



ESCUELA DE INGENIERÍA EN COMPUTADORES

CE-5302: PROYECTO DE DISEÑO INGENIERÍA EN COMPUTADORES

---

## Manual de Usuario

Generación de nubes de puntos a partir de *stacking* sin información de pose de la cámara

---

### AUTOR

José Julián Camacho Hernández  
2019201459

### ASESOR

Luis Alberto Chavarría Zamora

29 de mayo de 2024

## Tabla de contenidos

<b>1</b>	<b>Requisitos del Sistema</b>	<b>1</b>
<b>2</b>	<b>Instalación y Configuración</b>	<b>1</b>
2.1	Clonar el Repositorio . . . . .	1
2.2	Configuración del Entorno Virtual . . . . .	1
2.3	Instalación de Dependencias . . . . .	1
<b>3</b>	<b>Configuración del Sistema</b>	<b>2</b>
3.1	Estructura del Archivo de Configuración . . . . .	2
3.2	Descripción Detallada de los Parámetros . . . . .	2
3.2.1	input_path . . . . .	2
3.2.2	config_params . . . . .	2
3.2.2.1	voxel_size . . . . .	2
3.2.2.2	remove_outliers_params . . . . .	3
3.2.2.3	combinability_threshold . . . . .	3
3.2.3	output_file . . . . .	3
3.3	Uso del Archivo de Configuración . . . . .	4
3.4	Consideraciones Adicionales . . . . .	4
<b>4</b>	<b>Uso de la Aplicación</b>	<b>4</b>
4.1	Ejecución Básica . . . . .	4
4.2	Carga de Nubes de Puntos . . . . .	5
4.3	Preprocesamiento de Nubes de Puntos . . . . .	5
4.4	Combinación de Nubes de Puntos . . . . .	6
<b>5</b>	<b>Conclusiones</b>	<b>7</b>
<b>6</b>	<b>Licencia</b>	<b>8</b>
<b>7</b>	<b>Contacto</b>	<b>8</b>

El presente manual está destinado a proporcionar una guía exhaustiva para el uso del *Point Cloud Generator*. Este *software*, desarrollado como parte del curso CE-5302: Proyecto de Diseño en Ingeniería en Computadores, se enfoca en la generación de mapas tridimensionales a partir de nubes de puntos individuales, sin información de la pose de la cámara. A través de técnicas de visión por computadora y fotogrametría, el proyecto tiene como objetivo principal la construcción de representaciones tridimensionales a partir de datos de entrada de nubes de puntos.

## 1. Requisitos del Sistema

- Se requiere un sistema operativo compatible con Python 3.8 o superior.
- Acceso a internet para la instalación de dependencias.
- Acceso a archivos de nubes de puntos para procesar.

## 2. Instalación y Configuración

### 2.1. Clonar el Repositorio

Para obtener el código fuente del Point Cloud Generator, se debe clonar el repositorio desde GitHub utilizando el siguiente comando en la terminal:

```
git clone https://github.com/JulianCamacho/point-cloud-generator
cd point-cloud-generator
```

### 2.2. Configuración del Entorno Virtual

Se recomienda la creación de un entorno virtual para aislar las dependencias del proyecto. Esto puede lograrse mediante el siguiente comando:

```
python -m venv env
source env/bin/activate # En Windows: 'env\Scripts\activate'
```

### 2.3. Instalación de Dependencias

Una vez activado el entorno virtual, las dependencias necesarias se pueden instalar ejecutando el siguiente comando:

```
pip install -r requirements.txt
```

## 3. Configuración del Sistema

El archivo de configuración es un componente crucial del **Point Cloud Generator**, ya que permite personalizar y controlar los diversos parámetros del proceso de carga, preprocesamiento y combinación de nubes de puntos.

