中華民國第 63 屆中小學科學展覽會

作品說明書

高中組　電腦與資訊學科

第三名

052501

把影像提升一個維度－影片圖片３Ｄ化

學校名稱：新北市立板橋高級中學

作者：	指導老師：
高二　曹嵎榿 高二　詹糧有	鄭政富

關鍵詞：Python、深度估計、３Ｄ還原建模

摘要

本研究以稠密式深度預測為基礎，進行環繞掃描３Ｄ還原建模。在這項研究中，我們擇手機作為唯一的外部數據提供設備。首先用手機同時錄影和記錄角度變化、位移數據後，我們同步各數據的時間點，並計算每一幀影像的位置，接著將拍攝的影像輸入深度預測模型轉換成深度圖並將其儲存為點雲，最後利用三維旋轉和平移矩陣將點雲轉回正確的位置以進行疊合。我們開發的３Ｄ還原建模系統能夠輸入影像、特定時間點的位置和角度變化來建立３Ｄ模型。此系統可以解決結構光掃描儀需要額外標記點的劣勢，且在機動性上較傳統的點對點雷射掃描儀更方便。我們希望可以在未來將這套系統集成為一個手機應用程式，方便使用者在手機上進行相關操作。

壹、前言

一、研究動機

在對物件拍照時想到，如果照片可以看出遠近的話，就能更能知道這張照片裡的東西與環境長甚麼樣子。雖然市面上已經有很多以雷射掃描環境的儀器，但這些裝備體積龐大、不方便操作，因此我們想透過多個角度的相對距離計算出一個區域內的 3D 圖。在搜尋資料的過程中我們發現市面上有許多不同原理的測距儀及環境掃描器，其中價格便宜的測距儀測量精度太低，且容易被各種干擾因素影響如：紅外線、超音波等，價格高的雷射測距儀雖然精度較高，但缺點是昂貴的價位及複雜的保養。因為上述原因，我們決定使用另外一種方法──透過輸入多個角度的照片讓電腦計算出 3D 模型。這個做法無須用到測距儀，只需一台畫質足夠的拍攝設備，大大降低了生成 3D 模型所需的難度。

二、目的

(一) 製作鏡頭相對位移演算法

在不同的裝置中取得數據的狀況亦不相同，除了基礎的影像輸入外，是否具備位移及角度變化測量能力也是一大關鍵。

(二) 製作 3D 模型轉換器

結合以上移動及角度變化資料，透過處理後可得正確之 3D 模型

三、文獻回顧

在本研究中，希望將研究成果與雷射掃描儀一起比較。同時也簡單介紹所需軟體。

(一) 常見 3D 掃描儀

本研究希望以不同方式來達成 3D 還原重建，但市面上已存在多種原理不同之雷射掃描儀。為了方便後續比較，我們在 (表 1) 中簡介最常見的兩種技術為結構光與雷射掃描之技術與效果差異。

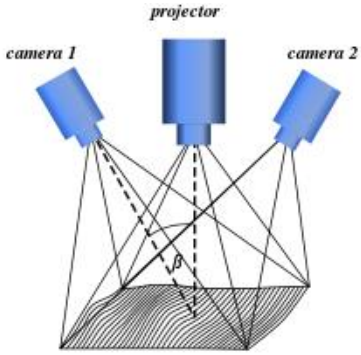
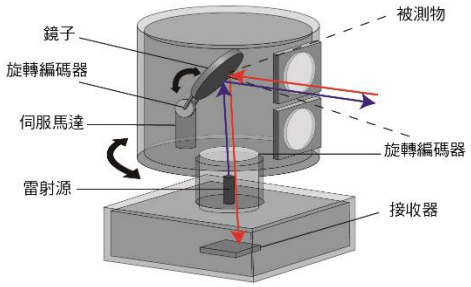
	結構光	點對點雷射掃描
示意圖		
原理簡介	將光帶投影到待測表面上，從其他角度觀測光帶形變，透過形變量與雷射測距可進行表面的幾何重建。組合多角度數據並配合標記點完成建模。	透過光波的反射逐一測定點到鏡頭間的距離，整理並統合所有數據集成點雲再進行建模。
最大有效距離	約 1 公尺	100~500 公尺 (依機器決定)
鏡頭是否要定點	無須定點 (可手持)	須定點 (機器自身旋轉)
是否需要標記點	需要 (反光圓貼紙)	不需要
受環境光線影響	主動投影，不受影響	主動雷射，不受影響
機器大小	小	大 (含腳架)

表 1 、結構光掃描儀與雷射掃描儀技術對比

而本研究可以透過位移和角度變化數據以連續的影像建立模型，可克服使用結構光掃描儀時需要標記點的麻煩；同時也僅以手機作為偵測工具，與傳統點對點雷射掃描儀相比大大減少儀器體積，讓運作整套系統更加便利。

(二) Sensor logger 偵測軟體 簡介

Sensor logger 是一款偵測手機物理量變化的軟體，包含空間加速度 (圖 1- a) 、三軸角度測量 (圖 1- b) 等偵測功能，是唯一一個能在偵測的同時也能進行攝影 (圖 1- c)的軟體，解決了部分手機無法多工攝影的問題。它可以在測量階段結束後將數據輸出為 csv 檔案，方便電腦讀取。

(a) (b) (c)

圖 1、加速度偵測(a)、角度變化偵測(b)、錄影影像(c)

(三) MiDaS 深度預測 簡介

與其他單目深度估計模型不同，MiDaS 有著大量針對不同運算效能及品質所分化的模型，且輸出僅一張深度圖，方便讀取以及運算。MiDaS 在[1]中提出，是一種用 CNN (Convolutional Neural Network , 卷積神經網路) 進行單目深度估計的方法。可透過輸入的 RGB 照片估計每個相素與鏡頭的相對距離，將數據組合後輸出一張深度估計圖。圖 2 為彩色輸入與輸出對比圖。

圖 2、輸入 RGB 影像與輸出深度圖對比(圖片來源：MiDaS-github)

貳、研究設備及器材

<table>
<tr><td colspan="4">硬體</td></tr>
<tr><td>電腦</td><td> 
圖 3 、電腦</td><td>手機</td><td> 
圖 4、手機</td></tr>
<tr><td>Nova Pi
主機板</td><td> 
圖 5、Nova P i 主機板</td><td>Makeblock Smart Servo MS 12A</td><td> 
圖 6、Makeblock Smart Servo MS 12A</td></tr>
<tr><td colspan="4">軟體</td></tr>
<tr><td>Anaconda</td><td> 
圖 7、anaconda</td><td>Python</td><td> 
圖 8、python</td></tr>
<tr><td>Pytorch</td><td> 
圖 9、pytorch</td><td>Tensorflow</td><td> 
圖 9、tensorflow</td></tr>
<tr><td>Pandas</td><td> 
圖 10、pandas</td><td>OpenCV</td><td> 
圖 11、opencv</td></tr>
</table>

參、研究過程或方式

一、架構

圖 12 為本研究之流程規劃，經過相關文獻及理論推導後，依序進行資料蒐集和前處理、深度圖轉換、主要演算法運算、以及結果輸出。

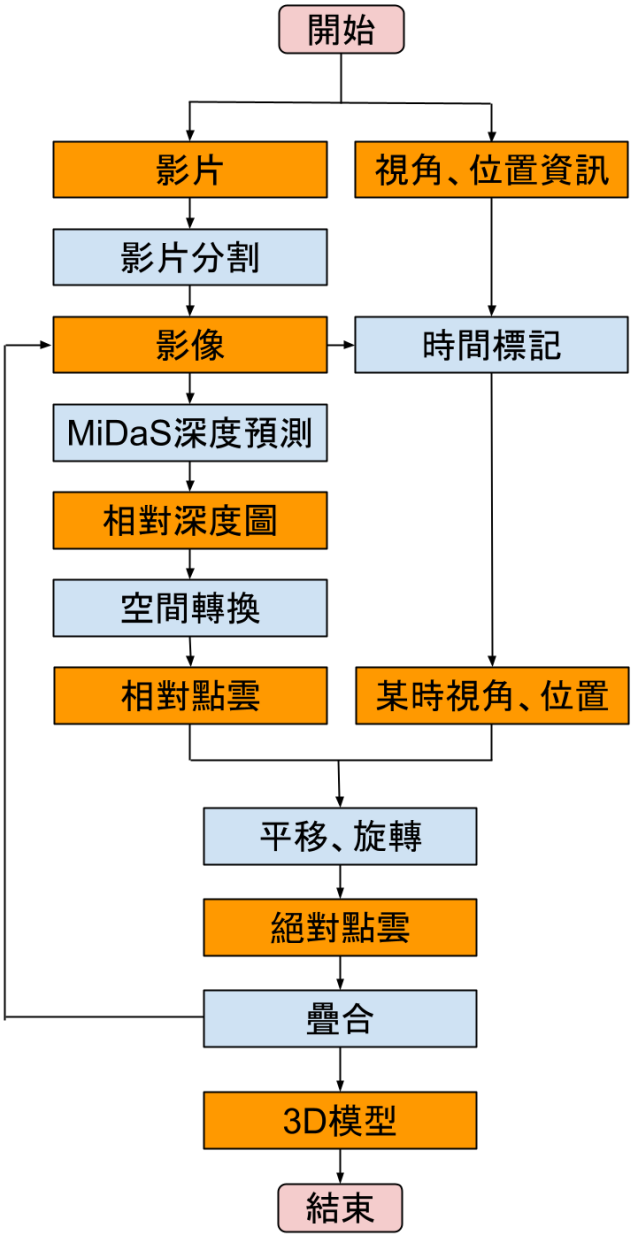


圖 12、研究流程圖

二、選定適合的 MiDaS 模型

在 MiDaS 中有針對不同硬體效能及應用場景所訓練的多種模型 (圖 13)，使用前需先根據硬體相容性與效能選定合適的模型，以免發生卡頓、噪點過多、畫面品質差等問題。此研究初期使用 v2.0_large_384 模型，但因之後推出 v3.0 及 v3.1 版本，經由原作者之測試數據可發現 v3.1 較 v2.0 版本在運行速度和輸出品質皆有提升，且 v3.1 使用的 transformer encoder 架構較 v2.0 使用的傳統 encoder-decoder 架構，更有效率且全域訊息流失較小。

我們取較穩定的兩模型 beit-L 512 與 swin2-L 384 進行單張深度圖還原模型比較後發現 beit-L 512 在廣闊環境較遠物件的估計表現極佳，但近距離估計表現不如預期，而 swin2-L 384 雖在廣闊環境中易忽略較遠物件，但中近距離估計較為準確。故選用 MiDaS v3.1 版本的 v3.1 Swin2-L 384。

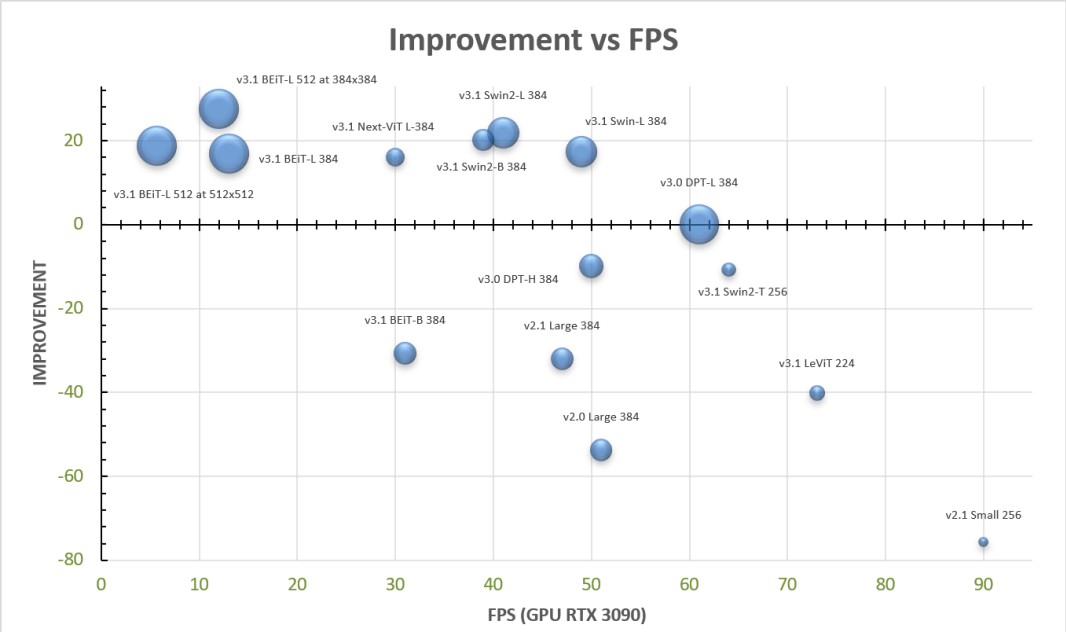


圖 13、各模型對比 v3.0 原始模型效能及輸出 FPS 關係圖 (圖片來源 : github-MiDaS)

