

DETERMINACIÓN DE PROXIMIDAD MEDIANTE EL ANÁLISIS ESTADÍSTICO DE IMÁGENES DE PROFUNDIDAD

VERANO DE LA CIENCIA 2023

DIVISIÓN DE INGENIERÍAS
CAMPUS IRAPUATO
SALAMANCA (DICIS)

Autores

Juan Carlos Rodríguez Ceballos

José Manuel Silva Alanís

Carlos Abraham Robles Saucedo

Encargado Del Proyecto

Dr. José Luis Contreras Hernández

Descripción general

En este proyecto se desarrollará un programa para el análisis estadístico de imágenes de profundidad obtenidas mediante una cámara Intel en ambientes de interior y exterior. Se adquirirán imágenes en los distintos entornos presentando obstáculos típicos de cada uno. El programa desarrollado permitirá determinar la presencia de obstáculos a partir de una distancia establecida por el usuario procesando digitalmente las imágenes fuera de línea.

Justificación del equipo

El desarrollo de un algoritmo que determine la presencia de obstáculos mediante análisis estadístico de imágenes en ambientes de interior y exterior aportará para el desarrollo futuro de proyectos enfocados en mejorar la adaptación en distintos medios de personas con problemas de visión.

Objetivos

Crear un manual de conexión de cámara, lectura de imágenes y procesamiento del programa creado con el fin de facilitar su uso en proyectos de investigación futuros.

Resultados esperados

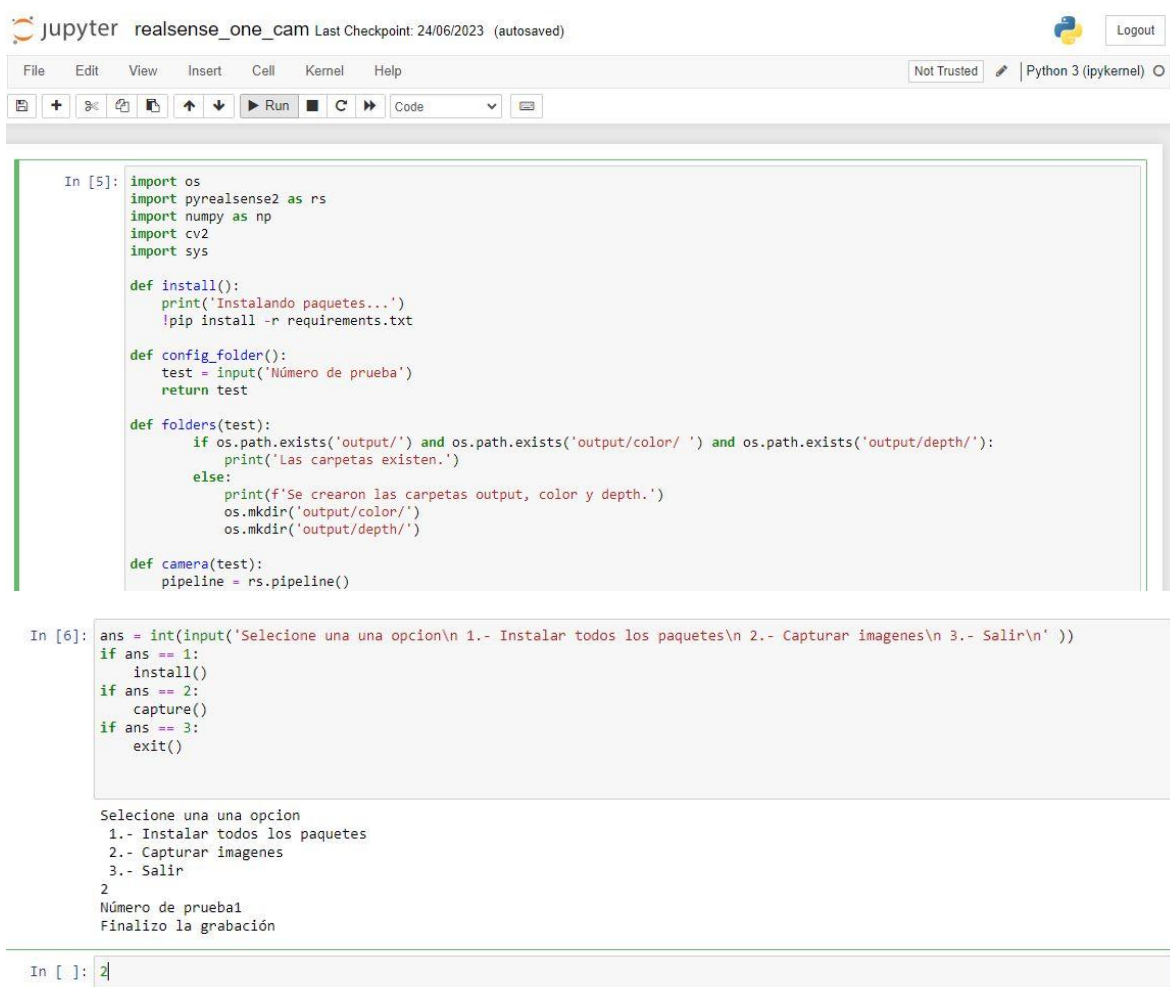
Redacción de un manual tutorial con la descripción de conexiones de cámara, lectura de imágenes y procesamiento del programa creado.