### 3.1. Estructura del Archivo de Configuración

El archivo de configuración está escrito en formato JSON y debe contener la siguiente estructura:

```
1 {
2   "input_path": "../data/cloud-points",
3   "config_params": {
4     "voxel_size": 0.02,
5     "remove_outliers_params": {
6       "nb_neighbors": 20,
7       "std_ratio": 2.0
8     },
9     "combinability_threshold": 0.5
10  },
11  "output_file": "nube_pcd_combinada.pcd"
12 }
```

### 3.2. Descripción Detallada de los Parámetros

#### 3.2.1. input\_path

**Descripción:** La ruta al directorio donde se encuentran los archivos de nubes de puntos que se van a procesar.

**Ejemplo:**

```
"input_path": "../data/cloud-points"
```

#### 3.2.2. config\_params

Este objeto contiene varios parámetros de configuración que controlan aspectos específicos del procesamiento de las nubes de puntos.

##### 3.2.2.1. voxel\_size

**Descripción:** Define el tamaño del voxel para el muestreo de las nubes de puntos. El muestreo por voxel es una técnica utilizada para reducir la cantidad de puntos en una nube de puntos, lo que puede mejorar la eficiencia del procesamiento.

**Ejemplo:**

```
"voxel_size": 0.02
```

### 3.2.2.2. `remove_outliers_params`

Este objeto contiene los parámetros utilizados para la eliminación de valores atípicos (outliers) en las nubes de puntos. La eliminación de outliers es importante para mejorar la calidad de la nube de puntos final.

- **`nb_neighbors`**: Número de vecinos a considerar para cada punto al evaluar si es un outlier. Un valor más alto considera más puntos vecinos y puede resultar en una eliminación más estricta de outliers.

**Ejemplo:**

```
"nb_neighbors":
```

```
\begin{verbatim}  
"nb_neighbors": 20
```

- **`std_ratio`**: Ratio estándar utilizado para definir el umbral de eliminación de outliers. Este valor determina cuán lejos de la media deben estar los puntos para ser considerados outliers. Un valor más bajo es más estricto y eliminará más puntos que se consideran atípicos.

**Ejemplo:**

```
"std_ratio": 2.0
```

### 3.2.2.3. `combinability_threshold`

**Descripción:** Umbral para determinar la combinabilidad de las nubes de puntos. Este parámetro define la sensibilidad del algoritmo para decidir si dos nubes de puntos pueden combinarse eficazmente. Un umbral más bajo puede resultar en una combinación más estricta, mientras que un umbral más alto permite una mayor flexibilidad.

**Ejemplo:**

```
"combinability_threshold": 0.5
```

### 3.2.3. `output_file`

**Descripción:** Nombre del archivo `.pcd` donde se guardará la nube de puntos combinada. Este es el resultado final del procesamiento y debe especificar la ruta y el nombre del archivo de salida.

**Ejemplo:**

```
"output_file": "nube_pcd_combinada.pcd"
```

### 3.3. Uso del Archivo de Configuración

Para utilizar el archivo de configuración, asegúrese de que esté ubicado en el directorio adecuado y que su ruta esté correctamente especificada en el código fuente. El archivo se carga y se lee al inicio del programa para configurar los parámetros necesarios para el procesamiento de las nubes de puntos.

Asegúrese de adaptar la ruta del archivo de configuración (`config_path`) según la estructura de su proyecto.

### 3.4. Consideraciones Adicionales

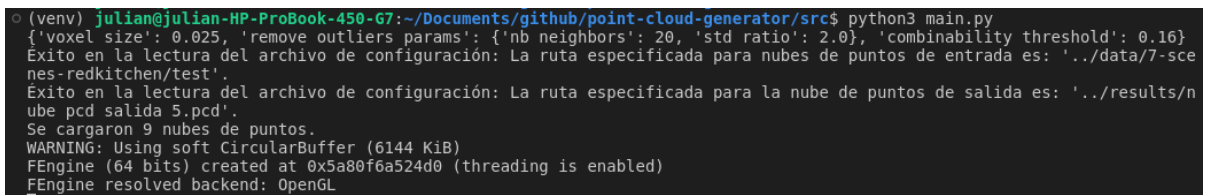
- Verifique que el archivo JSON esté correctamente formateado para evitar errores de lectura.
- Realice pruebas con diferentes configuraciones para encontrar los valores óptimos de los parámetros según sus necesidades específicas.
- Documente cualquier cambio en los parámetros del archivo de configuración para mantener la trazabilidad y facilitar futuras modificaciones.