三、空間中的仿射變換

在有鏡頭移動的狀況下，鏡頭可以在三維空間中任意方向的平移或轉動。若直接將所有點進行疊合會使大部分的點出現在錯誤的位置，可以透過三維仿射變換中的平移及旋轉，得到相對初始的點集合在三維空間中正確的位置擺放。疊合多張組經過此方法處理之點雲，可得完整 3D 模型。

(一) 旋轉

已知任一向量在三維空間中的旋轉都可以由分別沿 x , y , z 軸旋轉的角度表示。

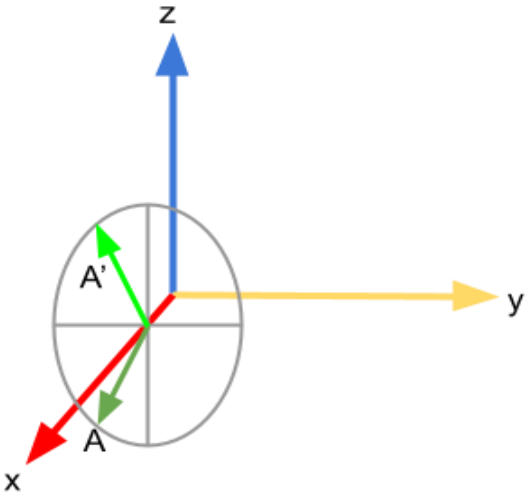


圖 14、三維向量 A 及旋轉後 A'

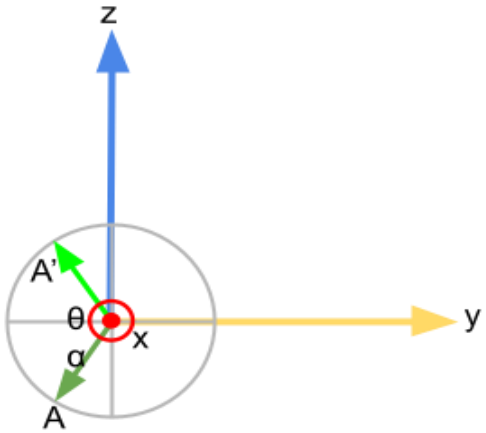


圖 15、向 x 軸正向之 A 與旋轉後 A'

圖 14 中有一三維空間中向量 A (x , y , z)與 y 軸正向夾角 α，沿 x 軸旋轉 θ 後為空間向量 A' (x' , y' , z')， A' 之旋轉半徑為 R。

由 x 軸方向觀察後 (圖 15) 可列：

$$x' = x$$
$$y' = R\,cos(\theta + \alpha)$$
$$z' = R\,sin(\theta + \alpha)$$

觀察 圖 14 , 15 後可知：

$$y = R\,cos(\alpha)$$
$$z = R\,\sin(\alpha)$$

根據三角函數和差角公式化簡後得:

$$y' = R\cos(\theta+\alpha) = R(\cos\alpha\cos\theta - \sin\alpha\sin\theta) = y\cos\theta - z\sin\theta$$
$$z' = R\sin(\theta+\alpha) = R(\cos\alpha\sin\theta + \sin\alpha\cos\theta) = y\sin\theta + z\cos\theta$$

將以上整理成矩陣形式得：

沿 x 軸旋轉
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

同理，對於沿 y 或 z 軸旋轉可以下列矩陣形式表達：

沿 y 軸旋轉
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

沿 z 軸旋轉
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

總結以上，假設鏡頭分別沿 x , y , z 方向旋轉 α , β , γ，其以矩陣形式表達為：

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & 0 \\ \cos\beta\sin\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & -\sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & 0 \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(二) 平移

將每個不同方向的位移量直接與旋轉後座標矩陣相乘即可得。

假設 x , y , z 方向分別平移 a , b , c。其以矩陣表達為：

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

四、針對鏡頭不同的移動數據狀況建立 3D 模型

使用手機蒐集數據時，除了鏡頭輸入外，也可以提供鏡頭相對位移及角度變化等資訊。我們可以充分利用此資訊來大幅增加輸出之 3D 模型的準確度。

(一) 手機鏡頭方向與位移、旋轉方向定義

在原廠的設定中手機可以偵測沿 X , Y , Z 軸不同方向的位移及旋轉。 +X 及 +Y 與螢幕共平面，在手機立直擺正時 +X 朝右、 +Y 朝上；+Z 垂直螢幕平面 (圖 16) ；而旋轉的方向亦是如此 (圖 17)。

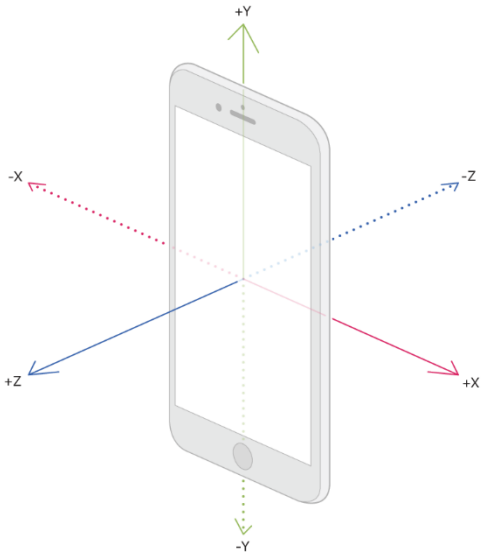


圖 16、XYZ 加速度方向定義

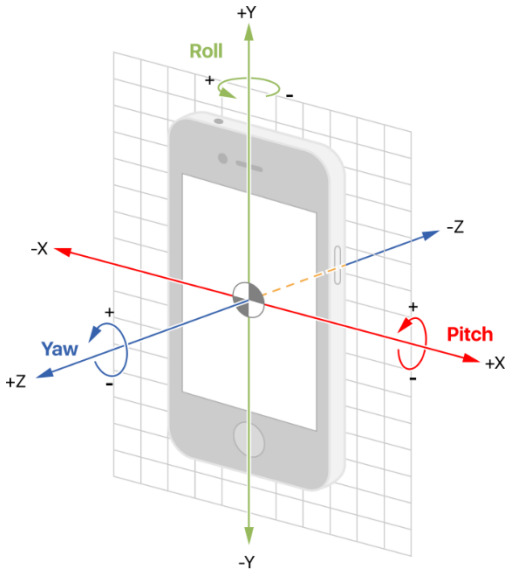


圖 17、XYZ 旋轉方向定義

（圖片來源：apple 官網）

(二) 影像、位移及角度變化資訊擷取

用 Sensor Logger 建立新的自訂偵測並設定偵測頻率為 60Hz ，使手機同時偵測加速度變化及角度變化，並同時開始以 2fps 錄影。

將取得手機之加速度資料經過兩次梯形法積分後得到每 1/60 秒所在的位置，計算後將其放在 excel 中暫存 。

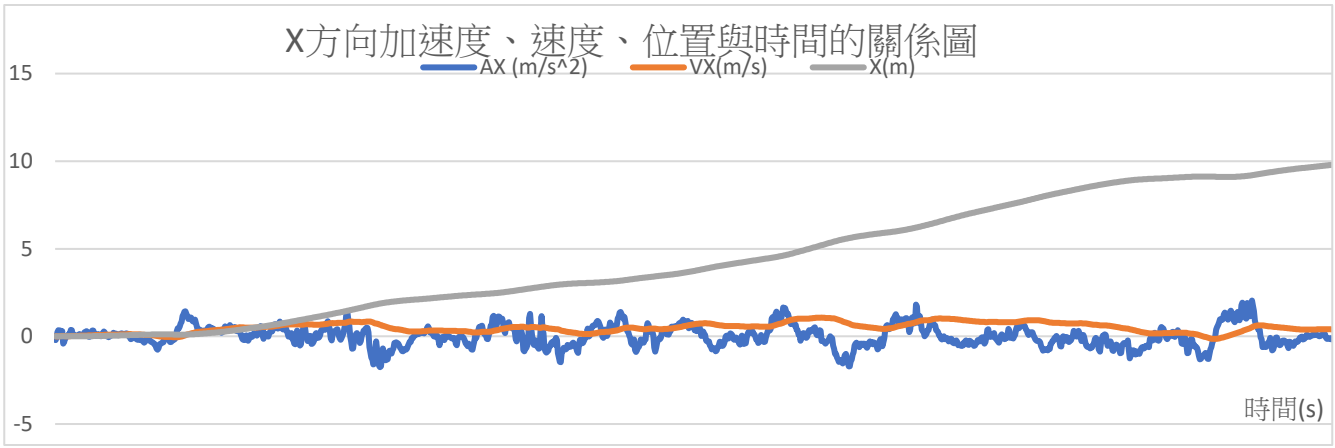


圖 18、X 方向加速度、速度、位置與時間關係圖

圖 19、Y 方向加速度、速度、位置與時間關係圖

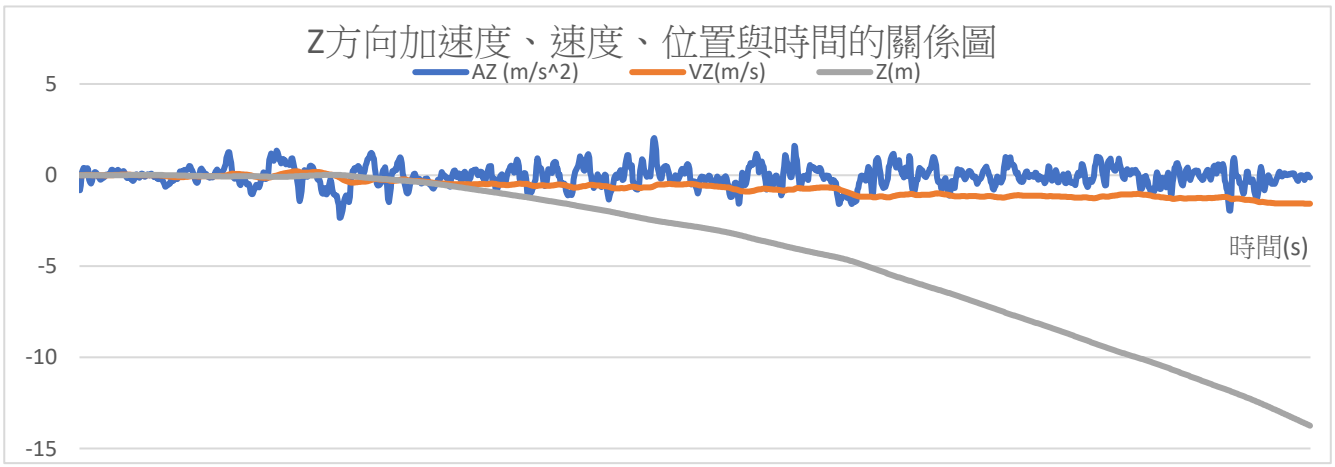


圖 20、Z 方向加速度、速度、位置與時間關係圖
(以上數據皆來自測試輸入)

