



**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ  
MÜHENDİSLİK MİMARLIK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**Nokta Bulutu İşleme PROJE RAPORU- 2**

152120171076 AHMAD ZAIDAN

152120181108 ENES KAŞIKCI

152120181109 MURAT MUSA YALÇIN

152120171073 BOBURSHOH ABDULOV

152120171097 KUMBUKANI KAMANGA

OCAK 2020

## 1. Giriş

Bu projede iki farklı dosyadan nokta verileri alındıktan sonra bunları filtreden geçirip dönüştürülerek görselleştirmeye hazır hale getirmek amaçlanmaktadır.

## 2. Tasarım

### 2.1 Tasarım

**Point Sınıfı:** Bu sınıf nokta bulutundaki 3B noktaların koordinatlarını tutar. Eşit operatörü ( $=$ ), iki noktanın eşit olup olmadığını denetler.

**PointCloud Sınıfı:** Sahip olduğu noktaları, dinamik olarak yaratılan bir Point listesinde tutar.  $+$  operatörü, her iki nokta bulutunun sahip olduğu noktalara sahip tek bir nokta bulutunu döndürür.  $=$  operatörü, bir nokta bulutunun başka bir nokta bulutuna kopyalanmasını sağlar.

**Transform Sınıfı:** Açıklamalar kısmında anlatıldığı gibi, iki koordinat eksenini orijinleri arasındaki uzaklık (trans) ve rotasyon açılarını (angles) Eigen kütüphanesi ile  $3 \times 1$ ,  $3 \times 1$  matris boyutlarında vector'e alır. Dönüşüm matrisini Eigen kütüphanesi ile  $4 \times 1$ ,  $4 \times 1$  matris boyutlarında oluşturur (transMatrix). Daha sonrasında doTrans fonksiyonu ile alınan nokta ya da nokta bulutunu bu dönüşüme tabi tutarak dönüştürülmüş nokta ya da nokta bulutunu döndürür.

**DepthCamera Sınıfı:** İsmi (fileName) verilen dosyadan, capture fonksiyonu çağrıldığında noktaları okur ve yaratılan nokta bulutu nesnesine bu noktaları atar. Nokta bulutunu döndürür. Bu işlem kameranın bir benzetimidir. Noktalar bir kamera yerine dosyadan alınır.

**PointCloudGenerator Sınıfı:** DepthCamera sınıfını, kendi abstract sınıfı altında toplar. capture ve captureFor üye fonksiyonları pure virtual fonksiyon olacaktır. capture fonksiyonu DepthCamera sınıfındaki gibi çalışacaktır. captureFor fonksiyonu ise, capture fonksiyonu ile öncelikle aynı işlemleri yapacak; ancak, nokta bulutu, üyesi olan filterpipe dan geçirdikten ve transform nesnesi ile dönüştürüldükten sonra döndürecektir.

**FilterPipe Sınıfı:** Birden fazla filtreden geçmesi gereken nokta bulutunun filtreleme işlemini yapacak sınıftır. Bu nesneye, farklı tip filtreler ya da farklı parametrelere sahip aynı tip filtreler addFilter fonksiyonu ile eklenecektir. Daha sonra filterOut fonksiyonuna verilen nokta bulutu, ekleme sırasında göre birer birer filtrelerden geçirilip, sonuç nokta bulutunu döndürecektir.

**PointCloudInterface Sınıfı:** Bu sınıf işlemleri basitleştirmek için kullanılan bir sınıftır. İki üye fonksiyona sahiptir. generate() fonksiyonu çağrıldığında, generators üyesinde bulunan tüm nesnelerden captureFor fonksiyonu çağrılarak nokta bulutları sağlanır. Daha sonra her bir nokta bulutu pointCloud üyesine eklenir.

**PointCloudFilter Sınıfı:** RadiusOutlierFilter ve PassThroughFilter sınıflarını PointCloudFilter abstract sınıfı altında toplar.

**RadiusOutlierFilter Sınıfı:** Bu nokta bulutunda filtreleme yapar. Filter fonksiyonu ile nokta bulutunu alır ve filtrelenmiş halini döndürür. Bu filtreleme işleminin algoritması şu şekildedir. Nokta bulutundaki her bir nokta için tek tek işlem yapılır. Noktaya, radius değerinden daha yakın başka bir nokta yok ise, bu nokta nokta bulutundan çıkarılır.