## 4. Uso de la Aplicación

### 4.1. Ejecución Básica

Para ejecutar la aplicación, simplemente se debe ejecutar el script `main.py` desde la terminal:

```
cd src
python3 main.py
```



```
o (venv) julian@julian-HP-ProBook-450-G7:~/Documents/github/point-cloud-generator/src$ python3 main.py
{'voxel size': 0.025, 'remove outliers params': {'nb neighbors': 20, 'std ratio': 2.0}, 'combinability threshold': 0.16}
Éxito en la lectura del archivo de configuración: La ruta especificada para nubes de puntos de entrada es: '../data/7-sce-
nes-redkitchen/test'.
Éxito en la lectura del archivo de configuración: La ruta especificada para la nube de puntos de salida es: '../results/n
ube pcd salida 5.pcd'.
Se cargaron 9 nubes de puntos.
WARNING: Using soft CircularBuffer (6144 KiB)
FEngine (64 bits) created at 0x5a80f6a524d0 (threading is enabled)
FEngine resolved backend: OpenGL
```

Figura 1: Ejecución del programa.

```
data > {} config.json > ...
1 {
2   "input_path": "../data/7-scenes-redkitchen/test",
3   "config_params": {
4     "voxel_size": 0.05,
5     "remove_outliers_params": {
6       "nb_neighbors": 20,
7       "std_ratio": 2.0
8     },
9     "combinability_threshold": 0.17
10  },
11  "output_file": "../results/nube_pcd_salida_5.pcd"
12 }
```

Figura 2: Ejemplo de archivo de configuración.

## 4.2. Carga de Nubes de Puntos

La aplicación permite cargar nubes de puntos desde archivos locales. Asegúrese de tener los archivos de nubes de puntos en el directorio especificado en el archivo de configuración.

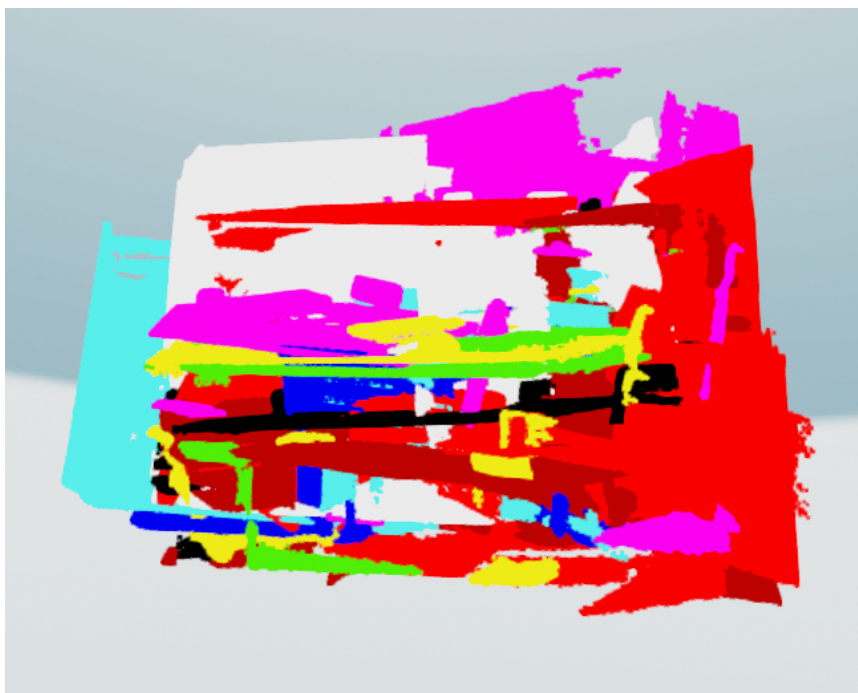


Figura 3: Carga de nubes de puntos individuales.