(三) 影像、位移及角度變化資訊輸入

不同的裝置中，除鏡頭輸入外，特定裝置可以提供鏡頭相對位移及角度變化等資訊。在有提供此資訊的狀況，我們可以充分利用他們來大幅增加輸出之 3D 模型的準確度；如果沒有提供這些資訊，我們就需要從輸入的影像中計算位移和角度變化數據。

1. 影像輸入

使用 python 之 OpenCV 套件連接在手機上錄好的影片作為影像輸入，並將影像分割成連續照片以 numpy array 形式暫存 (圖 21) 。

```
cap = cv2.VideoCapture('test')
while True:
    ret, frame = cap.read()
    cv2.imshow('frame', frame)
    F=frame
```

圖 21、影像輸入片段程式碼(60fps 影片輸入)

2. 位移及角度變化輸入

將先前錄製之數據輸出並使用 Pandas 套件將 excel 資料輸入至程式 (圖 24) 中暫存，供後續使用 (圖、22) 。

```
dfA= pd.read_excel(r'.\testfile.xlsx',sheet_name = 'Linear Accelerometer')
dfG= pd.read_excel(r'.\testfile.xlsx',sheet_name = 'Gyroscope')
for row in range(dfG.shape[0]):
    s_rowA= dfA.iloc[row,:]
    s_rowG= dfG.iloc[row,:]
    if s_rowA[0]!=s_rowG[0]:
        break
    else:
        Time=s_rowA[0]
        xm,ym,zm=s_rowA[1],s_rowA[2],s_rowA[3]
        xr,yr,zr=s_rowG[1],s_rowG[2],s_rowG[3]
        pkg=(Time,xm,ym,zm,xr,yr,zr)
```

圖 22、將位移及角度變化資料暫存

(四) 取得深度圖

將 RGB 影像透過 MiDaS 可得到一個將深度數據以灰階形式顯示之圖形。在單一影像的情況下，攝影機能抓取到的影像深度數據只有單一方向，在計算後也只會得到單一方向的深度數據，故僅須針對此組深度數據及位置進行建模即可。

從 MiDaS 的輸出 (圖 23) 我們可以得到帶有每像素相應深度數據的二維陣列 D，每一項表示為 $D_{i,j}$ ，透過整理並重新編排此陣列後將每一組深度數據表示為三維空間中的一點 $d_k (i , j , D_{i,j})$ ，最後透過此組三維點座標數據並建立三角網格模型 (圖 24)。

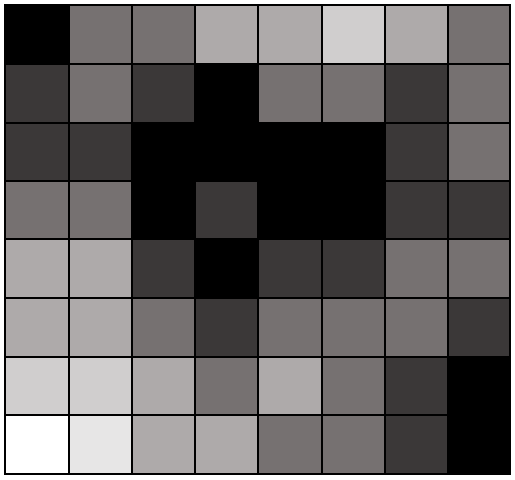


圖 23、以逆灰階表示的二維陣列 D
(擷取自測試輸出之 8*8 區塊)

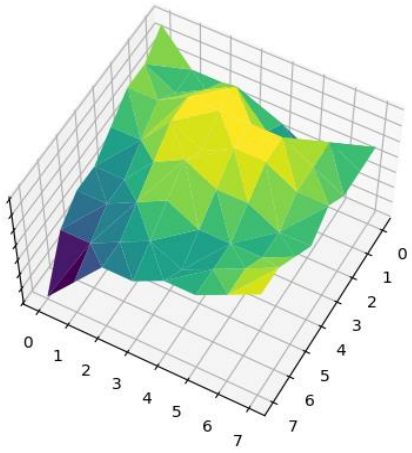


圖 24、以 D 建模後之 3D 模型

(一) 將圖 24 以位移及角度變化數據進行仿射變換(圖 25)

將讀取到的相對位移、視角變化數據和深度圖輸入先前推導出的仿射變換矩陣公式中，即得某影像的深度圖與相對初始影像的深度圖在三維空間中正確的相位置後，疊合所有點之位置。

```
def Atrf(angle,dx,dy,dz,A):
    a,b,c=angle[0],angle[1],angle[2]
    Rmat=[[cos(b)*cos(c),sin(a)*sin(b)*cos(c)-cos(a)*sin(c),cos(a)*sin(b)*cos(c)+sin(a)*sin(c),0],
          [cos(b)*sin(c),cos(a)*cos(c)+sin(a)*sin(b)*sin(c),cos(a)*sin(b)*sin(c)-sin(a)*cos(c),0],
          [-sin(b),sin(a)*cos(b),cos(a)*cos(b),0],
          [0,0,0,1]]
    Smat=[[1,0,0,dx],[0,1,0,dy],[0,0,1,dz],[0,0,0,1]]
    A=m.mult(Rmat,A,1)
    out=m.mult(Smat,A,1)
    return out
```

圖 25、仿射變換公式的程式碼

肆、研究結果

一、實際掃描結果量化定義

在掃描完成後，需要實際的數據對成果進行量化與比較。在此，我們將透過以下函數進行數據分析和比較。

(一) 高度占比 y :

高度占比 h 表示由模型最低點到某位置 H 占模型總高H_0的比例為 h。

$$h = \frac{H}{H_0}$$

(二) 截面積覆蓋率 C_h ：

定義截面積覆蓋率 C_h 表示還原模型與初始影像在高度佔比 h 時的截面$A_{r,h}$與$A_{0,h}$透過對角線交點疊合後的交集$A_{r,h} \cap A_{0,h}$與初始影像截面積的比例。

$$C_h = \frac{A_{r,h} \cap A_{0,h}}{A_{0,h}}$$

(三) 截面積溢出率 O_h：

定義截面積覆蓋率 O_h 表示還原模型與初始影像在高度佔比 h 時的截面$A_{r,h}$與$A_{0,h}$透過對角線交點疊合後的差集$A_{r,h} \setminus A_{0,h}$與初始影像截面積的比例。

$$O_h = \frac{A_{r,h} \setminus A_{0,h}}{A_{0,h}}$$

二、實驗步驟

經過初步測試後我們發現若只用手機的加速度去進行位置的計算，手機位置的誤差會隨著掃描的時間拉長而增加，嚴重影響建模的成果。於是決定以定速軌道掃描，直接輸入系統固定速率與位置訊息，避開加速度誤差大的問題，仍保留手機偵測之角度數據輸入。最後比較不同週期的掃描與還原建模效果。以下是我們的實驗步驟。

(一) 選定生活中常見且材質，形狀、材質不同的物體

物體１：圓柱體

物體２：正方體

物體３：透明盒子

(二) 製作定速軌道

使用 Nova Pi 主機板連接 Makeblock Smart Servo MS 12A 帶動平台定速轉動

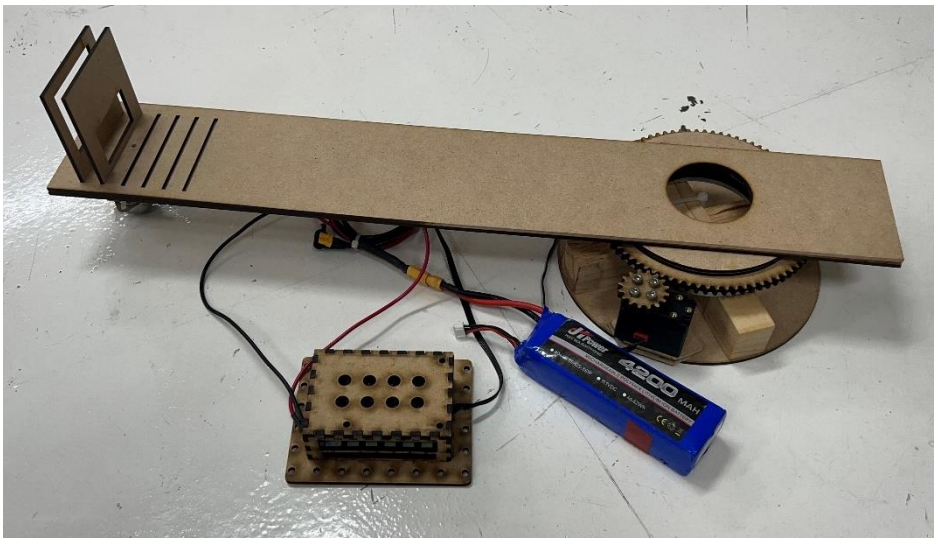


圖 26、定速軌道裝置圖

(三) 將物體和手機放上測試裝置

(四) 啟動裝置並使用手機開始以 2 fps 錄製

一共錄製四次，分別以 15、20、30、60 秒為週期。

(五) 將錄製的照片及轉動數據送進電腦裡

送進電腦後先輸入 MiDaS 進行深度圖轉換，並輸出點雲。

(六) 運行系統並建立 3D 模型

將點雲透過旋轉與平移矩陣變換後與先前點雲疊和。

三、手機在特定週期中不同物體的掃描結果

由於設計定速軌道裝置時的齒輪比為 1:5，又 Makeblock Smart Servo MS 12A 之轉速有最低轉速限制 5 rpm ，所以馬達轉速以 5 rpm 為單位。則馬達以 5 、10、15、20 rpm 旋轉時，定速軌道以 60、30、20、15 秒為週期旋轉。

(一) 圓柱體還原建模

在圓柱體的建模過程當中，我們發現到單一平面時 MiDaS 的深度預測極為精準，但估計曲面時的表現不佳，輸出之深度圖沒有層次感 (如 圖)，趨近於平面，再疊圖過程中可能會使模型表面不平整、圓滑。

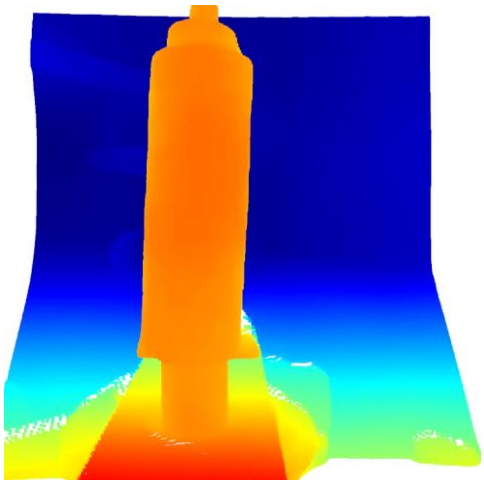


圖 27、一張圓柱體深度圖的點雲

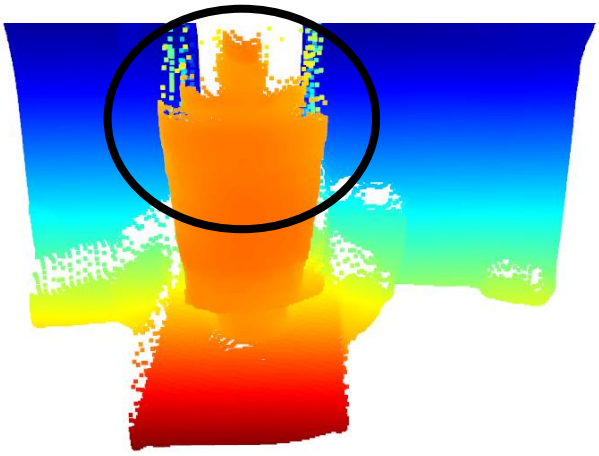


圖 28、圓形曲面近似平面(圈起處)

但實際建模後發現曲面雖近似平面，但左右兩側的點皆在正確位置，在所有數據都疊合完成後，模型外圍會由 左右兩側的點組成，而中間未凸出部分即被外圈包覆。此情況不會影響建模成果。