Requisitos

- Jupyter.
- Matlab R2020a o superior.
- Cámara Intel Realsense Depth Camera D415.

Instructivo

1. Se inicia Jupyter desde CMD con el comando *jupyter notebook* y se corre el archivo “realsense_one_cam.ipynb”.
2. Se corre el programa por partes con el numero de prueba y se selecciona la opción de “capturar imágenes”



The screenshot shows a Jupyter Notebook window titled "realsense_one_cam" with a last checkpoint of "24/06/2023 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help), a toolbar with icons for file operations and execution, and a status bar indicating "Not Trusted" and "Python 3 (ipykernel)".

```
In [5]: import os
import pyrealsense2 as rs
import numpy as np
import cv2
import sys

def install():
    print('Instalando paquetes...')
    !pip install -r requirements.txt

def config_folder():
    test = input('Número de prueba')
    return test

def folders(test):
    if os.path.exists('output/') and os.path.exists('output/color/ ') and os.path.exists('output/depth/'):
        print('Las carpetas existen.')
    else:
        print(f'Se crearon las carpetas output, color y depth.')
        os.mkdir('output/color/')
        os.mkdir('output/depth/')

def camera(test):
    pipeline = rs.pipeline()
```

```
In [6]: ans = int(input('Seleccione una una opcion\n 1.- Instalar todos los paquetes\n 2.- Capturar imagenes\n 3.- Salir\n' ))
if ans == 1:
    install()
if ans == 2:
    capture()
if ans == 3:
    exit()
```

Seleccione una una opcion
1.- Instalar todos los paquetes
2.- Capturar imagenes
3.- Salir
2
Número de prueba1
Finalizo la grabación

In []: 2

Ilustración 1. Programa de adquisición de fotos.

Al seleccionar esta opción se inicia una ventana con la foto que está tomando junto con la imagen de profundidad.



Ilustración 2. Imagen original y de profundidad.

Estas fotos se almacenan en la carpeta “Output” en las carpetas “color” y “depth” respectivamente.

3. Se ejecuta el programa “processImg.m” en Matlab. Es necesario tener dicho programa junto al programa “varianzaMat.m” en la misma carpeta en donde se tienen las imágenes obtenidas.

Para importar las imágenes para ejecutar el programa se cambian los comandos

```
ori=imread('prueba_220_color.png');  
imshow(ori); %Foto original a color  
img=imread('prueba_220_depth.png'); %foto de profundidad
```

con los nombres y formato de archivo respectivos. Con estos ajustes se corre el programa.