**PassThroughFilter Sınıfı:** Bu nokta bulutunda filtreleme yapar. Filter fonksiyonu ile nokta bulutunu alır ve filtrelenmiş halini döndürür. Bu filtreleme işleminde, noktanın x, y ve z değerlerinden en az birisi limitlerin dışında ise, bu nokta nokta bulutundan çıkarılır.

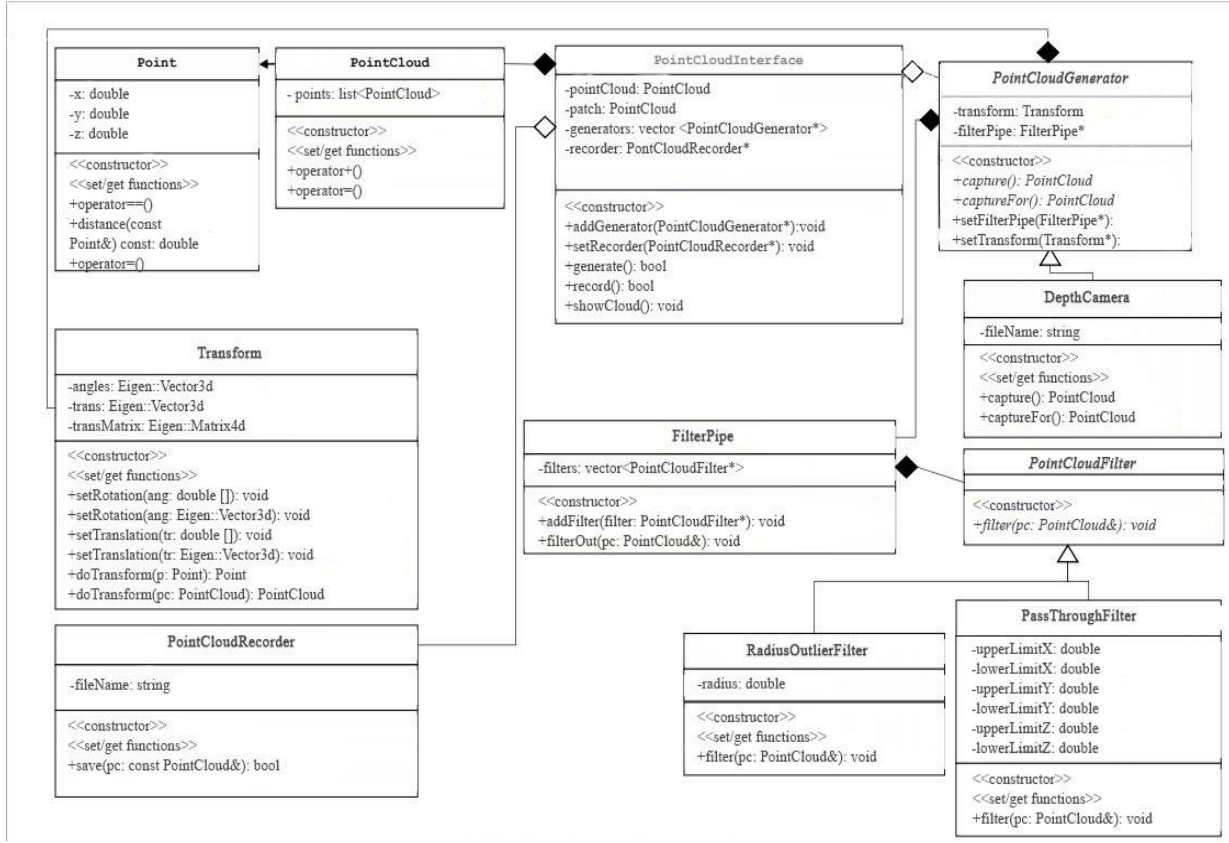
**PointCloudRecorder Sınıfı:** Bu nokta bulutlarının dosyaya kaydedilmesi için kullanılmaktadır. Save fonksiyonu çağrıldığında, fileName ile ismi verilen dosya açılır, parametre olarak verilen nokta bulutundaki noktalar bu dosyaya kaydedilir.

Sınıfların birçoğu için ayrıca test uygulaması geliştirildi.

## 2.2 Örnek Program Girdi ve Çıktıları

Girdi olarak programa iki adet içinde x,y ve z değerleri bulunduran iki adet .txt dosyası tanımlanır. Çıktı olarak ise bu değerlerin süzgeçten geçirilmiş ve dönüştürülmüş halini yine .txt dosyasına kaydeder.

## 2.3 UML Diyagramları

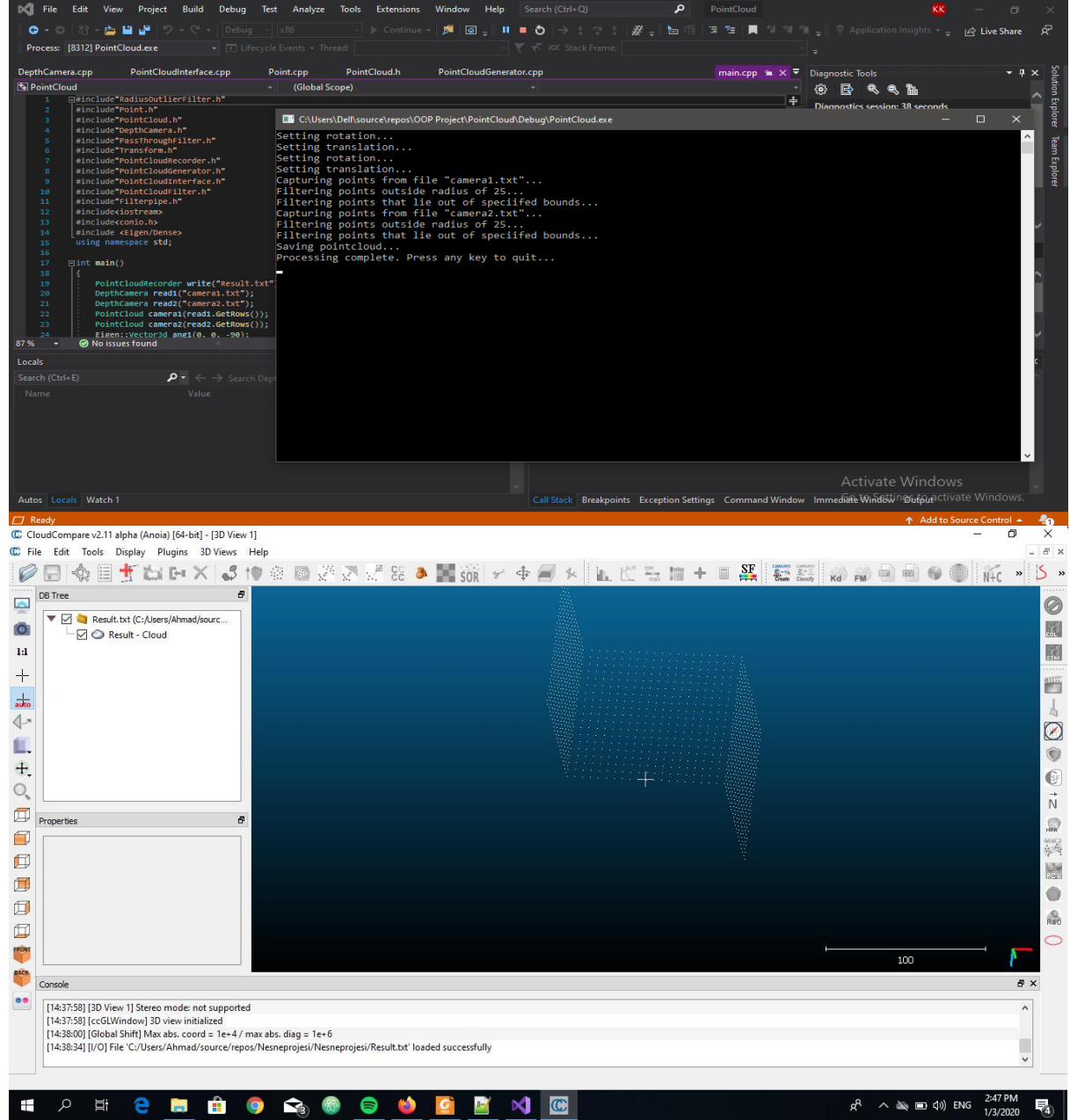


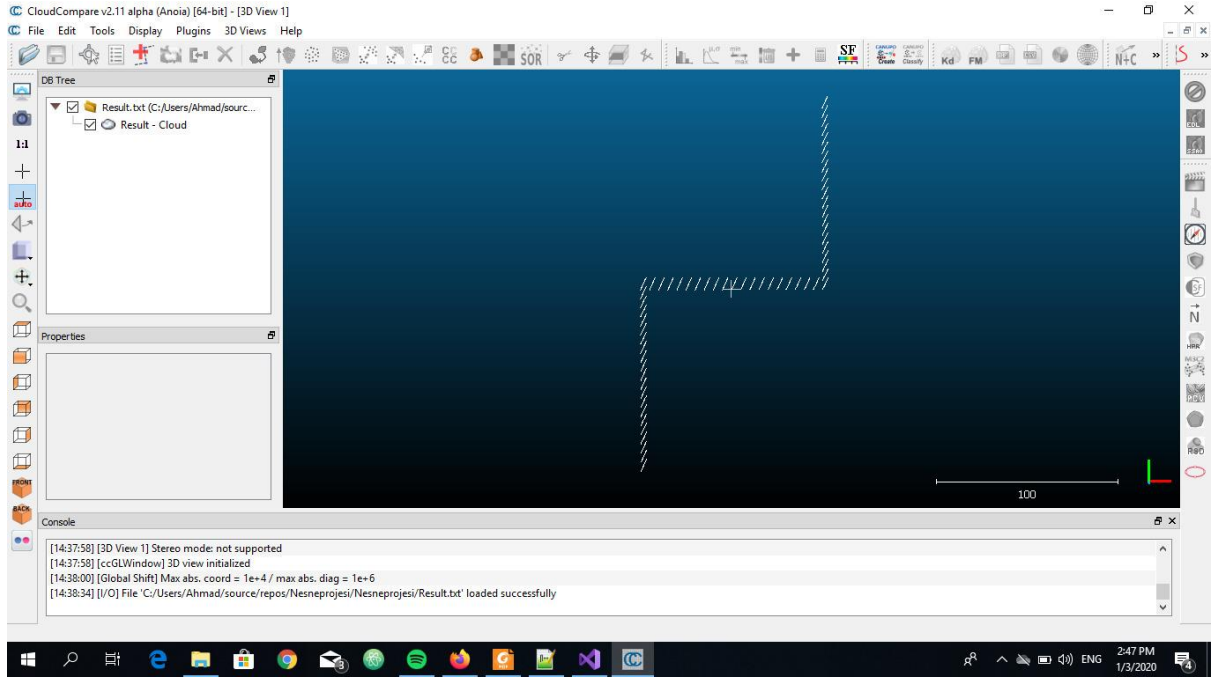
## 2.4 Görev Atamaları

Grup Üyesi	Görevler
AHMAD ZAIDAN	Transform ve DepthCamera Sınıfları.
ENES KAŞIKÇI	RadiusOutlier, PointcloudFilter Sınıfı.
BOBURSOH ABDULOV	Point, PointcloudGenerator Sınıfı
KUMBUKANI KAMANGA	PointCloudRecorder, PointCloud, Pointcloud Interface Sınıfları
MURAT MUSA YALÇIN	PassThroughFilter, Filterpipe Sınıfı ve Doxygen.

### 3. Sonular

Projede sonu olarak amaca ulaşıldı. Kod kontrolleri ve birleřtirmeler Bitbucket zerinde yapıldı. Ařağıda program ıktıları ve son nokta bulutu grsel olarak bulunmaktadır.





Takım çalışması iş bölümü yapıldığı için işimizi kolaylaştırdı.  
Birbirimizin eksikliğini yardımlaşarak çözmeye imkânı tanıdı.

Gelecek çalışmalar için takım arası iletişim güçlendirilebilirse  
daha iyi işler çıkarılabilir.