1. 結果

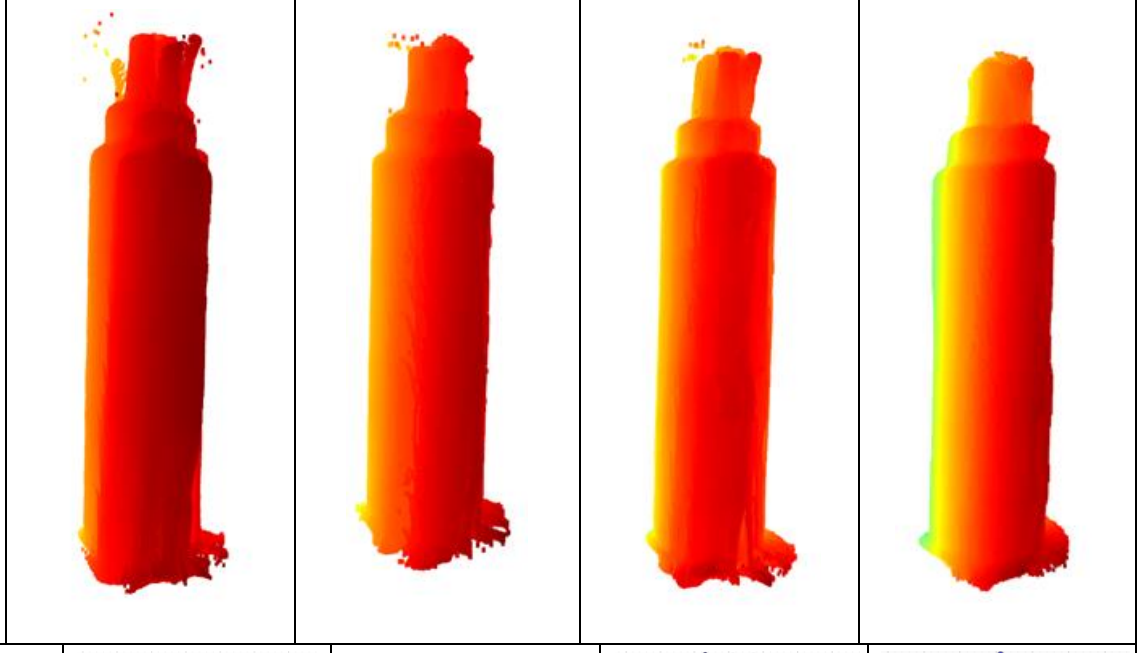
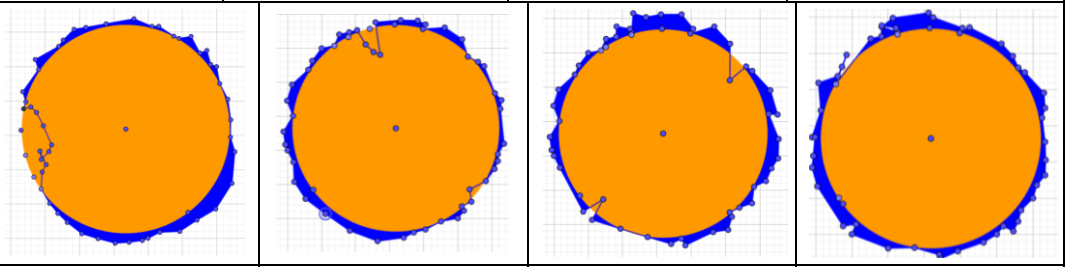
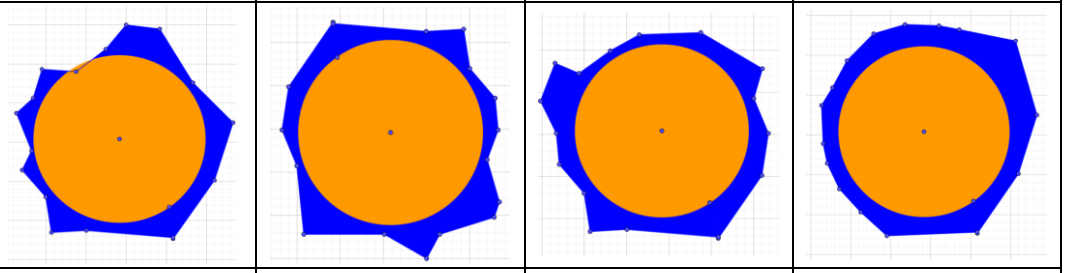
週期 (sec)	15	20	30	60
3d 還原模型				
模型 (藍) 與原物體 (橘) h=$\frac{1}{2}$ 截面邊界對比圖				
$C_{\frac{1}{2}}$	0.973	0.982	0.986	1
$O_{\frac{1}{2}}$	0.125	0.113	0.145	0.132
模型 (藍) 與原物體 (橘) h=1 截面邊界對比圖				
C_1	0.980	1	1	1
O_1	0.425	0.512	0.519	0.563

表２、為四種週期圓柱體還原建模成品與原物體之截面$C_{\frac{1}{2}}$、$O_{\frac{1}{2}}$、C_1、O_1之比較表

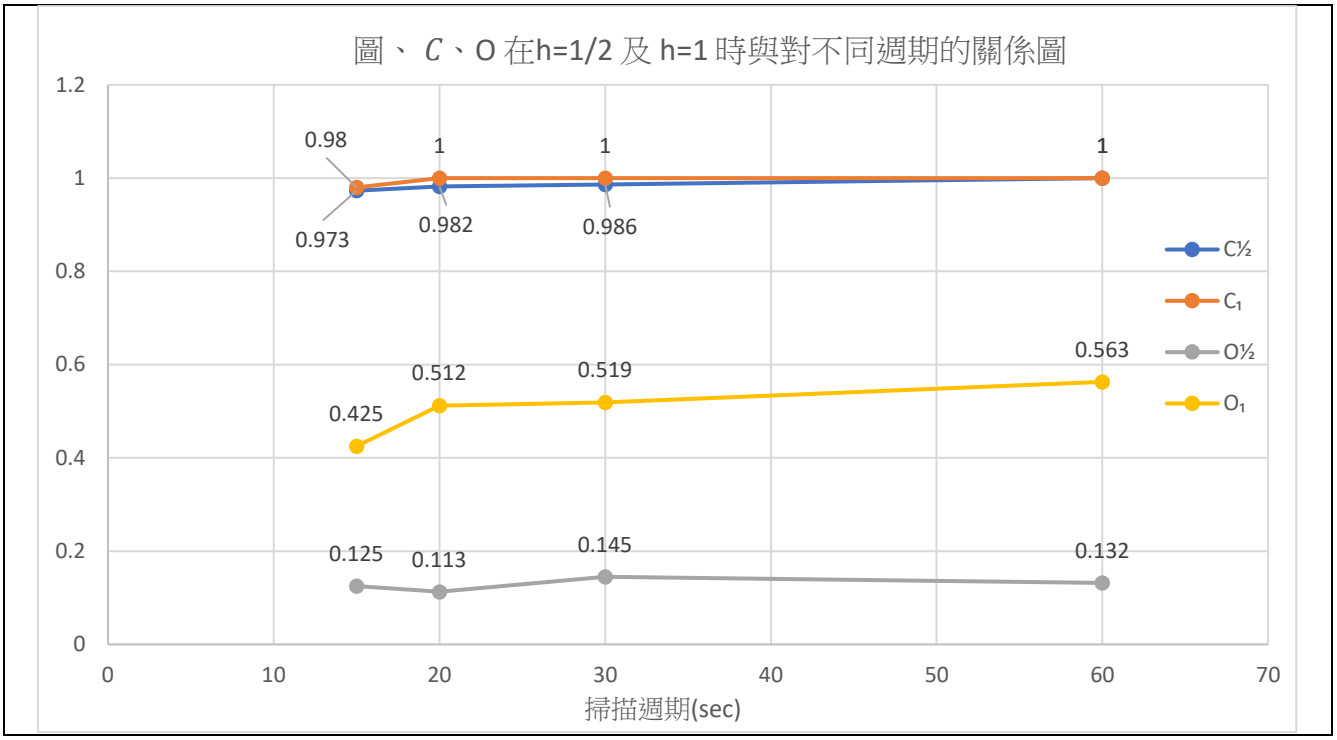


表 2 、為四種週期圓柱體還原建模成品與原物體之截面$C_{\frac{1}{2}}$、$O_{\frac{1}{2}}$ 、 C_1 、 O_1之比較表

2. 討論

(1) 由圖 可知 $C_{\frac{1}{2}}$ 隨週期增長而增加

(2) 由圖可知 $O_{\frac{1}{2}}$ 呈現不規則分布，找不到規律性。可能是測試數據不足所導致。

(3) 由圖可知 C_1 在任何週期皆已非常接近或已達到 1，且 O_1也大於 0.4，可知$A_{r,1}$ 範圍已遠大於$A_{0,1}$。 推測原因為先前提到的深度估計問題，MiDaS 的輸出景深差異不大，讓最頂端細窄處端點之旋轉半徑增加，增加$A_{r,1}$範圍所導致。

(4) 若以圓柱進行建模，根據測得結果可發現不同週期對建模結果無太大影響，推測是因為圓柱的每個方向形狀差異不大，增加數據量並不會有明顯效果。

(二) 正方體還原建模

在正方體的建模過程當中，我們發現到其中一鄰面無法在經過旋轉和平移矩陣調整後回歸正確位置，有面與邊界錯位產生，使整體邊界變的臃腫 (詳見 表 中 3D 還原模型圖) 。推測是 MiDaS 在正方體轉角面對鏡頭時 (圖) 深度預測不佳所導致鄰面夾角呈現不是 90˚（圖），亦有可能是圓面相差校準問題，但此影像經多次伸縮及裁切，難以確定其內部參數矩陣，目前直接忽略。

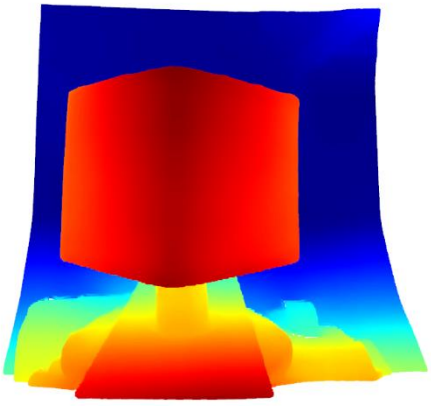


圖 29、轉角正對鏡頭的模型

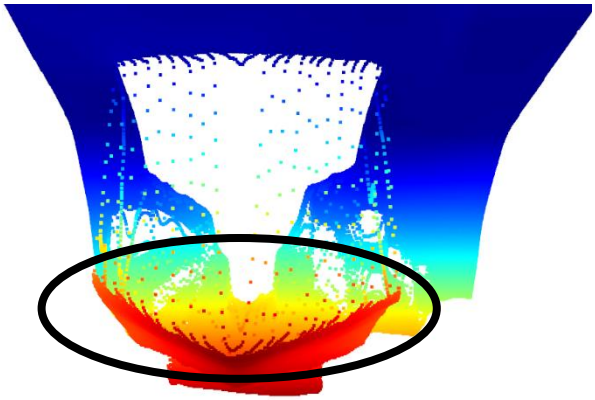
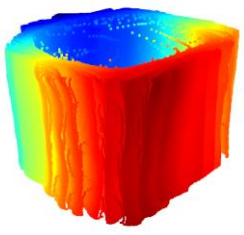
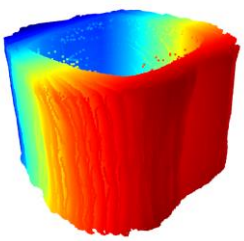
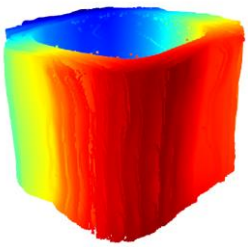
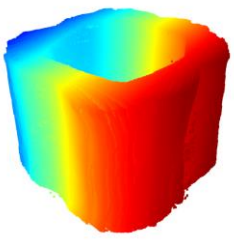
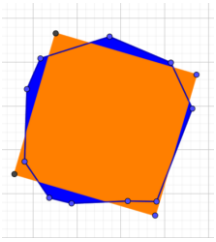
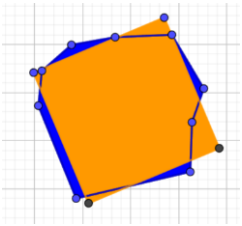
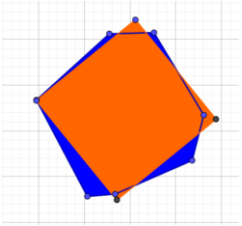
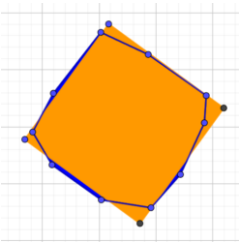


