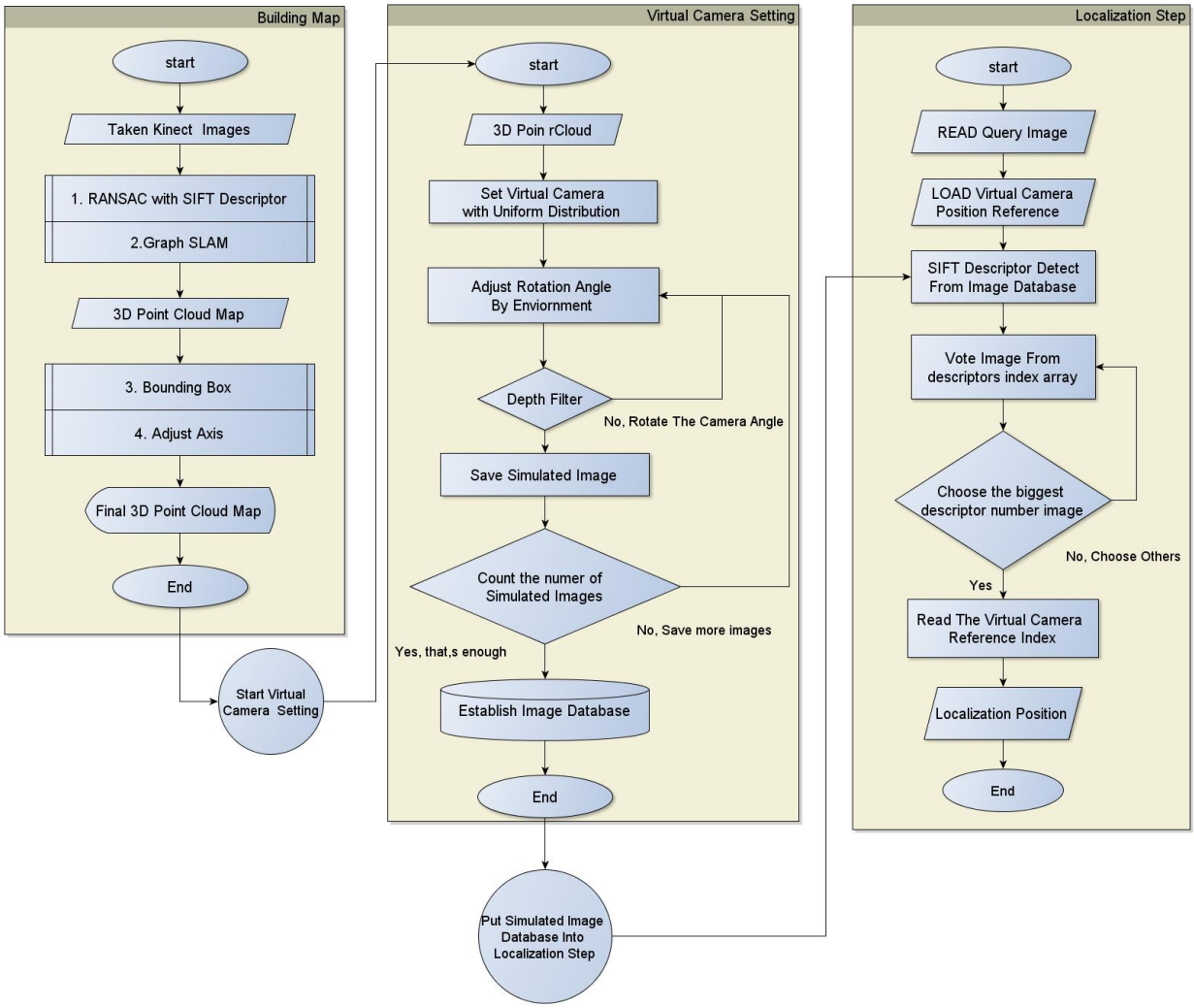


Figure 2.1: 3D 點雲環境座內定位整體流程圖

相機不會放置。在四周看不到任何景物的地方，避免製造出沒有使用價值的虛擬相片。接下來將如何製作 3D 點雲環境分成兩大部分做說明。

2.2.1 虛擬環境的建置

一開始先由環境中利用 Kinect 拍攝照片，照片取得的作法類似之前 [2] 的方法，在環境上對每個景物做一連串的拍攝，且每張對景物拍攝出來的照片都需要有一些相似之處。環繞整個環境的目的是為了不漏掉每個景物。當一個景物越多的環境代表所包含的特徵點越多，擁有豐富的特徵點數量代表之後在做 RANSAC 可以有更好的效果。像之前在相關研究的章節所述，一個密集的點群所找到的平面會越接近真實點群的表現，所做出的 Transformation Matrix 也會越精準，之後重合所做出點雲也會越準，不會有影像疊影或者是無法重合導致點雲中空、物體歪斜扭曲的現象產生。連續的環繞拍攝是為了在之後 RANSAC 的比對有循序的排列，保持每一張影像相對位置關係沒有錯誤。RANSAC 最怕沒有順序的影像排列，沒有順序就沒法找出影像的相對位置，也就是說 Transformation Matrix 所做出的點雲位置會和實際景物在環境中的位置相差甚遠。拍攝照片在點雲建置的步驟中是影響最大的因素，其中可能光線的不足或是玻璃的反射等一些外在的因素都會導致之後在建置點雲的困難，所以在作拍攝時最好都避免這些不利的因素。

有了拍攝的照片組後，利用這些照片取得特徵點的位置，再用這些位置作特徵點求取 RANSAC，使得每一張影像都能夠在 3D 坐標系在對的位置中重合。在這裡使用 MRPT (The Mobile Robot Programming Toolkit) 的應用程式介面 (API)，當我們將每張圖片由尺度不辨特徵向量 (Scale Invariant Feature Transform) 找出來的特徵點作配對，當我們求出來配對關係之後，再將這些配對關係作最小平方法 (Least Square Error) 求出想要的平面，根據不同平面求出 Transformation Matrix ，最後的結果回傳之前照片影像之位置的絕對關係，我們稱為 Global Pose ，Global Pose 包含照片在三維座標以及照片在當時拍攝的角度。Global Pose 在 2D 影像定位需要以它作每張影像定位的起始原點，在其他地方像是點雲之後需要調整亦或是找出界限範圍, 都需要利用 Global Pose 的起始原點當做參考, 但在我們的研究中，我們是使用虛擬相機的 Virtual Camera Pose ，至於 Virtual Camera Pose 的設置方式會在之後詳加說明。有了 Global Pose 的位置後，最後我們利用 Graph SLAM 將點雲作最後的調整。