## 4.3. Preprocesamiento de Nubes de Puntos

Es posible aplicar diferentes técnicas de preprocesamiento, como filtrado y eliminación de valores atípicos, a las nubes de puntos cargadas. Estos parámetros se especifican en el archivo de configuración.

```
0}, 'combinability threshold': 0.16}
Éxito al validar los contenidos del archivo de configuración, iniciando el procesamiento
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Remove statistical outliers: [=====] 100%
Se preprocesaron 9 nubes de puntos.
```

Figura 4: Ejecución del preprocesamiento.

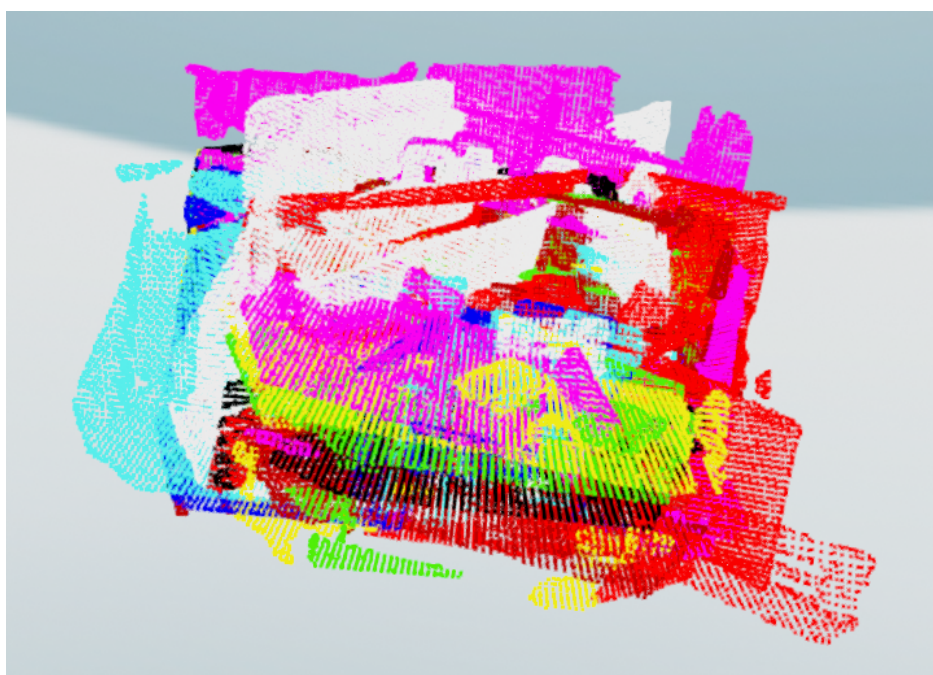


Figura 5: Nubes de puntos preprocesadas.