圖 30、鄰面夾角並非 90˚(圈起處)

其中一個鄰面錯位並不會影響底面內部邊界的形狀，於是我們採用底部截面作為建模結果量化依據，以C_h，O_h進行數據比較，判斷在哪種情況底部截面較相似於原物體。正方體還原建模結果如 表 3。僅討論 h=0 之狀況，因建模後物體成類似柱狀結構，對於所有 h 之高度的截面相等。

1. 結果

週期 (sec)	15	20	30	60
3D 還原模型				

模型 (藍) 與原物體 (橘) 底面內圈邊界對比圖				
C_0	0.713	0.855	0.960	0.975
O_0	0.192	0.163	0.180	0.015

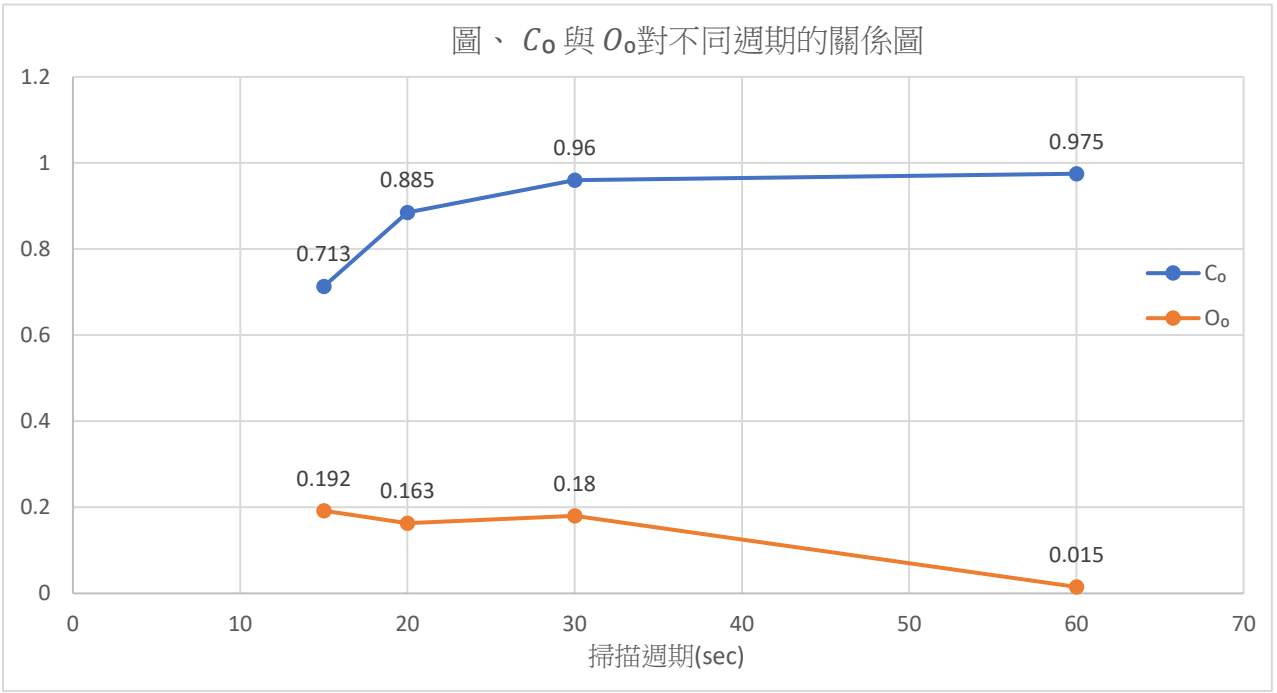


表 3 、為四種週期正方體還原建模成品與原物體之底部截面積C_0、O_0 之比較表

2. 討論

(1) 在圖 C_0 隨週期增長而增大

(2) 在圖 O_0 隨週期增長而減小

(3) 在圖 可見在掃描週期由 15 秒增為 20 秒時， C_0突然增加，推測在週期為 15 秒時，大部分點雲被用來填滿底面，但數據不足無法填滿每一個角度的位置，讓底面與正方形相差甚遠。而週期達 20 秒時，數據飽和，多餘點雲用於修整底面形狀。

(4) 總結，若不討論鄰面夾角，以週期為 30 秒，正方體進行還原建模時可達到不錯的效果。

(三) 透明盒子還原建模

在透明盒子的建模過程當中，我們發現到 MiDaS 無法估計透明物體的深度，在旋轉的過程中，因為折射率的關係，有時鏡頭所拍到的透明盒子不會是透明的，而這時 MiDaS 可以順利進行深度估計(圖)，若拍到的盒子是透明時，MiDaS 會輸出透明盒後面的深度訊息並忽略部分透明盒(圖)。

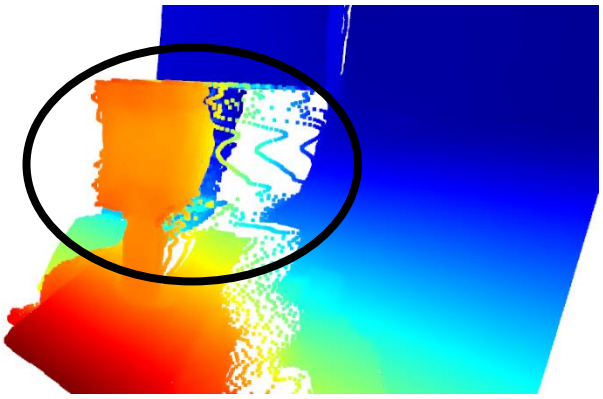


圖 31、非透明表面深度估計成

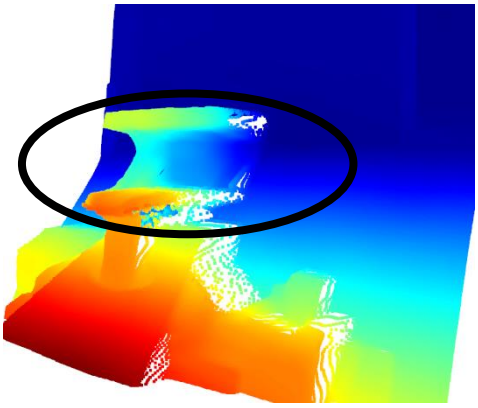
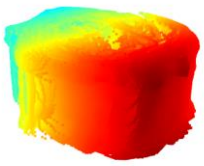
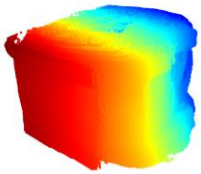
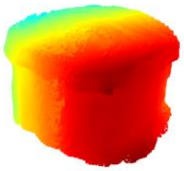
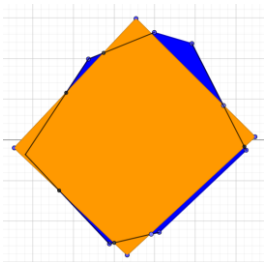
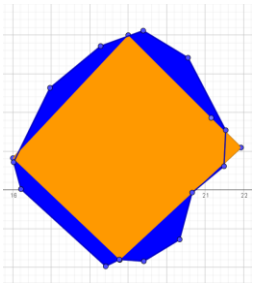
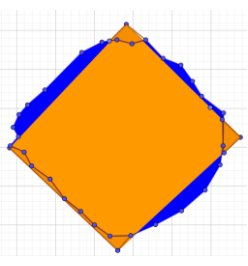


圖 32、透明且深度估計後方，表面

1. 結果

週期 (sec)	15	20	30	60
3D 還原模型	太多透明 無法建模			
模型 (藍) 與原物體 (橘) 底面內圈邊界對比圖				
C_0		0.949	0.988	0.954
O_0		0.064	0.304	0.049

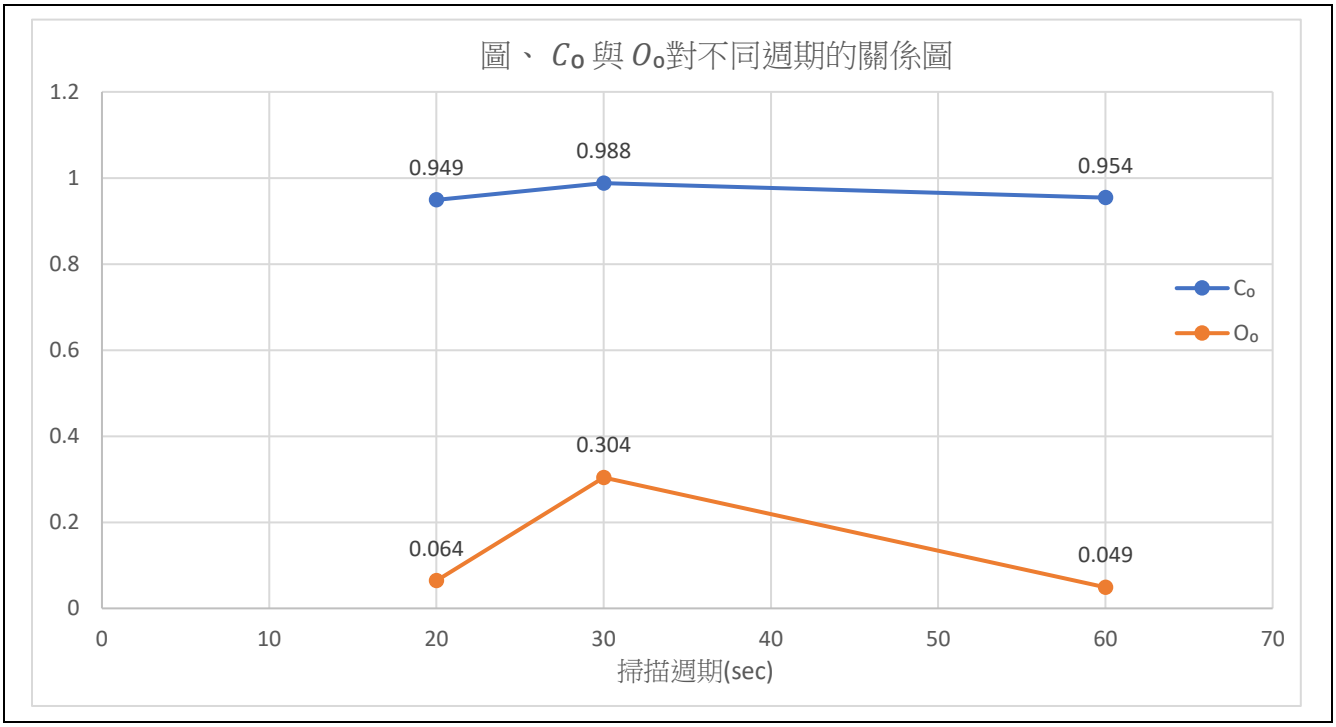


表４、為三種週期正方體還原建模成品原物體之底部截面積C_0、O_0之比較表

2. 討論：

(1) 與表３、表２相比，此物可能因透光度與折射等問題導致掃描結果不理想。

(2) 在本次測試週期為 30 秒的掃描當中，MiDaS 深度估計所估計之透明影像多於其他兩週期時的透明影像，在建模的時形成巨大誤差。

四、總結

(一) 我們以定速軌道裝置進行 3D 還原建模，經 4 種不同週期的測試，發現圓柱體的建模效果在週期為 60 秒時最好，較貼近原物件；立方體的效果次之，但僅針對底面內圈做討論，外圍部分的探討有待我們對相關知識了釐清；而透明盒子的建模效果極差，不只是針對 MiDaS，無法估計透明物件是多數單目深度估計的共同缺點。