Figure 2.2: 初步建置好的點雲環境

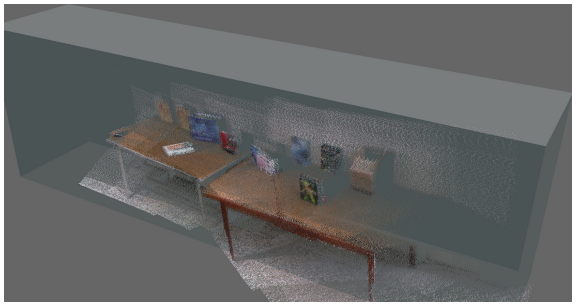


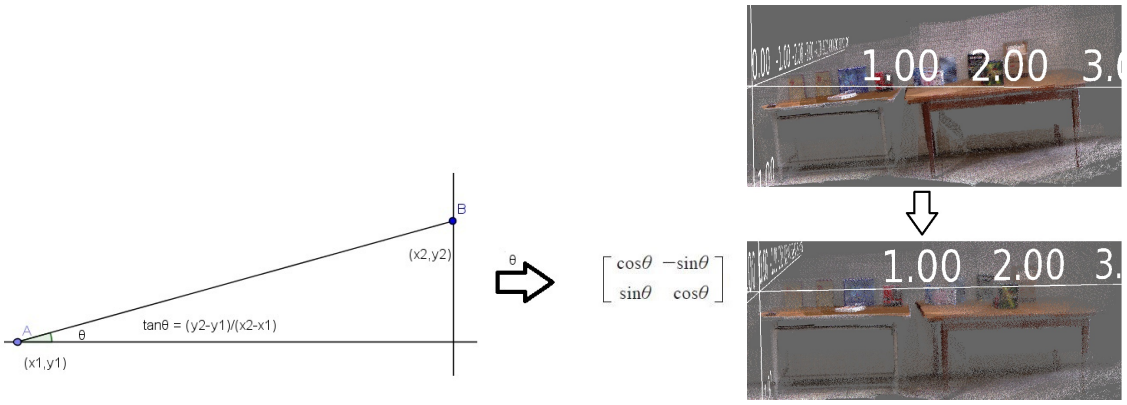
Figure 2.3: 將點雲給予界限範圍

Graph SLAM 描述

2.2.2 點雲界限範圍制定及大小調整

前三個步驟將點雲完成之後，在之後的過程會將點雲的範圍用 Bounding Box 來制定界線範圍，用照相機的 Global Pose 調整角度。這些目的是為了虛擬照相機能夠在限定的範圍擺放位置，而不使有照相機放置在點雲外面，無法產生出有效的虛擬影像作定位。具體的做法如下，當我們讀入整個點雲之後，找出位於坐落在點雲中最大與最小的 X 及 Y 座標，有了這四個座標之後，利用點雲所求出的原點，將最大與最小的 X 及 Y 座標互相相減，所求出的長度即為 Bounding Box 的長寬。有了這些長度之後，就可以知道整個點雲或者是說環境的長寬距離為多少，在之後可以防止虛擬照相機坐落在散布點雲以外的位置。

知道界限範圍之後，做出來的點雲可能會因為之前 Kinect 攝影機的 Global



(a) 使用攝影機的 Global Pose 求得 $\tan\theta$ (b) 坐標軸角度差異

Figure 2.4: 調整點雲坐標軸角度方法

Pose 歪斜分布而使得點雲也會有歪斜的狀況產生，這時候就必須將點雲作角度上的調整。這個步驟是為了之後的虛擬相機在照相時不會因點雲角度歪斜而使得照出來的角度與真實相機的角度差異過大，減少許多對應的特徵點，導致定位的誤差產生。具體的做法先看出點雲分布的情況是往哪個方向歪斜，利用 Global Pose 來算出兩個 Kinect 攝影機角度的 $\tan\theta$，求出 θ 之後，將每個點雲帶入旋轉矩陣利用算出來的 θ 將每個點旋轉至與座標軸平行的角度。調整完正確地角度之後，點雲的界線範圍與旋轉角度都與座標軸方向一致後，就可以準備虛擬相機的準備工作。

2.3 虛擬照相機設置及影像資料庫建置

在建置完環境之後，接下來利用這個章節來描述如何決定虛擬照相機的位置、角度以及虛擬照相機成像的原理。與一般 2D 影像定位方法不同的是，傳統的 2D 影像定位內的影像資料庫，大部分都是利用隨機位置取得影像資料，而在我們的作法是先利用均勻分布 (Uniform Distribution) 設置虛擬照相機的位置，再利用隨機分布的角度來決定照相機拍攝的角度。依照這樣的作法，我們能取得比一般影像定位更多的環境資訊，而這些資訊都是利用點雲所產生的，不必再額外人工存取 kinect 攝影機的影像資料，我們所要輸入的資料只需要點雲就可以了。之後我們將分成四個部分描述虛擬照相機的設置：