4. Al ejecutar el programa se abren dos ventanas mostrando la imagen original junto a la imagen de profundidad negativo como se muestra a continuación:



Ilustración 3. Imagen original y de profundidad negativa.

5. El programa invierte la imagen de profundidad y la divide en 9 segmentos de las mismas dimensiones. La imagen es de resolución 480x640.

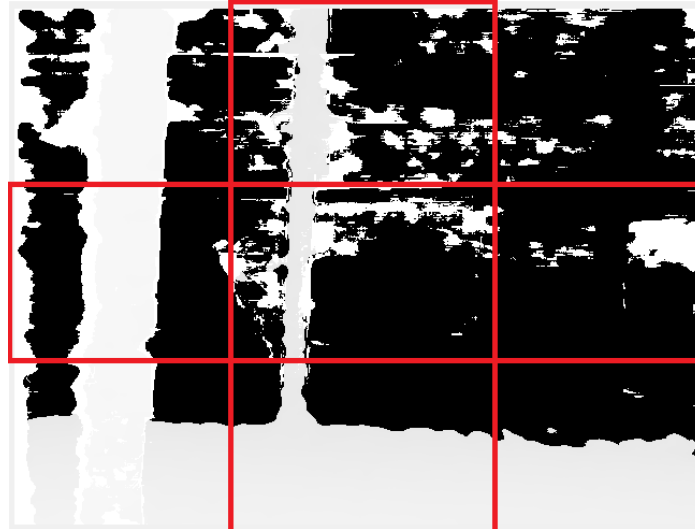


Ilustración 4. Segmentos de la imagen de profundidad.

6. El programa lee cada una de las secciones de la imagen y calcula el promedio, la varianza y la desviación estándar de la densidad de píxeles blancos. Dichos valores finalmente se vuelven a seleccionar los segmentos con mayor promedio, varianza y desviación respectivamente y coloca el valor probabilístico del objeto y su ubicación dentro de la matriz.

```
Command Window

Ubicacion por promedio
1.0e+04 *

    3.3916     0     0
    2.6861     0     0
    3.3879     0     0

Ubicacion por varianza
1.0e+04 *

    3.4320     0     0
    2.7433     0     0
    3.4947     0     0

Ubicacion por desviacion
185.2572 165.6287 186.9413
     0     0     0
     0     0     0
```

Ilustración 5. Ubicación del objeto por cálculos probabilísticos.

Ejemplos



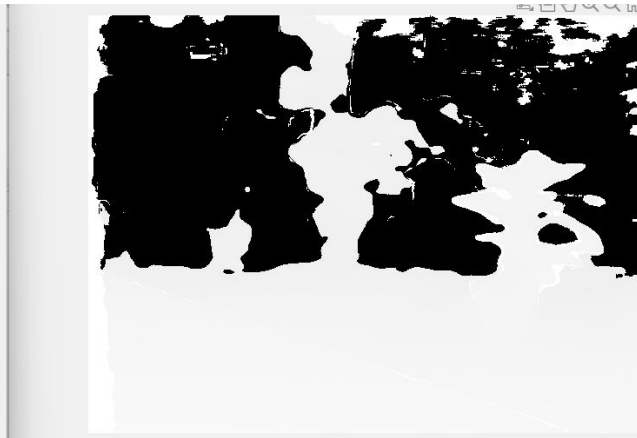
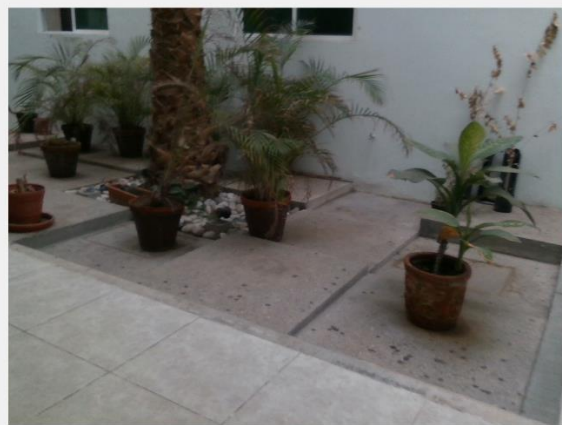
Ubicacion por promedio
1.0e+04 *