(二) 在這幾種測試中，我們發現除了本身深度圖轉換點雲就已經不理想的透明盒子外，其餘兩物件皆在週期最長的時候有最佳表現，因為週期常可以提供更多更完整的數據。

伍、討論

一、無法對透明物件進行有效的建模是因為本研究是基於單目深度估計的系統，無法辨認透明物件是目前電腦視覺的通病，未來可搭配超音波或紅外線協助提供此方面數據。

二、我們推測圓柱的曲率不足和立方體直角未完全還原是因為深度估計模型無法對較近物件進行精準定位，雖然輸入之影像已經過多次伸縮和裁切，我們認為內部參數矩陣可忽略不計，但亦有可能是此原因，目前有待深入確認。

陸、結論

本研究中利用深度預測結合旋轉及平移等變換，完成開發了可藉由影像、位置及角度變化輸出原影像之 3D 模型的系統。雖無法完美還原個物件原始狀態，但截面等基礎參數已經非常貼近原物件之參數，未來將持續針對其他方向進行還原。

希望以後能測試並使用其他深度預測模型來提高精確度。也將在此研究完全完成後開發適合手機運算的程式版本，增加此程式運用的機動性。

柒、參考文獻資料

[1] Rene Ranftl*, Katrin Lasinger*, David Hafner, Konrad Schindler, and Vladlen Koltun (2021)Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer .IEEE

[2] Weisstein EW . (2004) .mathworld wolfram .Affine transformation . https://mathworld.wolfram.com/AffineTransformation.html

[3] https://pandas.pydata.org/docs/user_guide/index.html Python-Pandas user guide

[4] S. Mahdi H. Miangoleh . Dille S . Mai L . Paris S .Aksoy Y(2021) Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging from

[5] Chintala ,S.DEEP LEARNING WITH PYTORCH: A 60 MINUTE BLITZ .https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

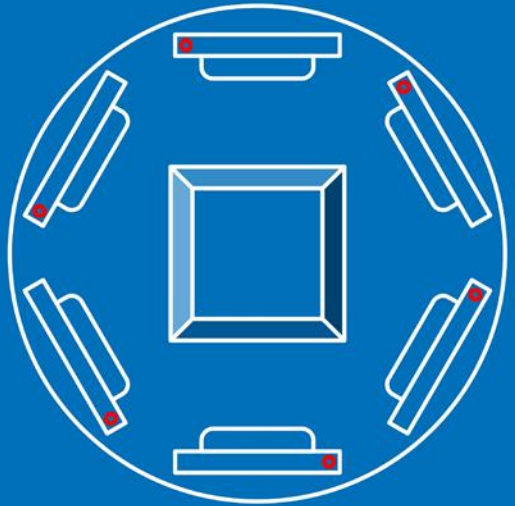
[6] OpenCV Crash Course.

https://opencv.org/opencv-free-course/

【評語】052501

此作品使用手機對某一物件進行 360 度環景拍攝多張相片，再利用 MiDaS 套件對拍攝的每張 2D 影像上的每一個 pixel 估計深度（離攝影機的距離），之後再使用這些相片的深度資訊進行此物件的 3D 建模。此作品針對不同形狀物體如圓柱體、正方體、和透明物體進行此方法的 3D 建模，發現建模結果與真實物體的形狀有一些誤差，實驗中有探討一些因素，例如不同掃描週期（也就是一圈 360 度平分成幾個角度拍攝）和物體形狀和物體顏色對建模誤差的影響。建議未來可以探討使用其它深度估計套件對對實驗結果的影響。

作品海報



把影像提升一個維度
—影片圖片３Ｄ化

摘要

本研究以稠密式深度預測為基礎，進行環繞掃描３Ｄ還原建模。我們以手機作為唯一的測量設備。用手機同時錄影和記錄角度、位移後，同步各數據時間點，並計算每一幀影像的拍攝位置。接著將拍攝的影像輸入深度預測模型轉換成深度圖並將其儲存為點雲，利用旋轉和平移矩陣將點雲轉回正確的位置以進行疊合並完成3D模型。我們開發的還原建模系統以影像、特定時間點的位置和角度來建立３Ｄ模型。此系統無須任何額外標記且硬體設備小巧方便。我們未來會將這套系統集成一個操作簡單手機應用程式，方便使用者在手機上進行相關操作。

壹、前言

一、研究動機

(一)市面上以單一輕便裝置進行3D掃描的儀器稀少
(二)結構光掃描儀需額外標記
(三)旋轉式點對點雷射掃描儀體積大、維護難且價格高
(四)傳統紅外線或超音波等易受環境因素干擾

二、研究目的

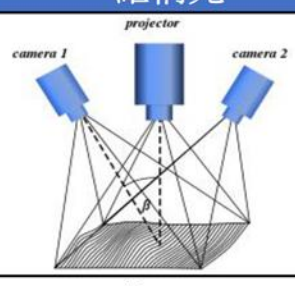
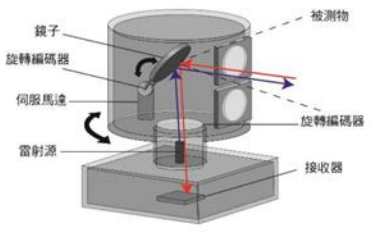
(一)偵測鏡頭旋轉角度、位移並計算位置
(二)製作3D還原建模系統

三、文獻回顧

(一)常見掃描儀比較

本研究希望以不同方式來達成3D還原重建，但市面上已存在多種原理不同之雷射掃描儀。在(表 1)中簡介最常見的兩種技術為結構光與雷射掃描之技術與條件差異。

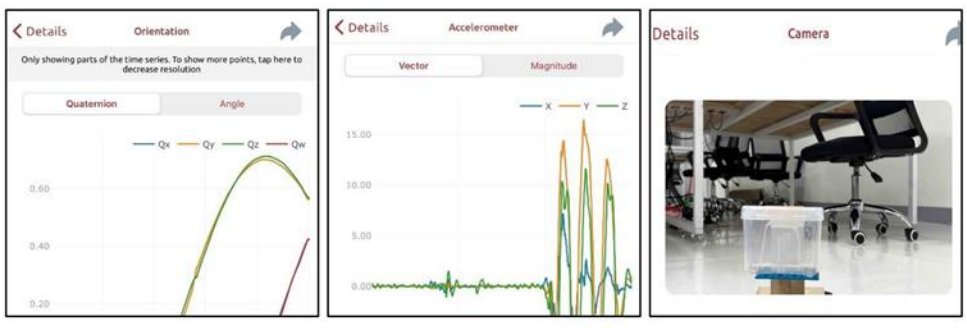
表 1、兩種市面常見雷射掃描儀比較

儀器	結構光	點對點雷射
示意圖		
有效距離	約 1 m	約 250 m
定點擺放	X	O
額外標記	O	X
機器大小	小 (可手持)	大 (需腳架)
價格高低	與手機價格接近	極為昂貴

本研究由位移及角度數據並以影像建立模型，克服結構光掃描時需要額外標記的麻煩；且僅以手機作為偵測工具，與點對點雷射掃描儀相比體積大幅減少，讓系統更加便利。

(二)Sensor Logger 偵測軟體

少數能在偵測物理量變化 (如 : 加速度 (圖1-a) 、三軸角度變化 (圖1-b)) 時同時進行的影像捕捉 (圖1-c) 的軟體，解決了部分手機無法多工攝影的問題。且可以在測量階段結束後將數據輸出為csv檔案，方便電腦讀取。



(a) (b) (c)

圖1、加速度偵測(a)、角度變化偵測(b)、錄影影像(c)

(三) MiDaS深度估計

在[1]中提出，一種用CNN (卷積神經網路) 進行單目深度估計的方法。可輸入RGB照片並輸出一張深度估計圖。

圖2、輸入的RGB影像與輸出的深度圖對比

貳、研究設備及器材

硬體：電腦 , 手機 , NovaPi主控版 , 伺服馬達
平台、軟體及主要插件：Python , Anaconda , OpenCV , Pytorch , TensorFlow , Pandas , Numpy , Matplotlib , Open3D , Sensor Logger

參、研究過程及方法

一、研究流程

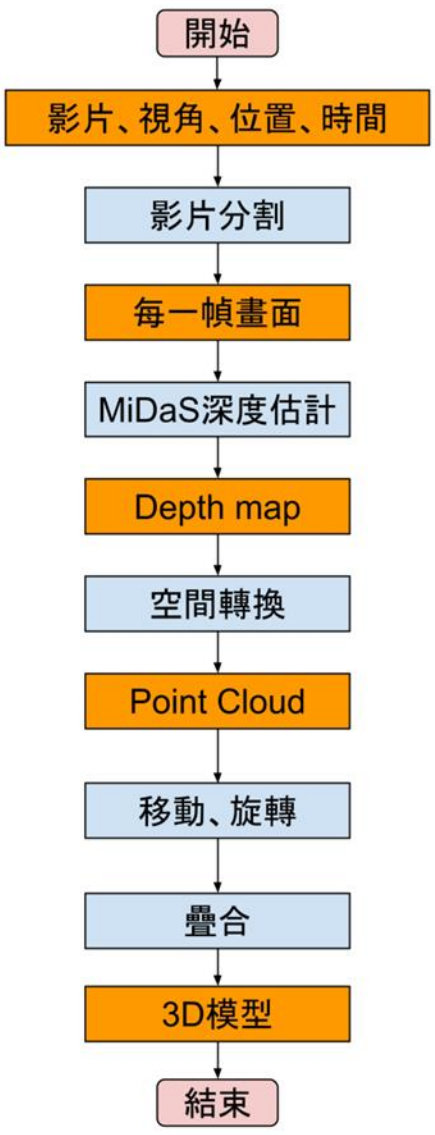


圖3、研究流程圖

二、選定適合的MiDaS模型

研究初期使用 v2.0_large_384 模型，但之後推出 v3.1 版本。因 v3.1 版全面採用transformer架構，使其全域訊息損失率較低，經原作者測試數據發現 v3.1 較 v2.0 版在運行速度和品質皆有提升(圖4)。故最後選用效能及品質較平衡且方便未來手機程式開發的 v3.1-swin2-Large中等模型。