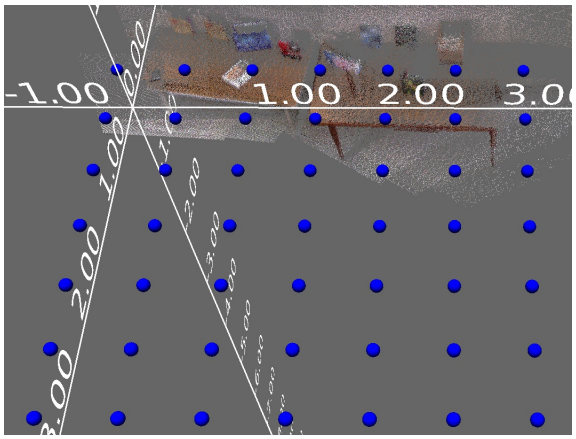


Figure 2.5: 在點雲上設置虛擬照相機位置

- (1) 均勻分布設置虛擬攝影機
- (2) 虛擬照相機成像原理
- (3) 根據深度來調整攝影機角度
- (4) 儲存虛擬照相機圖片

2.3.1 均勻分布設置虛擬攝影機

上一個章節中，我們完成了實驗環境的建置，也就是點雲的資料，將點雲切割成數等分的區塊，每個區塊都設置一個虛擬照相機，目的為了環境內每個景物都有充分去拍攝而取得足夠的特徵點。作法依據環境而有所改變，首先將長分成 8 個等分、寬分成 7 個等分，這樣每個等分都會有一樣的距離間隔。在每個間隔放置虛擬照相機，接下來再隨機分布照相機角度，完成虛擬照相機布置的工作。隨機分布照相機角度比一般 2D 影像定位所建置的資料庫比較有更寬廣的角度。一般影像資料庫可能只針對特定區域的特徵點作取樣，而導致特定區域內的影像定位效果非常好，但在其他區域卻沒有足夠的影像特徵點資料，使得定位誤差範圍過大，透過均勻分布不會有部分景物或場景沒有被拍攝到，也可以增加定位的覆蓋率。

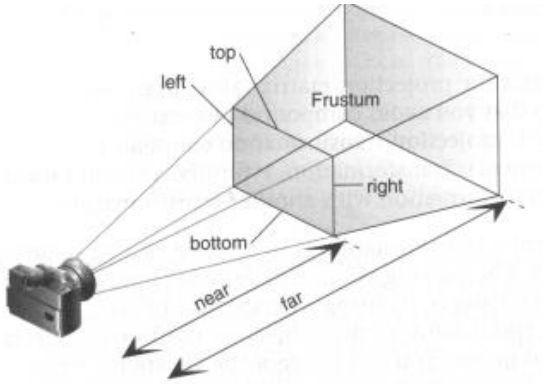


Figure 2.6: 攝影機的攝影近裁面 (near) 與遠裁面 (far) 表示

2.3.2 虛擬照相機成像原理

當虛擬相機位置固定之後，接下來利用虛擬相機拍攝照片，虛擬相機主要由 OpenGL 這套 API 來實作，透過攝影機的影像角錐來模擬相機的成像，將角錐內 3D 影像投影成 2D 影像，模擬照相機所照出的照片。在 OpenGL 的坐標系，先將視野調整到虛擬照相機的位置，再利用 glFrustum 的矩陣取得相機影像角錐，這個矩陣目的在於模擬相機光線經過透鏡成像，在投影到平面影像呈現看出來的照片，矩陣表示法如下：

$$glFrustrum = \begin{pmatrix} \frac{2near}{right-near} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & \frac{far+near}{far-near} & \frac{2far \times near}{far-near} \\ 0 & 0 & -1 & 0 \end{pmatrix} \tag{2.1}$$

將座標轉為齊次座標後，利用攝影機的攝影近裁面 (near) 與遠裁面 (far) 的相似三角形來推出這個矩陣，這個矩陣會將角錐內的景像投影到 2D 平面上，即可完成虛擬照相機的建置。這部分攝影機的焦距設定，以及解析度都參照 Kinect 紅外深度攝影機的參數設定。與真實相機的相機參數相同，再接下來實驗就相機拍出來的照片作對照，做為比較的依據。當我們截取下攝影機的圖片後，會先將角錐內影像的深度作平均，為的是求取平均深度來看虛擬相機取的位置會部會太逼近虛擬環境內的景物，或是虛擬相機位置太靠近點雲邊界，這部分會在之後的章節作說明。

2.3.3 根據深度來調整攝影機角度

當虛擬照相機的圖片擷取出來後，因為拍照的相機深度過淺，而導致拍攝的景物無法辨識，這時候我們利用深度過濾的機制來將照相機取得角度作過濾。一般深度 buffer 分為 z-buffer 與 w-buffer 兩種，先從兩種不同的深度分辨方式作探討：

首先作關於深度的計算，利用四維座標軸 (x, y, z, w) 表示三維座標軸 (x', y', z') 的點，空間關係的表示法為：

$$\begin{cases} x' = x/w \\ y' = y/w \\ z' = z/w \end{cases} \tag{2.2}$$

根據 figure 2-6 的示意圖表示，$Z_n = near$ 面的 z 範圍，$Z_f = far$ 面 z 範圍，$w = \frac{2 \times Z_n}{right - left}$，$Q = \frac{Z_f}{Z_f - Z_n}$ 所以由 z 座標求得 w 縮放的比例，式子可以寫為：

$$w = \frac{Q \times Z_n}{(Q - Z)} \tag{2.3}$$

z-buffer 是保存經過 glFrustrum 投影變換後的 z 坐標，投影後物體會產生近大遠小的效果，所以距離眼睛比較近的地方，z 坐標的分辨率比較大，而遠處的分辨率則比較小。換句話說，投影後的 z 坐標在其值得分布上，對於景物對眼睛的物理距離變化來說，不是線性變化的（即非均勻分佈），這樣的一個好處是近處的物體得到了較高的深度辨識，但是遠處物體的深度判斷可能會出錯。

w-buffer 保存的是經過投影變換後的齊次坐標系中的 w 坐標，而 w 坐標通常跟世界坐標系中的 z 坐標成正比，所以變換到投影空間中之後，其值依然是線性分佈的，這樣無論遠處還是近處的物體，都有相同的深度分辨率，這是它的優點，當然，缺點就是不能用較高的深度分辨率來表現近處的物體。

針對兩種不同的深度 Buffer 比較，因為我們的做法是來判別景物是否距離鏡頭過近，所以在深度判斷上是採用 z-Buffer 的作法，當我們判斷鏡頭與物體距離實際深度小於 80 公分時，我們會將照相機鏡頭角度轉向 180 度，也就是正後方來重新拍攝。

Figure 2.7: 內差法補強前後的差異圖

2.3.4 儲存虛擬照相機圖片

當已經決定取好的照片之後，利用虛擬照相機將取出的照片來儲存至影像資料庫，來進行接下來定位的前置作業。虛擬影像儲存是透過虛擬照相機鏡頭裡的每一個 pixel 寫入相片裡頭，主要做法如下。當從 z-buffer 讀出來的深度錯誤時，代表這個 pixel 對應在點雲上是一個黑點或者是說根本沒有點雲的資訊，則以黑色為代表，當深度沒影錯誤時，則代表它具有實際點雲的資料，我們找出點雲對應點的顏色資訊，寫入圖檔裡，這樣即可完成初步的虛擬相片。根據上述的方法，還會遇到透視的問題，就是說原本不應該出現的景物因為深度有誤差，而原本在障礙物之後的物體卻跑在障礙物之前，像是穿透障礙物一樣，例如 figure 2-9 原本不該出現桌子的地方，因為發生了透視的現象而出現了桌子。改進方法為根據周圍的深度來做內插補強。

虛擬影像的資料量因環境而變，主要根據 Global Pose 在每個位置上取出相隔 120 度的兩個不同角度的相片，在一般情況下環境中取出 50 點的 Global Pose，所以總共會有 100 張的虛擬相片。藉由這些虛擬相片，我們取得了環境所在內的不同位置與不同角度的資料，比起一般的影像定位資料多出了更豐富的特徵點資訊。之後的實驗可以比較出來，在不同位置以及距離特徵點的遠近對定位會帶來什麼樣的影響。到了最後定位的流程，將介紹虛擬影像的定位方法。

2.4 虛擬影像定位

Bibliography

[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[2] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox, "Interactive 3D modeling of indoor environments with a consumer depth camera," *Proceedings of the 13th international conference on Ubiquitous computing - UbiComp '11*, p. 75, 2011.

[3] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," *Computer Vision–ECCV 2008*, pp. 504–513, 2008.

[4] P. Viola and M. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.