6.3806	0	0
6.2972	0	0
6.1379	0	0

Ubicacion por varianza
1.0e+04 *

6.5128	0	0
6.4751	0	0
6.3240	0	0

Ubicacion por desviacion
255.2018 254.4631 251.4753
0 0 0
0 0 0



Ubicacion por promedio
1.0e+04 *

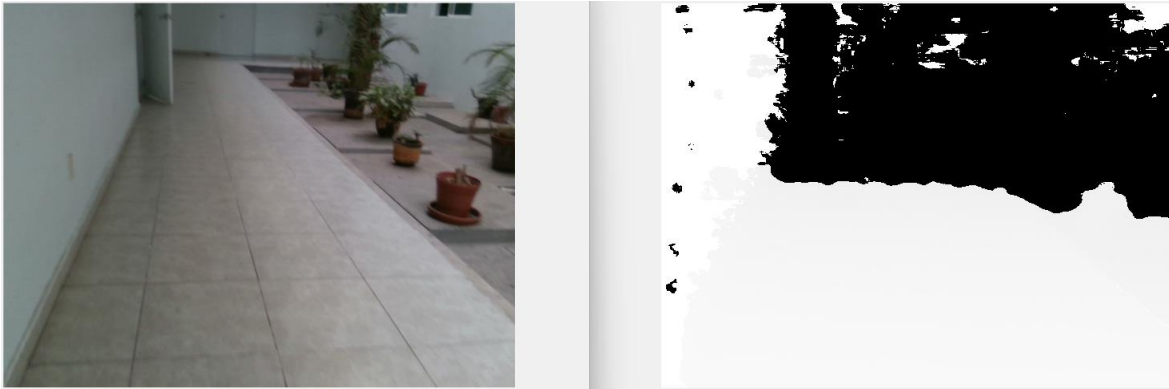
0	2.8196	0
0	2.9641	3.4359
6.3548	6.3092	6.2899

