

PCL Nedir

1. PCL Nedir

PCL, nokta bulutu üzerinde fitreleme, yüzey yapılandırma, model çıkartımı, segmentasyon işlemlerini gerçekleyebilen modern algoritmaları içermektedir.

1.1. Çok kullanılan işlemler

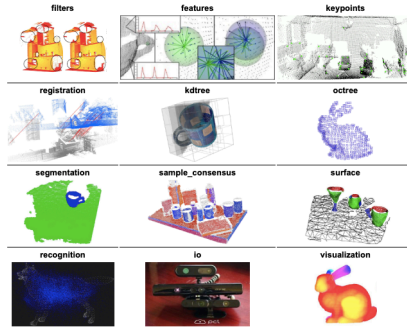


Figure 1. Sık kullanılan işlemler


- Filtreleme
- Görselleştirme
- Segmentasyon

Buradan Diğer İşlemleri Görebilirsiniz¹



Görselleştirme işlemlerinde harici olarak VTK kütüphanesi kullanılmaktadır.

Table 1. İşlem Detayları

İşlem	İşlev
Filtreleme 	Genel olarak nokta bulutu verisinde bulunan gürültü verileri veya istenmeyen verilerin filtrelenmesi için kullanılmaktadır

¹ <http://www.pointclouds.org/documentation/>

İşlem	İşlev
Görselleştirme 	Nokta bulutunun farklı renk veya farklı boyutlarda görselleştirmede kullanılmaktadır. VTK kütüphanesi gerekmektedir
Segmentasyon 	Basitçe nokta bulutu verilerinin içerisindeki objelerin birbirinden ayırt edilecek şekilde renklendirilmesi. Segmentasyonun farklı kullanım şekilleride bulunmaktadır

2. PCL'in Ubuntu 16.04'e Kurulumu



Bu işlemlerde **Cmake**'in kurulu olduğu varsayılmaktadır. Eğer kurulu değil ise [Cmake Download²](https://cmake.org/download/) bu linkten kurulumunu yapınız.



PCL, güncel olarak tüm Ubuntu sürümlerinde PPA üzerinden desteklemektedir. Terminale aşağıdaki komutları girdikten sonra PCL kullanıma hazır olacaktır

Aşağıdaki komutları sırasıyla konsola giriniz.

1. `sudo add-apt-repository ppa:v-launchpad-jochen-sprickerhof-de/pal`
2. `sudo apt-get update`
3. `sudo apt-get install libpcl-all`



Eğer kod (2) sırasında 404 hatası alınıyor ise farklı bir yöntem olarak terminale aşağıdaki satırlar yazılabilir.

```
sudo apt install libpcl-dev
```

Ya da

² <https://cmake.org/download/>

```
sudo apt-get install libpcl-dev
```

Yazılarak PCL, Ubuntu 16.04'e indirilmiş olacaktır.

3. PCL'in Kullanımı

PCL indirildikten sonra aşağıdaki adımları izleyiniz.

1. `mkdir pcd_write_test`
2. `cd pcd_write_test`
3. `nano CMakeLists.txt`

Terminalde açılan ekrana aşağıdaki komutları yapıştırarak CTRL+X' e basın 2 kere onaylayın.

```
cmake_minimum_required(VERSION 2.8 FATAL_ERROR)
project(pcd_write)
find_package(PCL 1.2 REQUIRED)
include_directories(${PCL_INCLUDE_DIRS})
link_directories(${PCL_LIBRARY_DIRS})
add_definitions(${PCL_DEFINITIONS})
add_executable (pcd_write pcd_write.cpp)
target_link_libraries (pcd_write ${PCL_LIBRARIES})
```

İşlem tamamlandıktan sonra CMake dosyası oluşturulmuş olur. Bu işlemeden sonra C++ kodunu oluşturmak için aşağıdaki işlemler yapılmalıdır.

```
nano pcd_write.cpp
```

Terminalde açılan ekrana aşağıdaki komutları yapıştırarak CTRL+X' e basın 2 kere onaylayın.

```
#include <iostream> #include <pcl/io/pcd_io.h> #include <pcl/point_types.h>
int main(int argc, char** argv){ pcl::PointCloud<pcl::PointXYZ> cloud; // Fill
in the cloud data cloud.width = 5; cloud.height = 1; cloud.is_dense = false;
cloud.points.resize (cloud.width * cloud.height); for (std::size_t i = 0; i <
cloud.points.size (); ++i){ cloud.points[i].x = 1024 * rand () / (RAND_MAX + 1.0f);
cloud.points[i].y = 1024 * rand () / (RAND_MAX + 1.0f); cloud.points[i].z = 1024
* rand () / (RAND_MAX + 1.0f);
```

```

    }
    pcl::io::savePCDFileASCII ("test_pcd.pcd", cloud);
    std::cerr << "Saved " << cloud.points.size () << " data points to
    test_pcd.pcd." << std::endl;
    for (std::size_t i = 0; i < cloud.points.size (); ++i)
        std::cerr << "      " << cloud.points[i].x << " " << cloud.points[i].y
    << " " << cloud.points[i].z << std::endl;
    return (0);
}

```

PCL ile point cloud'u PCD olarak kayıt etme.

```

#include <iostream>
#include <pcl/io/pcd_io.h>
#include <pcl/point_types.h>
int main(int argc, char** argv){
    pcl::PointCloud<pcl::PointXYZ> cloud;
    // Fill in the cloud data
    cloud.width      = 5;
    cloud.height     = 1;
    cloud.is_dense   = false;
    cloud.points.resize (cloud.width * cloud.height);
    for (std::size_t i = 0; i < cloud.points.size (); ++i){
        cloud.points[i].x = 1024 * rand () / (RAND_MAX + 1.0f);
        cloud.points[i].y = 1024 * rand () / (RAND_MAX + 1.0f);
        cloud.points[i].z = 1024 * rand () / (RAND_MAX + 1.0f);

    }
    pcl::io::savePCDFileASCII ("test_pcd.pcd", cloud);
    std::cerr << "Saved " << cloud.points.size () << " data points to
    test_pcd.pcd." << std::endl;
    for (std::size_t i = 0; i < cloud.points.size (); ++i)
        std::cerr << "      " << cloud.points[i].x << " " << cloud.points[i].y
    << " " << cloud.points[i].z << std::endl;
    return (0);
}

```



Kurulumu yapmadıysanız [Section 2, “PCL’in Ubuntu 16.04’e Kurulumu”](#)

Bu işlem de gerçekleştirdikten sonra CMake ve C++ kodu oluşturulmuş olur. Bu iki dosyanın derlenmesi için aşağıdaki satırlar terminale yazılmalıdır.

```
1. mkdir build
2. cd build
3. cmake ..
4. make
5. ./pcd_write_test
```

Böylece PCL kütüphanesinde faydalanılarak herhangi bir cismin nokta bulut görüntüsünü kaydetmiş oluruz.



PCL'in indirilmesi ve PCL'in kullanımı aşağıdaki videoda anlatılmaktadır.

3.1. UML Diagramı

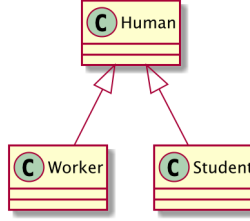


Figure 2. Örnek UML



Verilen Diagram Sadece Örnektir