#### 4.4. Combinación de Nubes de Puntos

La aplicación utiliza métodos de registro y optimización para combinar múltiples nubes de puntos en un único mapa tridimensional. El resultado final se guarda en un archivo .pcd.



```
Aplicando ICP punto a plano (registro fino)...
Construyendo el grafo de poses
Calculando normales para la nube de puntos objetivo...
Aplicando ICP punto a plano (registro grueso)...
Aplicando ICP punto a plano (registro fino)...
Construyendo el grafo de poses
Calculando normales para la nube de puntos objetivo...
Aplicando ICP punto a plano (registro grueso)...
Aplicando ICP punto a plano (registro fino)...
Construyendo el grafo de poses
Optimizando PoseGraph ...
[Open3D DEBUG] Validating PoseGraph - finished.
[Open3D DEBUG] [GlobalOptimizationLM] Optimizing PoseGraph having 5 nodes and 10 edges.
[Open3D DEBUG] Line process weight : 12.884203
[Open3D DEBUG] [Initial ] residual : 6.234978e+04, lambda : 1.466285e+00
[Open3D DEBUG] [Iteration 00] residual : 6.497258e+01, valid edges : 1, time : 0.000 sec.
[Open3D DEBUG] [Iteration 01] residual : 6.482135e+01, valid edges : 1, time : 0.000 sec.
[Open3D DEBUG] [Iteration 02] residual : 6.482096e+01, valid edges : 1, time : 0.000 sec.
[Open3D DEBUG] Delta.norm() < 1.000000e-06 * (x.norm() + 1.000000e-06)
[Open3D DEBUG] [GlobalOptimizationLM] total time : 0.001 sec.
[Open3D DEBUG] [GlobalOptimizationLM] Optimizing PoseGraph having 5 nodes and 5 edges.
[Open3D DEBUG] Line process weight : 14.710500
[Open3D DEBUG] [Initial ] residual : 6.151126e-01, lambda : 1.466193e+00
[Open3D DEBUG] [Iteration 00] residual : 6.150011e-01, valid edges : 1, time : 0.000 sec.
[Open3D DEBUG] Delta.norm() < 1.000000e-06 * (x.norm() + 1.000000e-06)
[Open3D DEBUG] [GlobalOptimizationLM] total time : 0.001 sec.
[Open3D DEBUG] CompensateReferencePoseGraphNode : reference : 0
Exito al aplicar el algoritmo, guardando archivo de salida
```

Figura 6: Ejecución del algoritmo de combinación.

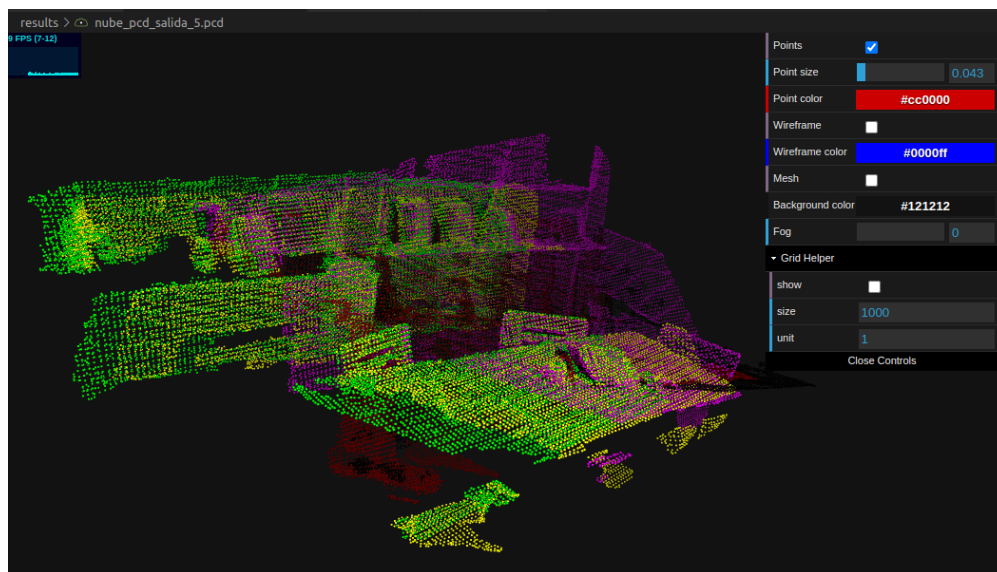


Figura 7: Nube de puntos resultante.

## 5. Conclusiones

En este manual, se han cubierto los aspectos fundamentales de la instalación, configuración y uso del Point Cloud Generator. Se espera que esta guía haya proporcionado la información necesaria para comenzar a utilizar la aplicación y explorar sus diversas funcionalidades para generar y procesar nubes de puntos.

## 6. Licencia

Este proyecto está licenciado como código abierto bajo la Licencia MIT. Consulte el archivo LICENSE en el repositorio para más detalles.

## 7. Contacto

José Julián Camacho Hernández  
jjulian.341@gmail.com  
Repositorio del proyecto: point-cloud-generator