Ubicacion por varianza
1.0e+04 *

0	2.9548	0
0	3.1517	3.6224
6.5535	6.5843	6.6150

Ubicacion por desviacion

0	0	255.9980
171.8964	177.5295	256.5983
0	190.3250	257.1971



Ubicacion por promedio

1.0e+04 *

4.4153	0	0
5.4399	3.6429	2.7068
6.3600	6.3455	6.3324

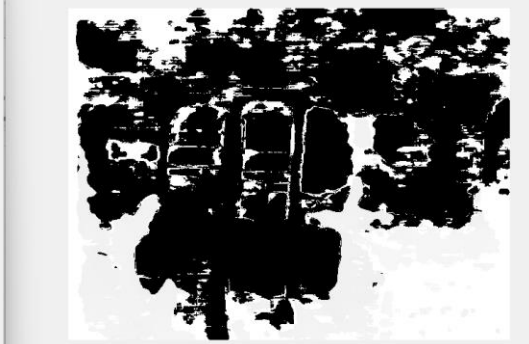
Ubicacion por varianza

1.0e+04 *

4.3980	0	0
5.5731	3.8500	2.8820
6.5283	6.5843	6.6150

Ubicacion por desviacion

209.7152	236.0750	255.5049
0	196.2132	256.5983
0	169.7653	257.1971



Ubicacion por promedio

1.0e+04 *

2.7195	0	0
0	0	0
4.2024	2.6114	6.4488

Ubicacion por varianza

1.0e+04 *

2.7026	0	0
0	0	0
4.3573	2.7158	6.6040

Ubicacion por desviacion

164.3953	0	208.7425
0	0	164.7958
0	0	256.9815

Ubicacion por promedio

1.0e+04 *

0	0	0
0	0	0
3.4322	3.4030	3.4879

Ubicacion por varianza

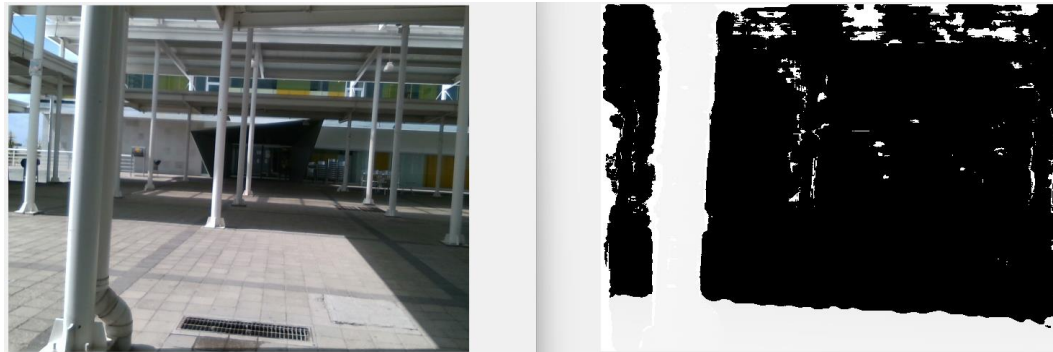
1.0e+04 *

0	0	0
0	0	0
3.5799	3.5954	3.7005

Ubicacion por desviacion

0	0	189.2072
0	0	189.6158
0	0	192.3674





Ubicacion por promedio
1.0e+04 *

2.7391	0	0
2.8303	0	0
4.0964	2.2802	0

Ubicacion por varianza
1.0e+04 *

2.7943	0	0
2.9042	0	0
4.2589	0	0

Ubicacion por desviacion

167.1619	170.4170	206.3716
0	0	155.5370
0	0	0



Ubicacion por promedio
1.0e+04 *

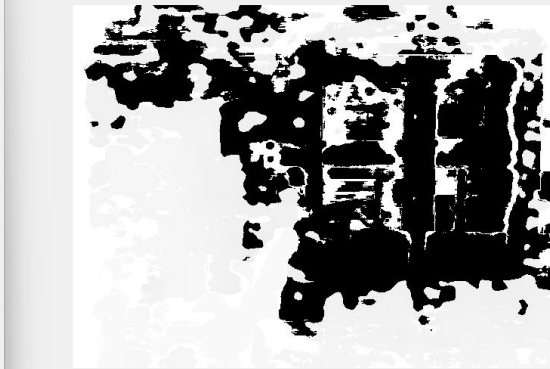
0	0	0
0	0	0
5.0971	3.7801	3.7938

Ubicacion por varianza
1.0e+04 *

0	0	0
0	0	0
5.2439	3.9293	3.9254

Ubicacion por desviacion

0	0	228.9950
0	0	198.2239
0	0	198.1275



Ubicacion por promedio

1.0e+04 *