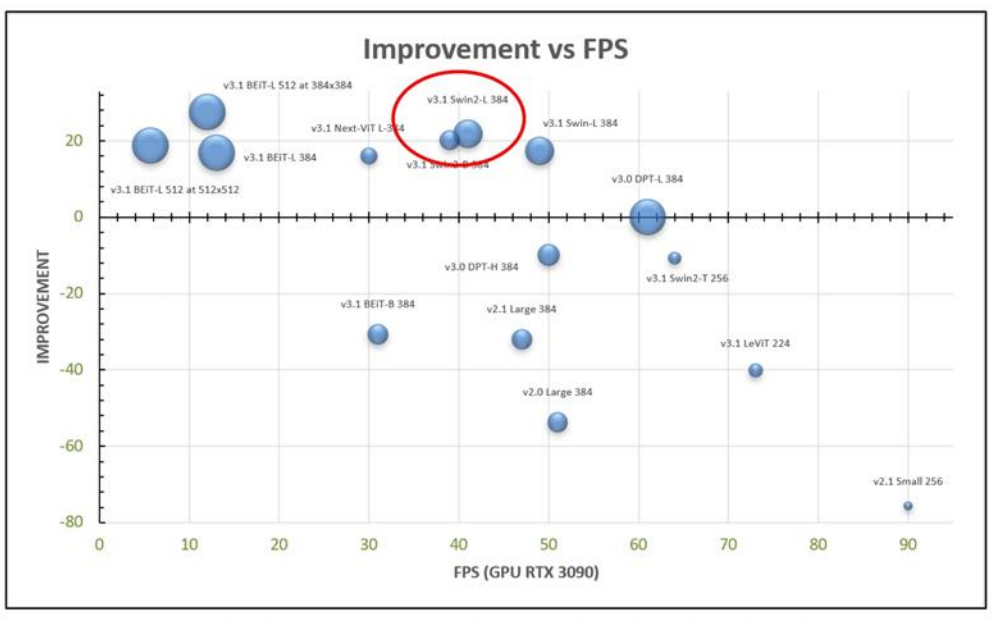


圖4、各模型對比v3.0基準模型效能及輸出FPS關係圖

三、空間中的仿射變換

把在其他角度位置的深度圖平移、旋轉後就可以得到統一基準點的多個深度圖，方便後續點雲進行建模及疊合。

(一)旋轉

在三維空間中的旋轉可以視為依序沿x,y,z方向的旋轉，其旋轉變換以矩陣形式表達為 $R = R_z R_y R_x$，其中 R_x, R_y, R_z 分別為：

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdots\cdots \text{沿 x 軸旋轉}$$

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdots\cdots \text{沿 y 軸旋轉}$$

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdots\cdots \text{沿 z 軸旋轉}$$

(二)平移

將位移矩陣與座標矩陣相乘可得平移後座標矩陣。假設沿 x,y,z 方向分別平移 a,b,c。以矩陣表示如下：

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

四、利用手機的移動數據協助建立3D模型

除了鏡頭輸入外，大多手機可以提供鏡頭相對位移及角度變化等數據。我們可以充分利用此數據來大幅增加3D模型的準確度。

(一)手機鏡頭方向與位移、旋轉方向定義

在手機立直擺正時 +X 朝右、+Y 朝上；+Z垂直螢幕平面 (圖5)；而旋轉方向的定義亦是如此 (圖6)。

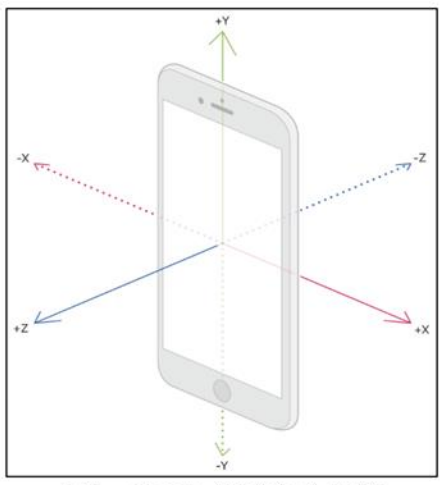


圖5、XYZ加速度方向定義

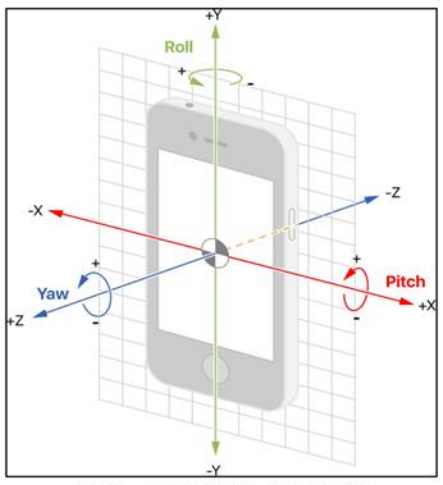


圖6、XYZ旋轉方向定義

(二)影像、位移及角度變化資訊擷取

用Sensor Logger 建立新的自訂偵測並設定偵測頻率為60Hz ，使手機偵測加速度變化及角度變化，並同時開始以2fps錄影。

將取得手機之加速度資料經過兩次梯形法積分後得到每1/2秒所在的位置，計算後將其放在excel中暫存 。

(三)影像、位移及角度變化資訊輸入

暫存在手機的數據需要被處理後輸入至電腦進行運算

1.影像輸入

使用OpenCV套件連接錄好的檔案作為影像輸入，並將影像分割成連續照片以numpy array形式暫存 (圖9)。

```
cap = cv2.VideoCapture('test')
while True:
    ret, frame = cap.read()
    cv2.imshow('frame', frame)
    F=frame
```

圖9、影像輸入片段程式碼(60fps影片輸入)

2.位移及角度變化輸入

將先前錄製之物理變化數據輸出並使用Pandas套件將excel資料輸入至程式中暫存 (圖10)。

```
dfA= pd.read_excel(r'.\testfile.xlsx',sheet_name = 'Linear Accelerometer')
dfG= pd.read_excel(r'.\testfile.xlsx',sheet_name = 'Gyroscope')
for row in range(dfG.shape[0]):
    s_rowA= dfA.iloc[row,:]
    s_rowG= dfG.iloc[row,:]
    if s_rowA[0]!=s_rowG[0]:
        break
    else:
        Time=s_rowA[0]
        xm,ym,zm=s_rowA[1],s_rowA[2],s_rowA[3]
        xr,yr,zr=s_rowG[1],s_rowG[2],s_rowG[3]
        pkg=(Time,xm,ym,zm,xr,yr,zr)
```

圖10、將位移及角度變化資料暫存

(四)取得深度圖與點雲

將RGB影像透過MiDaS可得到一個將景深數據以灰階形式顯示之圖片。在單一幀的情況下，攝影機能抓取到的景深數據只有單一方向。

從 MiDaS的輸出 (圖11) 可以得到像素景深的二維陣列D，每一項表示為$D_{i,j}$ ，經轉換表示為空間中的一點$d_k(i, j, D_{i,j})$ (圖12)。

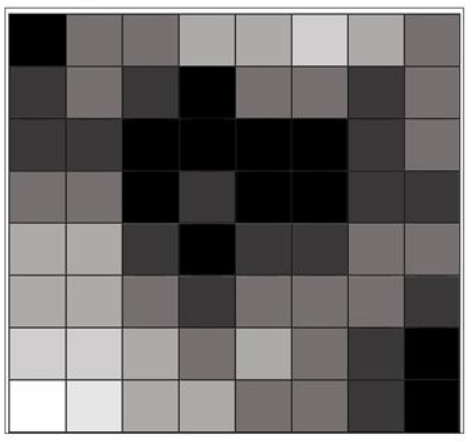


圖11、以逆灰階表示的二維陣列D
(擷取自測試輸出之8*8區塊)

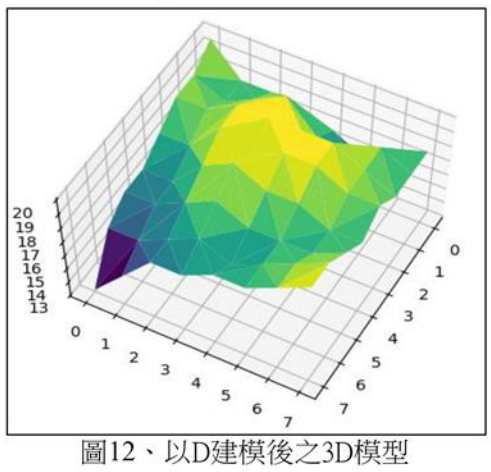


圖12、以D建模後之3D模型

(五) 將深度圖進行仿射變換並疊合

將位移、角度變化和轉換完的點雲進行仿射變換，即得某影像的點雲與相對初始影像點雲在三維空間中正確的位置。疊合所有經變換的點雲可得3D模型 (圖13)。

圖13、多組點雲疊合建模示意圖

肆、研究結果

一、成果評估

(一)一般化高度　$h = \frac{H}{H_0}$ (圖7)

H：高度 ， H_0：模型總高

(二)截面積覆蓋率 $C_h = \frac{A_{r,h} \cap A_{0,h}}{A_{0,h}}$ (圖8)

$A_{r,h}$：模型在 h 的截面積

$A_{0,h}$：實際模型在 h 的截面積

(三)截面積溢出率 $O_h = \frac{A_{r,h} \setminus A_{0,h}}{A_{0,h}}$ (圖8)

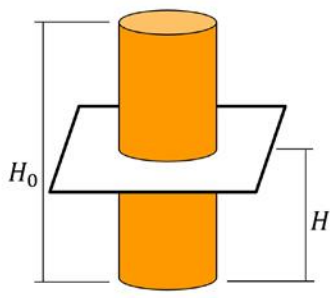
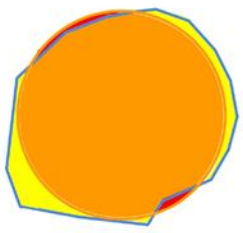


圖7、一般化高度示意圖



$A_{r,h}$：藍色外框 、 $A_{0,h}$：橘色外框

$C_h = \frac{\text{橘色外框}-\text{紅色部分}}{\text{橘色外框}}$

$O_h = \frac{\text{黃色部分}}{\text{橘色外框}}$

圖8、成果評估範例示意圖

二、系統測試步驟

經過測試後發現手機的加速度不準。於是改以定速軌道掃描，直接輸入系統固定速率與位置訊息，避開加速度的誤差，仍保留手機偵測之角度數據輸入。以下是我們的實驗步驟：

(一)選定生活中常見且材質，形狀、材質不同的物體

物體1：圓柱體，物體2：正方體，物體3：透明盒子

物體1

物體2

物體3

(二)製作定速軌道

使用Nova Pi 主機板連接馬達帶動平台定速轉動

(三)將物體和手機放上測試裝置

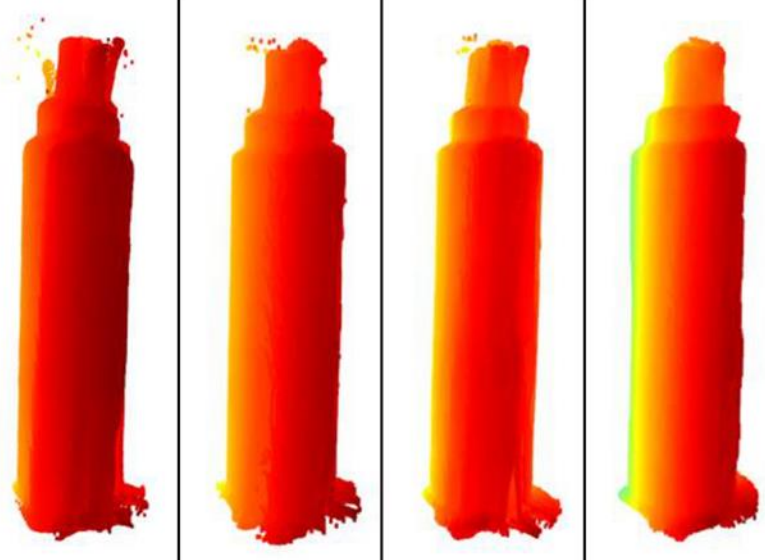
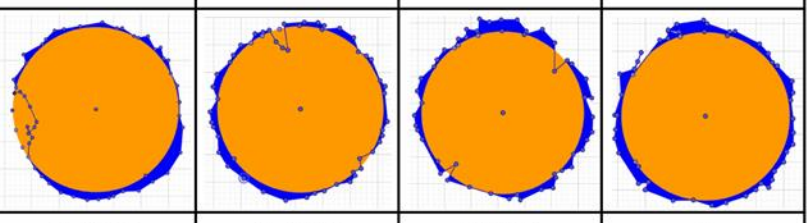
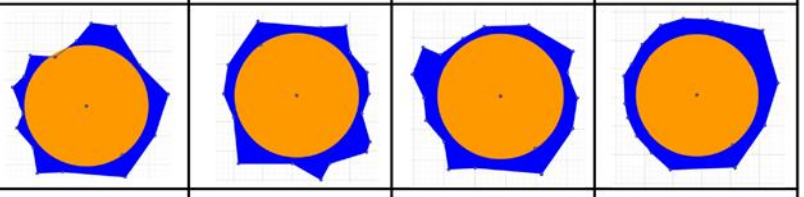
(四)啟動裝置並使用手機開始以2 fps錄製

