

B06505004 莊博翰 專題報告

主題:改上學期所開發的 3d 點雲疊合對齊程式

main :test.py

讀取bag檔

- function :rw.readallframe

移除背景

- function :rbg.remove_bg

獲取畫面中綠色/紅色部分

- function :imt.green_mask()
- function :imt.red_mask()

找出綠版與紅點

- function :gprd.find_green_plane_red_dot()

獲取平均過後的depth image

- function :pc.non_zero_mean()

找出綠版法向量，綠版x軸向量，y軸向量

- pc.findvec()

產生轉換用座標 test.py end

將整個豬隻點雲的座標係轉換 testpig.py start

- pc.transform()

儲存點雲

- rw.savepointtofile ()

package: read_write as rw

function:

readallframes(path,start,maxframe,redepth,recolor,timestamp)

path	string	bag 檔路徑
start	int	由第幾 frame 開始讀取
maxframe	int	最多讀出多少 frames
redepth	list	會將 depth image 存入此 list : depth image per frame:2d numpy int array (720,1280) 單位是 mm , redepth→ [frame1, frame2..... frameN] 30fps
recolor	list	會將 colorimage 存入此 list 因該更 depth image 一一對應 color image per frame:3d numpy uint8 array (720,1280,3) 單位是 rgb , recolor→ [frame1, frame2..... frameN] 30fps
timestamp	list	會將時間軸存入此 list 對應每個 frame 的時間 list of int 單位是 ms
return	none	

savetopointcloud(file_name,depth_frame,color_frame,re=False)

將 depth image+color image 轉換成 pointcloud 然後儲存(optional)

file_name	string	存檔檔名
depth_frame	2d numpy int array	深度影像(如果 depth =0 的點會自動忽視)
color_frame	3d numpy uint array (720,1280,3)	彩色影像
re	bool	是否 return if(!re)存檔
return	none or 2d list	[[x(float),y,z,r(uint8),g,b].....] each element is a point

savepointtofile (p,file_name)

p	list of 1d numpy array	所有要儲存的點
file_name	string	儲存的檔名

package: img_tool as img

green_mask(cframe) red_mask(cframe)

cframe	3d numpy uint array	彩色影像
return	2d bool numpy array	顯示那些部分是紅色/綠色(自訂義)

package greenplane_redot as gprd

find_green_plane_red_dot(gpfilter,rdfilter,p,debug=True)

gpfilter	2d numpy bool array	畫面屬於綠色部分 1 ,other 0
rdfilter	2d numpy bool array	畫面屬於紅色部分 1 ,other 0
p	3d numpy uint array	原始圖片
debug	bool	是否顯示結果
return	(gplane,redDot)	gplane 2d numpy bool array 屬於綠版部分 redDot [四個紅點畫面[i,j]值]→[[i,j],[i,j], [i,j],[i,j]]

package pointcloud as pc

findvec(gpmask,reddot,depth,picture,debug=False)

gpmask	2d numpy bool array	畫面屬於綠版部分
reddot	list of list (4,2)	[四個紅點畫面[i,j]值]
depth	2d numpy int array	深度影像
picture	3d numpy uint array	原始圖片
debug	bool	是否產生 debug pointcloud
return	list of numpy 1d float array	新座標的 vx, vy ,vz,中心點

transform(allpoints,vx,vy,vz,mid,debug=True)

allpoints	list of 1d numpy array	所有的點[[x,y,z,b,g,r]...]
vx vy vz mid	1d numpy array	要將現在的所有輸入點雲，轉換成以 vx vy vz mid 為 x y z 軸&原點的新點雲
return	list of 1d numpy array	新點雲

程式進入點:test.py 須備齊兩 bag file: 背景 (pathbg)& 綠板(pathgb)

```
16 filename= "23c2"
17 camera=filename[-2:]
18 debug=True
19 red=[]
20 rec=[]
21 tfr=[]
22 pathbg="/Users/percychien/Desktop/project/pig-main/test/"
23 pathgb="/Users/percychien/Desktop/project/pig-main/test/board/before"
```

執行完後會產生用來對其這一視角的 vectors (filename_vec.npy)

還有背景深度圖 (filename_bg.npy)

然後前往 testpig.py 放入這一視角拍攝豬隻的檔案

```
16 filename="23c2"
17 debug=False
18 camera=filename[-2:]
19 red=[]
20 rec=[]
21 tfr=
22 path="/Users/percychien/Desktop/project/pig-main/test/"
```

執行完後會產生轉換過座標的 pointcloud(<filename>pig<id of camera>.ply)

之後依序執行其他五個視角，最終因該會有六個同一座標軸的 pointcloud
(來自六個攝影機)