3.5872	3.0905	3.5112
6.2013	2.2677	0
6.4262	4.2410	4.2145

Ubicacion por varianza

1.0e+04 *

3.6923	3.1662	3.5945
6.4052	0	0
6.5535	4.3944	4.4519

Ubicacion por desviacion

192.1537	253.0850	255.9980
177.9381	152.8162	209.6286
189.5906	0	210.9962



Ubicacion por promedio
1.0e+04 *

0	0	0
2.6717	0	2.3034
5.1360	3.7040	5.6476

Ubicacion por varianza
1.0e+04 *

0	0	0
2.7156	0	0
5.2915	3.8903	5.8470

Ubicacion por desviacion

0	164.7900	230.0316
0	0	197.2381
0	153.7948	241.8055

Ubicacion por promedio

1.0e+04 *

3.3916	0	0
2.6861	0	0
3.3879	0	0

Ubicacion por varianza

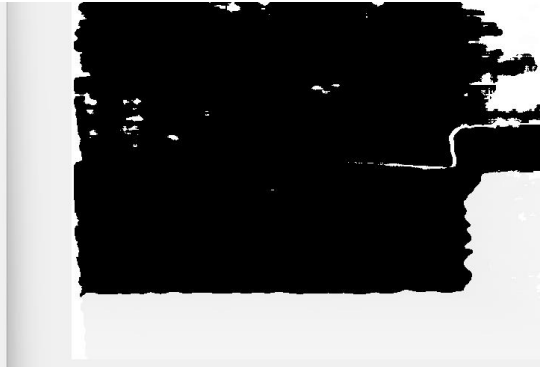
1.0e+04 *

3.4320	0	0
2.7433	0	0
3.4947	0	0

Ubicacion por desviacion

185.2572	165.6287	186.9413
0	0	0
0	0	0





Ubicacion por promedio
1.0e+04 *

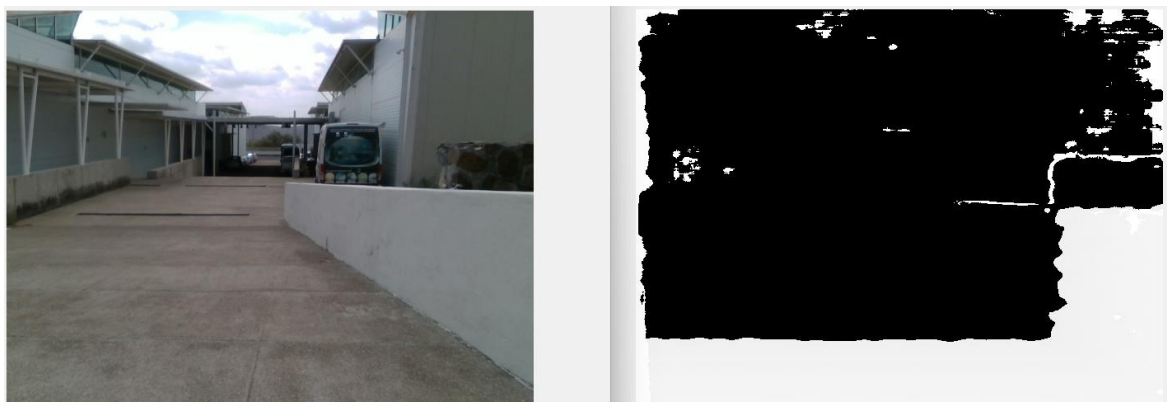
0	0	2.1578
0	0	0
3.5569	3.4148	4.8936

Ubicacion por varianza
1.0e+04 *

0	0	0
0	0	0
3.7212	3.6092	5.1906

Ubicacion por desviacion

0	0	192.9031
0	0	189.9783
0	0	227.8278



Ubicacion por promedio
1.0e+04 *

0	0	0
0	0	2.0688
3.2790	3.0581	4.9754

Ubicacion por varianza
1.0e+04 *

0	0	0
0	0	0
3.4305	3.2342	5.2779

Ubicacion por desviacion

0	0	185.2159
0	0	179.8396
0	0	229.7365

Ubicacion por promedio
1.0e+04 *

6.4220	6.4025	6.3384
6.4851	6.4638	6.4720
6.4236	6.4005	6.3959

Ubicacion por varianza
1.0e+04 *

3.5574	0	3.8023
4.9005	2.9601	3.3355
3.1152	0	0

Ubicacion por desviacion

188.6103	221.3698	176.4985
0	172.0489	0
194.9958	182.6323	0