共四次，以15、20、30、60秒為週期 (30、40、60、120 frames)。

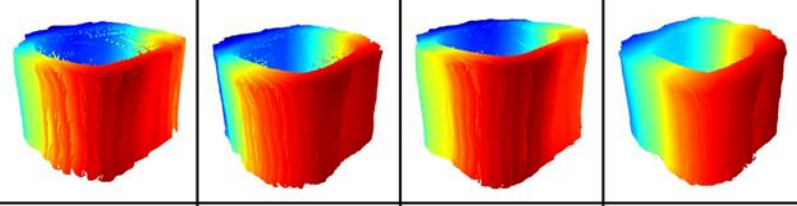
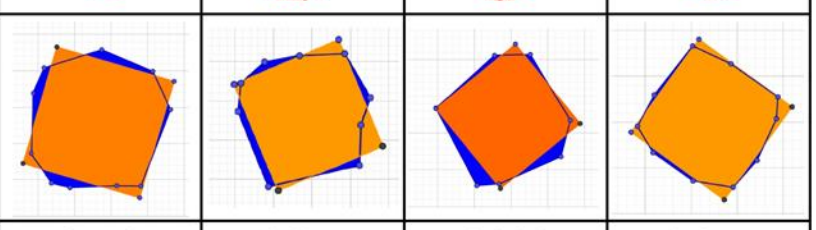
(五)運行系統並建立3D模型

三、手機在特定週期中不同物體的掃描結果

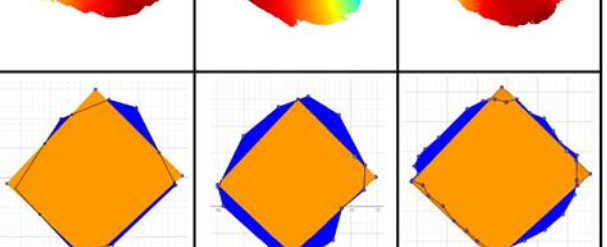
(一)物體 1 (圓柱體)

週期(秒)	15	20	30	60
掃描幀數	30	40	60	120
3D還原模型				
模型(藍)原物(橘) h=½ 的截面積對比				
$C_{h=½}$	0.973	0.982	0.986	1
$O_{h=½}$	0.125	0.113	0.145	0.132
模型(藍)原物(橘) h=1 的截面積對比				
$C_{h=1}$	0.980	1	1	1
$O_{h=1}$	0.425	0.512	0.519	0.563

(二)物體 2 (正方體)

週期(秒)	15	20	30	60
掃描幀數	30	40	60	120
3D還原模型				
模型(藍)原物(橘) h=0 的截面積對比				
$C_{h=0}$	0.713	0.855	0.960	0.975
$O_{h=0}$	0.192	0.163	0.180	0.015

(三)物體 3 (透明盒子)

週期(秒)	15	20	30	60
掃描幀數	30	40	60	120
3D還原模型	太多透明無法建模			
模型(藍)原物(橘) h=0 的截面積對比				
$C_{h=0}$		0.949	0.988	0.954
$O_{h=0}$		0.064	0.304	0.049

伍、問題與討論

一、物體 1、物體 2 掃描結果討論

圖14、物體1 C_h與 O_h 在 h=1及h=½ 對不同掃描幀數的關係圖

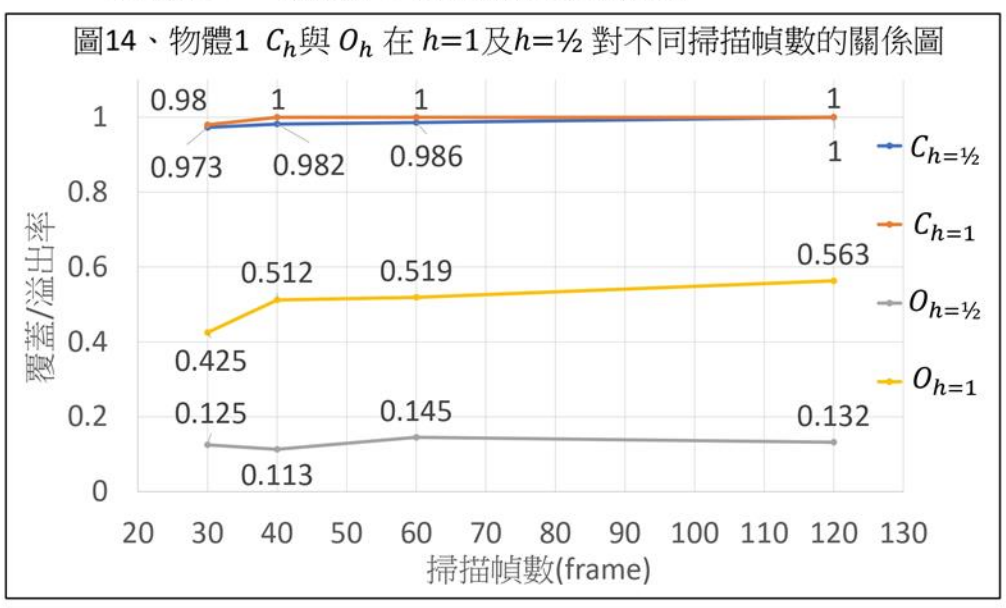
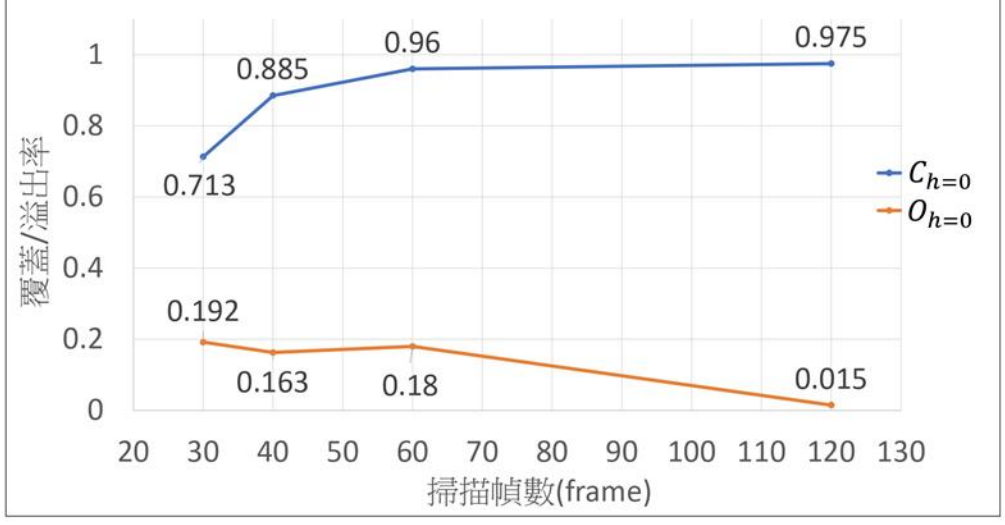


圖15、物體2 $C_{h=0}$ 與 $O_{h=0}$對不同掃描幀數的關係圖



(一) 由圖14、15 發現C_h (覆蓋率) 隨著掃描幀數增加而上升，可知數據量越多C_h 越接近 1 (最大值)。

(二)由圖14 發現$O_{h=1}$ (溢出率) 隨週期增大而明顯上升且$C_{h=1}$ (覆蓋率) 也為1 ，但C_h與O_h在 h=½ 均正常 ，可知其頂部截面已可完全覆蓋並超過原物體的截面，顯示出頂部數據點的深度估計不精準。

(三)由圖15發現在幀數為120時 ，已與實際物體極為接近，在圖14中亦可發現在幀數為120時 h=½ 之結果與實物接近，驗證了以本系統進行還原建模的可行性。

(四) 進一步探討(二)深度預估計不準，我們發現深度圖的深度層次不足，使物體 1 的圓弧表面似平面 (圖16)，而物體 2 的鄰面夾角非90˚ (圖17) ，其邊框變粗就是此受因素影響。

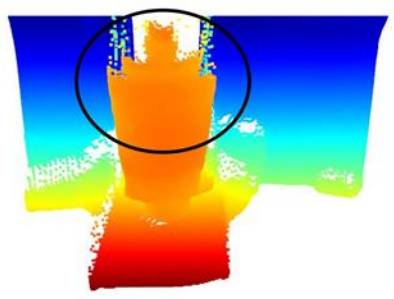


圖16、圓形曲面近似平面(圈起處)

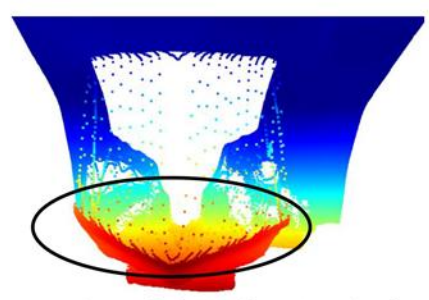
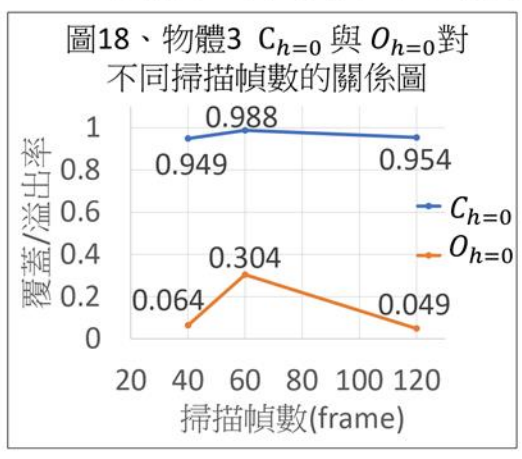


圖17、鄰面夾角並非90˚(圈起處)

推測此問題受兩因素影響：深度估計模型在小範圍精度不高，或本研究忽略的鏡頭內部參數矩陣。而影像經多次縮放、裁切，難計算參數矩陣，尚待後續釐清。

二、物體 3 掃描結果討論

圖18、物體3 $C_{h=0}$ 與 $O_{h=0}$對不同掃描幀數的關係圖



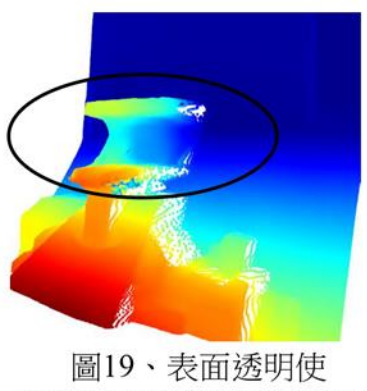


圖19、表面透明使深度估計模型穿透(圈起處)

(一) 由圖18發現$O_{h=0}$ (溢出率) 成不規則分布，形成此現象與深度估計模型無法識別透明物的景深有關 (圖 19)，是目前單目深度估計的通病。

陸、結論

本研究中利用深度預測結合旋轉及平移等變換，開發了可藉由影像、位置及角度變化建立原影像之3D模型的系統。雖無法完美還原個物件原始狀態，但截面等基礎幾何參數已經可以非常貼近原物件之參數，未來將持續針對其他方向進行還原。

希望以後能嘗試計算鏡頭參數矩陣來提高精確度，也將在此研究完成後開發適合手機運算的系統版本，進一步增加此程式運用的機動性。

柒、參考文獻資料

[1] Rene R, Katrin L, David H, Konrad S, and Vladlen K (2021)Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer .IEEE

[2] Weisstein EW . (2004) .mathworld wolfram .Affine transformation . https://mathworld.wolfram.com/AffineTransformation.html

[3] https://pandas.pydata.org/docs/user_guide/index.html Python-Pandas user guide

[4] S. Mahdi H. Miangoleh . Dille S . Mai L . Paris S .Aksoy Y(2021) Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging from