



**NOKTA BULUTU VERİLERİNDEN  
NESNE AYIRT ETME**

**Yük. Müh. Erdal ÖZBAY**

**Doktora Tezi**  
**Bilgisayar Mühendisliği Anabilim Dalı**  
**Danışmanı: Dr. Öğr. Üyesi Ahmet ÇINAR**

**HAZİRAN - 2018**

T.C.  
FIRAT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

NOKTA BULUTU VERİLERİNDEN NESNE AYIRT ETME

DOKTORA TEZİ  
Yük. Müh. Erdal ÖZBAY  
(131129203)

Tezin Enstitüye Verildiği Tarih : 16.05.2018

Tezin Savunulduğu Tarih : 20.06.2018

Tez Danışmanı : Dr. Öğr. Üyesi Ahmet ÇINAR (F.Ü.) 

Diğer Jüri Üyeleri : Prof. Dr. Abdulkadir ŞENGÜR (F.Ü.) 

: Doç. Dr. Burhan ERGEN (F.Ü.) 

: Prof. Dr. Ali KARCI (İ.Ü.) 

: Dr. Öğr. Üyesi Burhan SELÇUK (K.Ü.) 

HAZİRAN - 2018

## ÖNSÖZ

Bu tez çalışması, kapalı ortamlardaki nesnelerin üç boyutlu şekillerine ait tam bir nokta bulutunun, ilk bakışta buna uygun olmadığı düşünülen Kinect algılayıcı ile elde edilebileceğini ortaya koymaktadır. Bu nedenle aynı nesnenin, farklı açılardan elde edilen taramalarına ait nokta bulutlarının, 3B geometrik dönüşümlerinin gerçekleştirilmesi için bir çerçeve oluşturulmuştur. Pratik bir şekilde tarama verisine doğrudan uygulanabilen bir hizalama tekniği, 3B dönüşüm işlemi sonrasında yürütülebilmesi için çerçeveye eklenmiştir. Elde edilen 3B taramanın daha düzenli ve doğal bir görünüme sahip olması için bir iyileştirme yöntemi uygulanmıştır. Deneysel bulgulara göre noktaların daha düzenli ve yüzey örgü yapısının daha yumuşak bir yapıya kavuştuğu gösterilmiştir. Bir sahne içerisindeki nesnenin ayırt edilmesi için elde edilen 3B tam bir nokta bulutu üzerinden nesnelerin iskelet yapılarına ait bilgiler çıkarılmaktadır. Bu iskelet çıkarımları, 3B nesnelerin, algılama, ayırt etme ve tanıma işlemlerine önemli ölçüde katkı sağlamaktadır.

Yaptığım tez çalışmasının, konu ile ilgilenen lisans ve lisansüstü öğrencilerinin yanı sıra bu konuda çalışmak isteyenlere bir rehber olacağını umuyorum.

Doktora eğitimim boyunca hem akademisyenlik mesleğinde hem de özel hayattaki yaklaşımlarını örnek aldığım, mesleki engin bilgi ve deneyimlerini cömertçe benimle paylaşmaktan kaçınmayan, her konuda sonsuz desteğini yanımda hissettiğim saygıdeğer hocam, Dr. Öğr. Üyesi Ahmet ÇINAR'a ve bu süreç içerisinde her konuda sabırla bana destek veren değerli eşime ve aileme çok teşekkür ederim.

**Erdal ÖZBAY**

**ELAZIĞ - 2018**

## İÇİNDEKİLER

|                                                          | <u>Sayfa No</u> |
|----------------------------------------------------------|-----------------|
| <b>ÖNSÖZ</b> .....                                       | <b>II</b>       |
| <b>İÇİNDEKİLER</b> .....                                 | <b>III</b>      |
| <b>ÖZET</b> .....                                        | <b>IV</b>       |
| <b>SUMMARY</b> .....                                     | <b>V</b>        |
| <b>ŞEKİLLER LİSTESİ</b> .....                            | <b>VI</b>       |
| <b>TABLOLAR LİSTESİ</b> .....                            | <b>VIII</b>     |
| <b>KISALTMALAR</b> .....                                 | <b>IX</b>       |
| <b>SEMBOLLER LİSTESİ</b> .....                           | <b>X</b>        |
| <b>1. GİRİŞ</b> .....                                    | <b>1</b>        |
| 1.1. Tez Çalışmasının Amacı .....                        | 4               |
| 1.2. Tezin İçeriği .....                                 | 6               |
| <b>2. NESNE TARAMA YÖNTEMLERİ</b> .....                  | <b>7</b>        |
| 2.1. Temaslı Yöntemler .....                             | 8               |
| 2.2. Temassız Aktif Yöntemler .....                      | 9               |
| 2.2.1. Uçuş Zamanı (Time-of-Flight) Yöntemi .....        | 10              |
| 2.2.2. Lazer Tarama Yöntemi.....                         | 12              |
| 2.2.3. Yapısal Işık Yöntemi.....                         | 14              |
| 2.2.4. Konoskopik Holografı Yöntemi.....                 | 16              |
| 2.3. Temassız Pasif Yöntemler .....                      | 17              |
| 2.4. Sonuç .....                                         | 22              |
| <b>3. ÜÇ BOYUTLU (3B) NESNE TARAMA</b> .....             | <b>23</b>       |
| 3.1. Nokta Bulutu .....                                  | 24              |
| 3.2. Kamera İç Yapısı ve Kalibrasyonu .....              | 26              |
| 3.3. Yazılım Bileşenleri ve Algılayıcı Yerleştirme ..... | 31              |
| 3.4. Sayısal Yöntemler .....                             | 34              |
| 3.4.1. ICP (Yinelemeli En Yakın Nokta) .....             | 35              |
| 3.4.2. RANSAC (Rastgele Örnek Konsensüsü) .....          | 37              |
| 3.4.3. Derinlik-Ölçeği Yöntemi.....                      | 38              |
| 3.4.4. Lowe'nin SNN Yöntemi .....                        | 39              |
| 3.4.5. 3B Katı Dönüşüm Tahmini .....                     | 42              |
| 3.5. Sonuç .....                                         | 43              |
| <b>4. 3B TARAMA SİSTEMİ</b> .....                        | <b>44</b>       |
| 4.1. 3B Kayıt İşlemi .....                               | 46              |
| 4.2. 3B Geometrik Dönüştürme .....                       | 53              |
| 4.3. Hizalama.....                                       | 57              |
| 4.4. İyileştirme.....                                    | 58              |
| 4.5. Sonuç .....                                         | 62              |
| <b>5. İSKELET ÇIKARMA</b> .....                          | <b>64</b>       |
| 5.1. Laplace Yöntemi .....                               | 66              |
| 5.2. Medial Yöntemi.....                                 | 68              |
| 5.3. Uygulama .....                                      | 72              |
| 5.4. Sonuç .....                                         | 75              |
| <b>6. SONUÇ</b> .....                                    | <b>76</b>       |
| <b>KAYNAKLAR</b> .....                                   | <b>77</b>       |
| <b>ÖZGEÇMİŞ</b> .....                                    | <b>86</b>       |

## ÖZET

Nesnelerin ayırt edilmesi problemlerine, bilgisayar grafiği ve bilgisayar görmesi alanlarında, uzun yıllardır çok çeşitli çalışmalarla yanıt aranmaktadır. Bu problemin çözümü için görüntü tabanlı birçok yaklaşım üretilmesine rağmen, bu çözümlerin bazı alanlarda kullanılabilmesi için yetersiz kaldığı düşünülmektedir.

Günlük hayatta sıklıkla karşılaşılan nesneler ve şekiller arasında ayırım yapabilme ve tanıma, bu alanlardaki önemli bir sorundur. Tez çalışmasında, kapalı bir sahne içerisindeki fiziksel nesnelerin ayırt edilmesi adına, gerçek hayatta kullanılan nesnelere ait nokta bulutları üzerinden yeni bir iskelet eğrisi çıkarma algoritması uygulanmaktadır. 3B nesnenin iskelet eğrisi, 3B şekillerin soyut olarak, geometrik ve topolojik bir gösterimidir. Bu iskelet yapısı var olan nokta bulutu içerisinde elde edilmiş bir ortalama nokta bulutu verilerinin bütünleşmiş halidir.

Nokta bulutu verileri üzerinden iskelet bilgisi çıkarımı için Medial iskelet çıkarma algoritması ve ona yardımcı Laplace daraltma algoritması uygulanmaktadır. Mevcut iskelet çıkarım algoritmalarının her birinin kendine özgü kullandığı bir veri türü ve işleyiş biçimi bulunmaktadır. Sağlam bir iskelet yapısı elde etmenin en temel yolu, nesnelere ait nokta bulutunun, mümkün olduğunca kusursuz olmasıdır. Bunun yolu, nesnelere ait iyi bir tarama işlemi yerine getirmekten geçmektedir. Bu alanlardaki tarama işlemlerinde, çoklu ya da hareketli sensör kullanımı şeklinde, çeşitli uygulamalar bulunmaktadır. Bu bakımdan tez çalışmasında, kullanımı geleneksel tarayıcılara kıyasla daha pratik ve maliyeti düşük olan, sabit bir Kinect sensör vasıtasıyla, kullanıcının nesne için uygulayacağı doğal bir hareketle, tarama üreten bir yaklaşım uygulanmaktadır. Bu yaklaşım kapalı mekân ve kısa mesafede bulunan 3B nesnelerin, geniş bir yelpazede tam (360 derece) nokta bulutu verisinin üretilmesine olanak tanımaktadır. Nesne tarama için oluşturulan sistem kolay kurulabilir, basit ve etkileyici sonuçlar üretmektedir. Sabitlenmiş tek bir Kinect sensör karşısında, döner tabla üzerinde duran 3B nesnenin, belirli açılarla (ör. 90) döndürülerek birden fazla nokta bulutu tarama verisi elde edilmektedir. Birleştirme ve hizalama işlemleri için her bir taramanın ağırlık merkezi (0, 0, 0) olacak şekilde tüm nokta bulutları ötelenmektedir. İkinci ve daha sonraki taramalardan elde edilen nokta bulutu verileri için sırasıyla, merkez noktasına (0, 0, 0) göre y-ekseni hizasında dönüşüm işlemi yapılmaktadır. Farklı açılardan elde edilmiş, dönüştürülmüş nokta bulutu verileri, belirlenen birleşim anahtar noktalarına göre ötelenerek, sırasıyla birbirlerine göre hizalanmaktadır. Elde edilen tam bir kaba 3B nokta bulutu verisi üzerinde, düzeltme ve iyileştirme işlemleri için kesitler üzerinde çalışan bir algoritma yürütülmektedir. Böylece elde edilen tarama, veri kalabalığından arınmış, 3B, sade ve düzenli bir yapıya kavuşmuştur. Bu bakımdan, nokta bulutlarının, iskelet çıkarımı yapılabilecek en uygun hale getirilmesi sağlanmıştır.

**Anahtar Kelimeler:** Nokta bulutu, Nesne ayırt etme, Kinect, İskelet eğrisi, 3B Tarama, Hizalama, İyileştirme.

## SUMMARY

### Distinguishing Objects On Point Cloud Data

For many years, a variety of studies have been sought in the problems of distinction of objects, in computer graphics and computer vision areas. Although there are many image-based approaches to solve this problem, it is thought that these solutions are insufficient to be used in some areas.

Distinguishing and recognizing between objects and shapes that are frequently encountered in everyday life is an important question in these areas. In this thesis, a curve skeleton extraction algorithm is applied to the point clouds of objects used in real life in order to distinguish the physical objects in a scene. The curve skeleton of the 3D object is an abstract, geometric and topological representation of 3D shapes. This skeleton structure is the integrated state of an average point cloud data obtained from the existing point cloud.

Medial skeleton extraction algorithm and its assistant Laplace shrinking algorithm are applied for extracting curve skeleton information on point cloud data. There is a data type and a method of operation that each of the existing skeletal extraction algorithms uses in its own way. The most basic way of achieving a robust skeletal structure is that the point cloud of objects is being as perfect as possible. Its way is to implement a good scanning of the objects. There are a variety of applications in this area, such as the use of multiple or moving sensors in the scanning process. In this thesis, an approach which generates a scan with a natural movement of the user for the object is implemented by means of a fixed Kinect sensor, which is more practical and cost-effective than conventional scanners. This approach allows the production of indoor and short-range 3D objects in a wide range of full (360 degrees) point cloud data. The system for object scanning is easy to set up, producing simple and impressive results. In response to a single fixed Kinect sensor, the 3D object standing on the turntable is rotated at specific angles (eg. 90) to obtain multiple point cloud scan data. All the point clouds are shifted so that the center of gravity of each scan is (0, 0, 0) for merging and aligning operations. The 3D rotation process is performed in the y-axis direction with respect to the center point (0, 0, 0) for the point cloud data obtained from the second and subsequent scans. The transformed point cloud data obtained from the different angles are aligned with respect to each other, shifted relative to the specified junction key points. An algorithm running on the cross sections for the improvement and reduction processes is carried out on the obtained full 3D coarse point cloud data. Thus, the obtained scan is free from the data and has a 3D, simple and regular structure. In this respect, it has been ensured that the point clouds are optimized to allow skeleton extraction.

**Keywords:** Point cloud, Object distinguishing, Kinect, Curve skeleton, 3D Scanning, Alignment, Improvement.

## ŞEKİLLER LİSTESİ

### Sayfa No

|             |                                                                         |    |
|-------------|-------------------------------------------------------------------------|----|
| Şekil 1.1.  | Nokta bulutu üzerinden iskelet yapısı çıkarılmış Ejderha heykeli .....  | 3  |
| Şekil 1.2.  | At nesnesi tanıma işlemi için çıkarılan iskelet yapısı örneği .....     | 4  |
| Şekil 2.1.  | Sert dik bir kola sahip bir koordinat ölçme makinesi.....               | 8  |
| Şekil 2.2.  | Bir ToF Kamera (Microsoft Kinect Xbox One).....                         | 10 |
| Şekil 2.3.  | ToF Kameranın çalışma prensibi .....                                    | 12 |
| Şekil 2.4.  | Lazer üçgenleme sensörünün çalışma prensibi .....                       | 12 |
| Şekil 2.5.  | Yapısal ışık yöntemi kullanılarak oluşturulan 3B model .....            | 14 |
| Şekil 2.6.  | Konoskopik holografi sensörünün çalışma prensibi .....                  | 17 |
| Şekil 2.7.  | Fotometrik yöntem ile elde edilen 3B yüz modeli .....                   | 19 |
| Şekil 2.8.  | Görüntü tabanlı yöntem ile oluşturulmuş bir model .....                 | 21 |
| Şekil 3.1.  | Birden çok nokta bulutunun kayıt edilmesi.....                          | 23 |
| Şekil 3.2.  | Kinect ile yakalanan bir iç sahnenin nokta bulutu verisi .....          | 24 |
| Şekil 3.3.  | Nokta bulutu voksel yapısı .....                                        | 25 |
| Şekil 3.4.  | Kinect bileşenlerinin kamera üzerinde gösterimi.....                    | 26 |
| Şekil 3.5.  | Kinect algılayıcının derinlik ölçüm aralığı .....                       | 27 |
| Şekil 3.6.  | Kinect algılayıcının derinlik ölçümü.....                               | 29 |
| Şekil 3.7.  | Kinect ile yakalanmış RGB, derinlik ve nokta bulutu görüntüleri.....    | 30 |
| Şekil 3.8.  | Kinect uygulamasında yazılım-donanım etkileşimi.....                    | 32 |
| Şekil 3.9.  | Kinect SDK mimarisinin çalışma prensibi.....                            | 33 |
| Şekil 3.10. | Algılayıcı ve tarama nesnesinin yerleştirme pozisyonları.....           | 33 |
| Şekil 3.11. | ICP algoritmasında örtüşen nokta bulutu setleri .....                   | 35 |
| Şekil 3.12. | ICP algoritmasında kapalı formda tam örtüşen nokta bulutu setleri.....  | 37 |
| Şekil 3.13. | RANSAC algoritmasının iki adım uygulanması .....                        | 38 |
| Şekil 3.14. | Lowe'nin ölçütü ile KNN eşleşme grafiği .....                           | 40 |
| Şekil 4.1.  | Tarama sistemi kurulumu.....                                            | 45 |
| Şekil 4.2.  | Tarayıcı ölçümlerinde koordinat eksen ve sensör arasındaki ilişki ..... | 47 |
| Şekil 4.3.  | Tek bir sabit sensör ile hareketli nesne tarama sistemi .....           | 48 |
| Şekil 4.4.  | Tek bir hareketli sensör ile sabit nesne tarama sistemi .....           | 49 |
| Şekil 4.5.  | Çoklu sabit sensörler ile sabit nesne tarama sistemi .....              | 50 |
| Şekil 4.6.  | Çoklu sabit sensörler ile hareketli nesne tarama sistemi.....           | 51 |

|                    |                                                                                                             |    |
|--------------------|-------------------------------------------------------------------------------------------------------------|----|
| <b>Şekil 4.7.</b>  | Tarama sistemimizdeki insan vücuduna ait ham verilerin farklı açılardan tek bir Kinect ile yakalanması..... | 53 |
| <b>Şekil 4.8.</b>  | Ağırlık merkezi noktası orijine kaydırılan farklı taramalar .....                                           | 54 |
| <b>Şekil 4.9.</b>  | Orijin merkezine taşınan taramalarda 3B geometrik dönüşüm işlemi .....                                      | 56 |
| <b>Şekil 4.10.</b> | Çoklu tarama için gerçekleştirilen hizalama adımları.....                                                   | 57 |
| <b>Şekil 4.11.</b> | Kesit çıkarım senaryosu ve üç-boyutlu çıkarılan kesitler .....                                              | 59 |
| <b>Şekil 4.12.</b> | Kesitlerin hayali vokselleştirme prosedürü.....                                                             | 60 |
| <b>Şekil 4.13.</b> | Nokta bulutu kaydı ve farklı boyutlardaki iyileştirme sonuçları.....                                        | 60 |
| <b>Şekil 4.14.</b> | Kayıtlı ve iyileştirilmiş nokta bulutu yüzey örgü özellikleri değişimi.....                                 | 62 |
| <b>Şekil 5.1.</b>  | Ham nokta bulutuna uygulanan Laplace daraltma adımı.....                                                    | 67 |
| <b>Şekil 5.2.</b>  | Medianların verilerdeki gürültüye ve aşırılıklara duyarlılığı .....                                         | 70 |
| <b>Şekil 5.3.</b>  | Farklı kuvvetlere ( $\mu$ ) sahip koşullu öteleme yapılan nokta bulutları.....                              | 70 |
| <b>Şekil 5.4.</b>  | Medial iskelet algoritmasının tam bir eğri iskelet çıkarımı.....                                            | 72 |
| <b>Şekil 5.5.</b>  | Farklı veri setleri için uygulanan yöntemlere ait sonuçlar .....                                            | 74 |



## TABLÖLAR LİSTESİ

|                                                                                                       | <u>Sayfa No</u> |
|-------------------------------------------------------------------------------------------------------|-----------------|
| <b>Tablo 3.1.</b> Kinect kalibrasyonunun döndürme ( $R$ ) ve dönüştürme ( $T$ ) parametreleri. ....   | 30              |
| <b>Tablo 3.2.</b> SIFT, SURF eşleştirme algoritması sonuçları.....                                    | 41              |
| <b>Tablo 3.3.</b> SIFT, SURF özellik eşleştirme sayısı ve çalışma süreleri. ....                      | 41              |
| <b>Tablo 5.1.</b> İskelet çıkarım algoritmalarının çalışma süreleri (sn) $t_1$ , $t_2$ ve $t_3$ ..... | 73              |



## KISALTMALAR

|               |                                            |
|---------------|--------------------------------------------|
| <b>1B</b>     | : Bir Boyutlu                              |
| <b>2B</b>     | : İki Boyutlu                              |
| <b>3B</b>     | : Üç Boyutlu                               |
| <b>3DMD</b>   | : 3 Dimensional Multi Dynamic              |
| <b>3DMM</b>   | : 3D Morphable Models                      |
| <b>API</b>    | : Application Programming Interface        |
| <b>BSD</b>    | : Brekel Software Development              |
| <b>CCD</b>    | : Charge-Coupled Device                    |
| <b>CMOS</b>   | : Complimentary Metal-Oxide Semiconductor  |
| <b>CMOS</b>   | : Complementary Metal-Oxide Semiconductor  |
| <b>CSP</b>    | : Colored Surface Point                    |
| <b>DI4D</b>   | : Dimensional Imaging 4 Diemension         |
| <b>DoF</b>    | : Degree of Freedom                        |
| <b>GPU</b>    | : Graphics Processing Unit                 |
| <b>ICP</b>    | : Iterative Closest Point                  |
| <b>KNN</b>    | : K-Nearest Neighbors                      |
| <b>KÖM</b>    | : Koordinat Ölçme Makinesi                 |
| <b>LIDAR</b>  | : Laser Imaging Detection and Ranging      |
| <b>MCS</b>    | : Model Coordinate System                  |
| <b>NUI</b>    | : Doğal Kullanıcı Arabirimi                |
| <b>OpenNI</b> | : Açık Doğal Etkileşim                     |
| <b>PCL</b>    | : Point Cloud Library                      |
| <b>RANSAC</b> | : Random Sample Consensus                  |
| <b>RGB</b>    | : Kırmızı, Yeşil, Mavi bileşenleri         |
| <b>ROS</b>    | : Robot Operating System                   |
| <b>SCAPE</b>  | : Shape Completion and Animation of People |
| <b>SDK</b>    | : Software Development Kit                 |
| <b>SFS</b>    | : Shape-From-Silhouette                    |
| <b>SLAM</b>   | : Simultaneous Localization and Mapping    |
| <b>SNN</b>    | : Second-Nearest Neighbors                 |
| <b>ToF</b>    | : Time-of-Flight                           |

## SEMBOLLER LİSTESİ

|                      |                                                             |
|----------------------|-------------------------------------------------------------|
| $c$                  | : Işık hızı                                                 |
| $c_{xir}, c_{yir}$   | : IR algılayıcının kamera temel nokta pozisyonlarını        |
| $c_x, c_y, c_z$      | : Yön kosinüsleri                                           |
| $d_m$                | : Derinlik mesafesi                                         |
| $D$                  | : Dengesizlik durumu                                        |
| $f_{xir}, f_{yir}$   | : Odak uzaklıkları                                          |
| $h$                  | : Destek yarıçapı                                           |
| $I$                  | : Yansıtılan noktalar kümesi indisi                         |
| $J$                  | : Giriş noktalar kümesi indisi                              |
| $L_f$                | : Kızılötesi kameranın odak uzaklığı                        |
| $L_b$                | : Taban uzunluğu                                            |
| $L^{t+1}$            | : Laplace operatörü                                         |
| $N$                  | : Küme sayısı                                               |
| $P^{t+1}$            | : Mevcut nokta bulutu                                       |
| $P'$                 | : Daralan nokta bulutu                                      |
| $q_1$                | : Işık yayıldığında pikselde biriken yük                    |
| $q_2$                | : Işık olmadığı zaman biriken yük                           |
| $R$                  | : Rotasyon matrisi                                          |
| $S$                  | : Modeli oturtmak için gereken minimum sayı                 |
| $S_i^0$              | : $i$ noktasının orijinal komşuluk kapsamı                  |
| $S_i^t$              | : $i$ noktasının mevcut komşuluk kapsamı                    |
| $t$                  | : Işık uzunluğu                                             |
| $T$                  | : Dönüşüm matrisi                                           |
| $V$                  | : Gürültü matrisi                                           |
| $W_L, W_H$           | : Çapraz ağırlık matrisleri                                 |
| $x_p, y_p$           | : Temel ofset noktaları                                     |
| $x_0, y_0, z_0$      | : Dönüştürme koordinat merkezi                              |
| $Z_p$                | : Referans düzlemine olan uzaklık                           |
| $Z_t$                | : Nesnenin derinlik mesafesinin $t$ noktasına olan uzaklığı |
| $\delta_x, \delta_y$ | : Lens bozulma katsayıları                                  |
| $\theta$             | : Ağırlık fonksiyonu                                        |
| $\mu$                | : Öteleme kuvveti                                           |

## 1. GİRİŞ

Günümüzde insan gözünün gördüğü şekliyle, üç boyutlu gerçek dünya nesnelerini, iki boyutlu bir görüntü yapısı ile içeriğini düzenleyip, yeniden yapılandırmak için çözüm üretmek son derece zordur. Bu etkenlerin çoğu mevcut görüntü tabanlı yöntemlerin, ışık miktarı, görüş açısı yetersizliği, yansımalar gibi dış ortama olan hassasiyetinden ve öğrenme tabanlı yöntemlerin de kıyaslama yapabileceği bir veri tabanı gereksiniminden kaynaklanmaktadır. Bu nedenlerden dolayı, özellikle son birkaç yılda yapılan 3B (Üç Boyutlu) nesne ayırt etme uygulamalarında, 2B (İki Boyutlu) görüntü işlemleri yerine, doğruluğu daha yüksek seviyelerde ve dış ortam değişkenlerinden daha az etkilenen nesnelerin derinlik bilgisi üzerinde çalışılmaktadır. Üç boyutlu nesne tarama teknolojileri, uzun yıllardır bilgisayar görmesi ve 3B grafik uygulamaları üzerinde çalışan geliştiriciler arasında yaygın olarak kullanılmaktadır ve araştırmacılar için geliştirilmeye açık ve oldukça popüler bir konudur. Bu teknolojiler çeşitli uygulamalarla birlikte kullanılabilir. Bu uygulamalar arasındaki en yaygın olanları, insan tespiti için geliştirilen gözetim sistemleri, nesne algılama, tanıma ve kimlik vermek, robotik uygulamalar, animasyon, video oyunları, üç boyutlu grafik, mobil robot, tıbbi teşhis, ameliyatlar ve rehabilitasyon gibi uygulamalardır [1].

Bilgisayar grafiği alanlarında, son yıllarda yapılan uygulamaların önemli bir çoğunluğunda, sürecin ilk adımı olarak 3B tarama işlemi yerine getirilmektedir. Çoğu 3B tarama teknolojilerinin, veri girişi için başvurduğu geleneksel yöntemlerde stereo kameralar kullanılmaktadır. Bu teknolojilerin temel çalışma ilkesinde bir trigonometri tekniği yer almaktadır. Bu teknik, kameraların, insan gözünün görebildiği binoküler görüntünün benzetimini yapabilmesine ve böylece üç boyutlu görüntüleri (stereo fotoğraf) yakalayabilmesine imkân vermektedir. Bilgisayar görmesi alanında son yıllarda yapılan birçok araştırmada, stereo kameralar için farklı teknikler geliştirilmiştir ancak hala bazı hata ve kararsızlıkların olduğu aşikârdır. Doğası gereği birden fazla kamera kullanımını gerektiren stereo sistemine, bu açıdan bakıldığında, kısmen kusurlu olduğu söylenebilmektedir. Tarama ve derinlik hesaplama yeteneklerine sahip cihazlar arasında, son birkaç yıldır popüler olan ve yeni bir tür derinlik kamerası özelliği taşıyan Microsoft Kinect kamera, çeşitli alanlarda yaygın olarak kullanılmaya başlanmıştır. Derinlik özelliği taşıyan bu tür kameralar arasında hızla yerini alan Kinect algılayıcı kamera yeni bir tür

olarak toplumda büyük bir ilgi çekmiştir. Kinect algılayıcı ile geleneksel 3B tarayıcılar karşılaştırıldığında, ışık ve doku durumuna karşı aşırı toleranslı olan şartlarda bile daha az dikkatle, video oranında görüntü ve derinlik bilgisi üretmeyi başarmaktadır. Kinect'in bu tür özelliklerinin yanı sıra, normal bir video kamera olarak da kullanılabilmesi, düşük maliyeti, kompakt olması ve genel kullanıcılar tarafından kolaylıkla elde edilebilir olması onun önemli avantajlarından. Bu anlamda çalışmamızda stereo kameralar yerine yeni bir cihaz sayılan Microsoft Kinect gibi derinlik algılayıcı kameraların kullanılmasının faydalı olacağı kanaatindeyiz. Microsoft Kinect'in algılayıcı kameraların, uygun maliyetli, pratik ve kolay uygulanabilir olması gibi birçok artıları bulunmaktadır. Bunun yanında, tarama nesnesi ile sensör arasındaki mesafenin artması çözünürlük hassasiyetinin azalmasına neden olması, onun en büyük eksisidir [2].

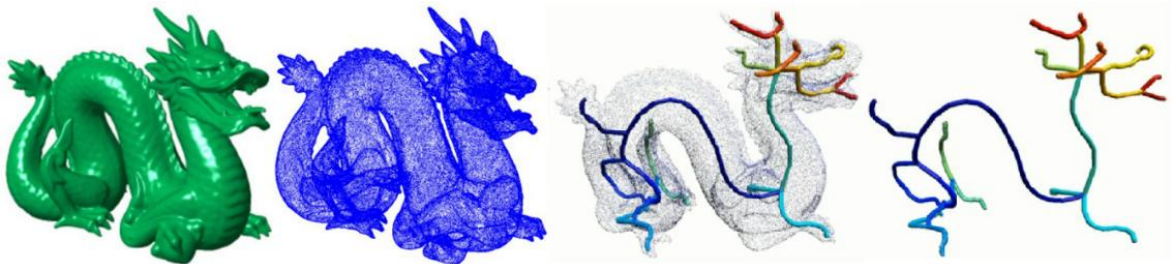
3B tarama yöntemleri ile elde edilen nokta bulutu verileri üzerinden nesne ayırt etme işlemi zorlu bir uygulamadır. 3B verilere ait sayısal bilgiler üzerinden nesneye ait çıkarımlarda bulunmak zahmetli bir süreçtir. Çünkü 3B bir nesnenin tek bir bakış açısı ile derinlik verileri üzerinden ayırt etme işlemleri yetersiz kalmaktadır. Bu nedenle nesne tarama işlemlerinde önceliğimiz 3B tam bir 360 derece tarama üretmektir. Ancak tek bir derinlik algılayıcı kamera kullanarak tam bir 3B nokta bulutu verisi üretmek de başlı başına bir problemdir. Ancak birkaç yıl öncesinde Kinect'in küresel piyasa için serbest bırakılmasıyla bilgisayar görmesi konusundaki uygulamaların geliştirilmesi için yeni bir kapsam oluşturulmuştur. Bu durumun başlıca nedeni, Kinect'in çalışma prensibini daha hızlı ve kolay hale getirmesine yardımcı olan, her bir çerçeve için derinlik bilgisini üretebiliyor olmasıdır. Buna karşın, Kinect tarafından sağlanan derinlik verilerinde mesafeye bağlı olarak artış gösteren gürültülü veriler onun en büyük dezavantajdır.

Bir sahne içerisindeki 3B nesnelerin nokta bulutu verileri üzerinden ayırt edilmesi için nesnelerin iskelet yapılarının oluşturulması hedeflenmiştir. Ancak nokta bulutu üzerinden bir nesneye ait iskelet yapılarını çıkarabilmek için öncelikle tam bir tarama işlemi gerçekleştirilmelidir. Çünkü tek bir bakış açısı ile üretilen nokta bulutu verileri ile çıkarılan iskelet yapıları, tam bir taramada olduğu kadar sağlıklı ve doğal sonuçlar vermemektedir. Bu bakımdan öncelikle nesnelerin tam bir 360 derece taramasını gerçekleştirmek adına, sabit tek bir Kinect kamera kullanarak aynı nesneye ait elde edilen birden fazla taramanın, 3B döndürme, hizalama ve birleştirme işlemleri gerçekleştirilmektedir. Detaylarına tezin üçüncü bölümünde yer verilen bu adımların uygulanması, bir yazılım çerçevesi içerisinde uygulanmıştır. Bu işlemlerle, nokta bulutlarının tek bir veri seti içerisinde birleştirilmesi

neticesinde 3B nesneye ait yoğun bir nokta bulutu kümesi oluşturulmaktadır. Ayrıca farklı taramaların birleştirilmesi için yapılan hizalama işlemlerinde, tarama birleşim kısımlarında, nokta bulutlarında aşırılık veya eksiklik gibi durumlar meydana gelebilmektedir. Yine bu gibi olumsuz durumların önüne geçmek için bir nokta bulutu azaltım ve iyileştirme işlemi yazılım çerçevesi içerisine dâhil edilmiştir. Böylece nokta bulutu yapısı daha seyrek ancak daha düzenli bir yapıya kavuşturulmaktadır.

Nesnelerin temsil edildiği 3B modellere ait veriler üzerinden çıkarımlar üretmek, son yıllardaki popüler yöntemler arasındadır. Bu yöntemler içerisinde en popüler olanlardan bir tanesi de iskelet verileri ile çalışmaktır. İskelet yapıları varlıkların fiziksel dış görünümünün birebir karşılığını yansıtmaktadır. Birçok çalışmada canlılara ait olan iskelet yapılarından bunların hangi tür canlı sınıfına ait olduğunu ve de hangi coğrafyada yaşam sürdüğüne dair çıkarımlarda bulunabilmektedir. 3B modellere ait nokta bulutu yapısı üzerinden elde edilen eğri iskelet yapısı ile bu çıkarımlar sayesinde modellerin topolojik karakteristik özellikleri temsil edilmektedir [3]. 3B nokta bulutları gerçek nesnelerin uzaysal konumlara karşılık gelen sayısal verilerin birleşiminden meydana gelmektedir. Nokta bulutları üzerinden elde edilecek eğri iskelet yapısı yine bir nokta kümesi ile temsil edilmektedir. Topolojik açıdan sade ve anlaşılır bir şekilde komple 1B (Bir Boyutlu) iskelet elde etmek için nesne eklem bölgeleri birleştirilerek tamamlanmaktadır. Nokta bulutu iskelet eğrisi bir nesnenin geometrik ve topolojik olarak soyutlanmış üç boyutlu şeklini temsil etmektedir. Eğri iskelet yapısı 3B nesneleri temsil etmek için yeterince özlü ve etkileyicidir. Ayrıca, nesnelerin topolojik yapısı bu yolla verimli bir şekilde temsil edilirken yüzey ayrıntılarından kaçınılmış ve veri işlenmesi için daha kolay bir ortam sağlanmıştır [4].

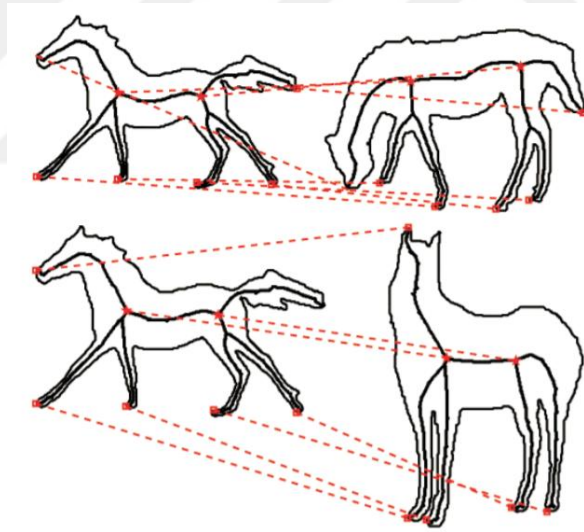
Şekil 1.1’de bir ejderha objesinin 3B tam taraması üzerinde yapılan bir iskelet çıkarımı örneği gösterilmektedir.



**Şekil 1.1.** Nokta bulutu üzerinden iskelet yapısı çıkarılmış Ejderha heykeli [5].

### 1.1. Tez Çalışmasının Amacı

Tez çalışmasındaki temel motivasyon, bir sahnenin, bir veya daha fazla tarama sonucunda elde ettiğimiz nokta bulutu ile sahne içerisindeki nesneye ait muhtemel iskelet yapısının oluşturularak, cismin ortamdaki ayırt edilebilmesini sağlamaktır. Çünkü nokta bulutu diye isimlendirilen yapı sadece üç boyutlu sayısal verilerden oluşmaktadır. Elimizde sadece somut olarak bulunan sayısal verilerin, nesnelerin ayırt edilebilmesi için çıkarım yapılabilecek hale getirilmesi gerekmektedir. Bu yapının, sadece geometrik yapıya sahip nesneler dışındaki tüm nesnelerde mevcut olan, iskelet yapısı üzerinden gerçekleştirilmesi planlanmıştır. Bu düşüncenin oluşmasındaki en önemli etken, günümüze kadar yapılan birçok çalışmada gerek gerçek nesnelerin, gerekse grafik alanında kullanılan modellerin, iskelet yapıları üzerinden doğrudan nesne tanıma hatta belirleme çalışmalarının yapılabilmesidir [6, 7]. Şekil 1.2’de bir çalışmada at nesnesine ait farklı açılardan elde edilen iskelet yapısı üzerinden tanıma işlemi yapılmaktadır.



Şekil 1.2. At nesnesi tanıma işlemi için çıkarılan iskelet yapısı örneği [6].

Nesnelere ait ayırt etme, algılama ya da tanıma işlemleri için kullanılan yöntemlerin büyük çoğunluğunda görüntü tabanlı filtrelemeler ve algoritmalar kullanılmaktadır. Görüntü tabanlı uygulamalarda uygulamalar nispeten daha kolay yapılabilmesine karşı sayısal bilgiler ile sonuca ulaşmadaki doğruluk oranı, daha yüksektir. Buna en basit bir örnek, son yıllarda üretilmeye çalışılan otonom araçlar için kullanılan teknolojiler, görüntü tabanlı değil, Lidar tarzı stereo kamera teknolojileridir. Çünkü bu tarz uygulamalar hata

kabul etmemekte ve kesinlik istemektedir. Bu bakımdan tez çalışması, görüntü tabanlı değil üç boyutlu sayısal veriler üzerinden gerçekleştirilecek şekilde tasarlanmıştır.

Sahnelerin sayısal verileri elde edilirken 3B tarayıcılar gibi Kinect algılayıcı kamera da yaygın olarak kullanılmaktadır. Tez çalışması, 3B nesne tarama yöntemlerine örnek bir çalışma olarak, kapalı ortamlarda yoğun 3B haritalar oluşturmak için piksel başına derinlik bilgileriyle birlikte RGB (Kırmızı, Yeşil, Mavi bileşenler) görüntüleri kullanmaktadır [8].

Kinect algılayıcı kameralar, yeni bir kavram olmasına rağmen derinlik algılayıcı kamera olarak, herkes tarafından erişilebilir hale gelmiş ve derinlik kameraları arasında yerini almıştır. Derinlik algılama, kalite, düşük maliyet ve gerçek zamanlı uygulanabilir olması açısından Kinect, alanında algılayıcıların kullanımı açısından meraklıları ve araştırmacılar arasında popüler hale gelmiştir. Biz nokta bulutu kütüphanesi PCL (Point Cloud Library) kullanarak kapalı bir ortamda bir 3B model oluşturmak için bu Kinect algılayıcı kamera yeteneklerinden faydalanmaktayız. Ayrıca Kinect ile elde edilen, birden fazla derinlik bilgisi içeren bir resim çerçeve kamera pozisyonu ve onun rotasyonu ile birleştirilirse daha yoğun bir 3B harita elde edilmek için de kullanılabilir. Metriksel işlemlerle, doğrudan derinlik bilgileri kullanılarak uygulanan iskelet çıkarım algoritmaları genellikle lazer tarayıcılardan elde edilmiş tam (360 derece) bir nokta bulutu ile çalışmaktadır. Bu bakımdan geliştirilen iskelet çıkarım algoritmaları da işlenecek verinin tam bir 360 derece tarama yapısında olması beklemektedir. Çalışmada, tam bir 360 derece tarama işlemi gerçekleştirilmesi öncesinde tek açıdan elde edilmiş nokta bulutları üzerinden iskelet çıkarım algoritmalarının yürütülebilmesi için denemeler yapılmıştır. Bu aşamada nokta bulutu verisine ait sağlıklı bir iskelet yapısının çıkarılması düzenli bir tam taramadan geçmektedir. Yöntem olarak, tek açıdan elde edilmiş nokta bulutu verileri üzerinden, Medial iskelet çıkarma algoritması kullanılarak, doğal iskelet sonuçları elde edilmeye çalışılmıştır. Bunun için doğrudan Kinect ile elde edilen, ham nokta bulutuna uygulanması mümkün olmayan veriler, Laplace daraltma işleminden geçirilmiştir. Böylece iskelet bilgisi çıkarılabilecek tek bir açıdan (örneğin profilden) elde edilen veriler üzerinde sağlık bir şekilde yürütülmesi sağlanmıştır. Amaç tek açıdan elde edilebilecek olan iskelet yapılarının, 360 derece tam bir taramaya tamamlanması için yapılacak, 3B dönüştürme, hizalama gibi işlemlerden soyutlamaktır. Bu çalışmada, gerçek hayatta kullanılan 3B gerçek nesnelere ait ayırt etme işlemi için nokta bulutları üzerinden iskelet eğrisi çıkarma algoritması yürütülmektedir. Elde edilen iskelet yapısı sayesinde gürültülü giriş veya



karmaşık geometriye sahip nesneler ile başa çıkabilen nesne ayırt etme tekniğine bir yöntem sunulmaktadır.

## **1.2. Tezin İçeriği**

Tez çalışmasının içeriğindeki ilk adım, gerçek fiziksel nesnelerin modellerinin taranmasında kullanılacak yöntemlerin belirlenmesidir. Bunun için çalışmanın başlarında farklı nesne tarama yöntemleri incelenmiş, bize en uygun olan Kinect v1 algılayıcı kameranın kullanıldığı kısmen ToF (Time-of-Flight) ve yapısal ışık yöntemini birlikte kullanan yöntem tercih edilmiştir. İkinci bölümde, geniş kapsamlı bir tarama yöntemi incelemesi yapılmıştır. Üçüncü bölümde, tarama yöntemi için uygulanabilecek derinlik hesaplama yöntemleri üzerinde durulmuştur. 3B sayısal nokta bulutu verilerinin üretilmesi için kullanılan yöntemlerden ve çeşitli çalışmalarda, faydalandığı düşünülen algoritmaların çalışma prensiplerinden bahsedilmiştir. Dördüncü bölümde uygulama olarak geliştirdiğimiz tam bir 360 derece nokta bulutu üreten sistem tanıtılmaktadır. Bu kısım sağlıklı iskelet yapısı oluşturmak adına önem taşımaktadır. Mevcut çalışmalarda faydalanılan birçok algorithmadan bağımsız olarak sunulan tarama sisteminde gerçekleştirilen 3B dönüştürme, hizalama ve nokta bulutlarını iyileştirme süreçlerinden bahsedilmektedir. Beşinci bölümde tam bir tarama ile elde edilen nokta bulutu kayıtları ile nesnelerin, bulunduğu sahneden ayırt edilmesi için kullanılan iskelet çıkarım algoritmalarından bahsedilmektedir.

## 2. NESNE TARAMA YÖNTEMLERİ

3B tarayıcılar gerçek dünyadaki fiziksel nesnelerin veya içerisinde bulundukları ortamın muhtemel şekli ve yapısı ile ilgili veri toplamak için kullanılan cihazlardır. Elde edilen veriler ile dijital üç boyutlu modeller oluşturmak, bu modellerle ilgili çıkarımlarda bulunmak ya da veriler üzerinden analiz etme işlemleri yapılabilmektedir.

Günümüzde gerçekleştirilen birçok çalışmada kullanılan 3B tarama cihazları bünyesinde çok farklı teknolojiler barındırmaktadır. Kullanılan tüm bu teknolojiler, kendi sınırlamalarını, avantajlarını, dezavantajlarını ve maliyetlerini beraberinde getirmektedir. 3B sayısallaştırmada örneğin optik teknolojisinden olduğu gibi nesnelerin türüne bağlı olarak, parlak, yansıyan yüzey ya da saydam nesneler şeklinde birçok sınırlama ile karşılaşılabilir.

Elde edilen 3B verilerden birçok farklı alanda gerçekleştirilen uygulamalarda yararlanılmaktadır. Kullanılan bu teknolojilere, sanal gerçeklik de dâhil olmak üzere, film ve video oyunlarının üretiminde ve eğlence endüstrisinde yaygın olarak başvurulmaktadır. Teknolojilerin bir diğer yaygın kullanım alanlarının başlıcaları arasında, endüstriyel tasarım, ortez, protez, tersine mühendislik, prototipleme, kalite kontrol, inceleme ve kültürel miras eserlerinin sayısallaştırılması gelmektedir [9].

3B tarayıcı sistemlerin genel amacı üç boyutlu birer model üretmektir. Bu 3B modeller, nesnelerin yüzeyindeki geometrik örneklerinin birer nokta bulutundan meydana gelmektedir. Bu 3B nokta bulutu yapısı farklı alanlardaki birçok çalışmada kullanılmaktadır. Örneğin, taranan nesnenin 3B şeklini tahmin etmek için kullanılabilmektedir (yeniden yapılandırma). Bu aşamada bulut yapısındaki her bir noktaya ait renk bilgisi de eklendiğinde, nesne yüzeyindeki renkler de belirlenebilmektedir.

3B tarayıcılar ile günlük kameralar arasındaki birkaç özellik ortak olarak paylaşılmaktadır. 3B tarayıcılar, çoğu kamerada olduğu gibi, koni benzeri bir görüş alanına sahiptirler, aynı şekilde sadece gizlenmeyen (tıkantısız) yüzeyler hakkında bilgi toplayabilmektedirler. Bu bakımdan, bir kamera kendi görüş açısı içerisindeki yüzeyler hakkında bilgi toplarken, bir 3B tarayıcı kendi görüş açısı içerisindeki yüzeyler hakkında mesafe bilgisi toplamaktadır. Bir 3B tarayıcı tarafından üretilen bir "derinlik resmi", resmin her bir noktasındaki bir yüzeye olan mesafeyi tanımlamaktadır. Bu şekilde resimde

yer alan her bir noktanın uzayda kapladığı konum itibariyle tanımlanabilmesine olanak sağlamaktadır.

Birçok durumda, tek bir tarama işlemi vasıtasıyla 3B nesnelerin tam bir modelini üretebilecek kapasitede bir sistem tasarlanmamıştır. Nesnelerin 360 derecelik yapısına ait bilgiler edinmek için genellikle nesnelerin tüm açıları ile çok sayıda tarama işlemi yerine getirilmek zorundadır. Bu çok sayıda tarama işlemleri, bir 3B model üretebilmek açısından genellikle hizalama ve kayıt olarak adlandırılan yöntemler ile birleştirilmektedir. Neticede haritalama olarak adlandırılan her bir taramadan, bütün bir 3B modele doğru giden bu süreç, genellikle 3B tarama olarak bilinmektedir [10].

Günümüzde 3B bir nesnenin şeklini sayısal olarak elde etmek için çeşitli teknolojiler bulunmaktadır. Temel olarak bu teknolojiler, temaslı ve temassız olmak üzere iki ayrı sınıfta değerlendirilmektedir [11]. Temassız çözümler de kendi içerisinde aktif ve pasif olmak üzere iki ayrı kategoriye ayrılmaktadır. Her bir kategori kendi içerisinde farklı teknolojileri barındırmaktadır. 3B nesnelerin ve içinde bulundukları sahnelerin üç boyutlu yapısına ait verilerinin elde edilmesi amacıyla kullanılacak olan yöntemin belirlenmesi sırasında tercih edilebilecek birçok yöntem, maliyeti, kullanılabilirliği ve geliştirilebilirliği açısından değerlendirilmiştir. 3B nesnelerin ayırt edilmesi için kullanılacak tarama yöntemlerinin eksiklik ve üstünlükleri arasında yapılan değerlendirmeler neticesinde, ideal bir sistem tasarlanmıştır. Bu bölümde 3B nesnelere ait verilerinin toplanmasında literatürde kullanılan mevcut yöntemler ile ilgili değerlendirmeler yapılmaktadır.

## 2.1. Temaslı Yöntemler



Şekil 2.1. Sert dik bir kola sahip bir koordinat ölçme makinesi [12].

Temaslı 3B tarayıcılar, nesnelerin, tarayıcı üzerindeki plakasıyla temas halinde üzerinde dururken, nesne yüzeyine tarayıcının temas ucu ile fiziksel bir dokunuş sağlanarak yapılan ölçüm işlemlerinde kullanılmaktadır. Taranacak nesnenin düzlemsel olmadığı veya düz bir yüzeye sabit olarak oturamayacağı durumlarda, bir sabitleme tertibatı ile sağlam bir şekilde desteklenmekte ve tutulmaktadır. Şekil 2.1’de sert dik bir kola sahip örnek bir koordinat ölçüm makinesi gösterilmektedir [12].

Bu tarayıcı mekanizmaları üç farklı biçimde çalışma şekline sahip olabilmektedir;

- Katı kollara sahip bir taşıyıcı sistemi, nesne ve tarayıcı arasındaki dikey ilişkide sıkıca tutulur ve her eksen bir ray boyunca süzülür. Bu tür sistemlerde düz profil şekilleri veya basit dışbükey kavimsel yüzeyler ile en iyi şekilde çalışabilmektedir.
- Katı uzuvlara sahip, yüksek hassasiyetli açısal sensörler yardımıyla oluşturulmuş eklemli bir kol barındıran sistem. Bu sistem, kolun ucunun konumunu, her eklem bilek dönüş açısını ve menteşelerinin açılarının birlikte hesaplandığı karmaşık bir matematik içermektedir. Bu sistem özellikle dar bir ağız açıklığı, çatlaklar veya iç alanlarda sayısallaştırma işlemleri için idealdir.
- Seyahat vagonunda asılı duran mafsallı bir kol yapısına benzeyen, her iki yöntemin bir kombinasyonu olan, büyük nesnelerin iç yapısındaki boşluklarını ya da üst üste binen (tıkalı) yüzeyleri eşleştirebilen bir sistemdir.

Bir KÖM (Koordinat Ölçme Makinesi), diğer bir adıyla sayısallaştırıcı, temaslı bir 3B tarayıcıya örnektir. Bu yapıdaki sistemler genellikle imalat sektöründe kullanılmaktadır ve çok hassas olabilmektedirler. KÖM'lerin en büyük dezavantajları taranacak 3B nesne ile temas halinde olmaları gerektirir. Her türden nesneye uygulanamayan bu sistem, tarama eylemi sırasında sistem nesnenin yapısını değiştirebilir veya ona zarar verebilmektedir. Bu durum özellikle gerçek tarihi eserler gibi hassas veya değerli objelerin taranması sürecinde çok önemli bir husustur. KÖM'lerin bir diğer dezavantajı ise diğer tarama yöntemlerine kıyasla nispeten daha yavaş olmalarıdır. Sayısallaştırıcı ucun (probun) üzerine monte edildiği kolu, fiziksel olarak hareket ettirmek çok yavaş olabilmekte ve en hızlı KÖM'ler dahi sadece birkaç yüz Hz üzerinde çalışabilmektedir. Bunun aksine, bir lazer tarayıcı gibi optik bir sistem 10 ile 500 kHz arasında çalışabilmektedir [13].

## **2.2. Temasız Aktif Yöntemler**

Bu sistemler, 3B tarama alanındaki yakalanabilecek detay seviyesine göre özellikle temasız ölçüm gerektiren çok küçük nesneleri ölçmek için tercih edilen yöntemlerdir.

Temaslı 3B tarayıcılar, nesne yüzeydeki her nokta ile temas eden bir cihazla nesneyi fiziksel olarak tarayarak ölçüm verilerini toplarken, temassız 3B tarayıcılar, nesnenin geometrik yapısını değiştirmeden hızlıca ve çok miktarda veri toplayabilmektedirler. Bu aynı zamanda nano ölçekte ölçümlerin toplanması için bir avantaj sağlamaktadır. Temassız yöntemler kendi içerisinde aktif ve pasif olarak iki farklı kategoriye ayrılmıştır. Aktif tarayıcılar bir çeşit radyasyon veya ışık yaymaktadırlar ve bir nesneyi veya ortamı araştırmak için nesneden geçen yansımayı veya radyasyonu algılamaktadırlar. Kullanılabilecek emisyon türleri arasında ışık, ultrason veya röntgen bulunmaktadır.

### 2.2.1. Uçuş Zamanı (Time-of-Flight) Yöntemi



Şekil 2.2. Bir ToF kamera (Microsoft Kinect Xbox One)

ToF ismiyle adlandırılan 3B lazer tarayıcılar, literatüre Türkçe’de uçuş zamanı olarak geçmiştir. Bu tarayıcılar nesnelerin taranması için lazer ışığı teknolojisini kullanan aktif birer tarayıcıdır. Bu tarz tarayıcıların merkezinde uçuş süresini hesaplama yeteneğinde bir lazer mesafe bulucu bulunmaktadır. Lazer telemetre, bir ışığın nabız atış zamanı üzerinden hesaplama yaparak bir yüzeyin mesafesini bulabilmektedir. Burada bir ışık nabzını yaymak için bir lazer kullanılmaktadır ve yüzeyden yansıyan ışığın sensör üzerindeki bir detektör tarafından görülmesinden önceki zaman miktarı ölçülmektedir [14]. Şekil 2.2’de bir ToF kamera özelliği taşıyan Microsoft Kinect Xbox One sensörü gösterilmiştir.

ToF, genel anlamda bir kaynaktan yayılan tek darbelik lazer veya kızılötesi ışınların gidiş ve dönüş sürelerini hesaplayarak, kaynağın nesneye olan uzaklığıyla ilgili bilgi verilebilen temassız etkin bir yöntem olarak ifade edilebilir. ToF yöntemine ait mesafe hesaplama işlemleri formüsel olarak ifade edilirse, ışık hızı  $c$  bilinmektedir, tarayıcı ile nesne yüzeyi arasındaki mesafenin iki katı, ışığın toplam gidiş-dönüş süresi olarak hareket mesafesini belirtmektedir. Buradan eğer  $t$  gidiş-dönüş süresi olarak belirlenirse mesafe

$c.t/2$  kadar olmaktadır. 3B lazer tarayıcının uçuş süresi doğruluğu  $t$  zamanın ne derece hassas ölçüldüğü ile ilişkilidir. Bu bakımdan, ışığın 1mm hareket etmesi için geçen süre yaklaşık 3.3 piko saniyedir.

Lazer telemetre, sadece görüş açısındaki bir noktanın mesafesini algılayabilmektedir. Bu bakımdan tarayıcının görüş açısı değiştirilerek, tüm görüş alanı içerisindeki her bir farklı nokta her defasında taranabilmektedir. Tarayıcı cihazın görüş yönü, telemetrenin kendisinin döndürülmesiyle ya da bir döner ayna sistemi kullanılarak değiştirilebilmektedir. Genellikle ikinci yöntem tercih edilmektedir çünkü aynalar daha hafiftir dolayısıyla daha hızlı ve daha fazla hassasiyetle döndürülebilmektedir. Bilinen ToF 3B lazer tarayıcılar, her saniyede 10.000 ~ 100.000 nokta arasındaki mesafeyi ölçebilme kabiliyetindedir. Bu ölçüm, bir ortam aracılığıyla belli bir zaman standardı kullanarak hızı veya yol uzunluğunu ölçmek için bir yöntem olarak sunulmaktadır. Böylece hareket eden nesne doğrudan veya dolaylı olarak algılanabilmektedir.

Bir ToF kamera, aşağıdaki bileşenlerden oluşur;

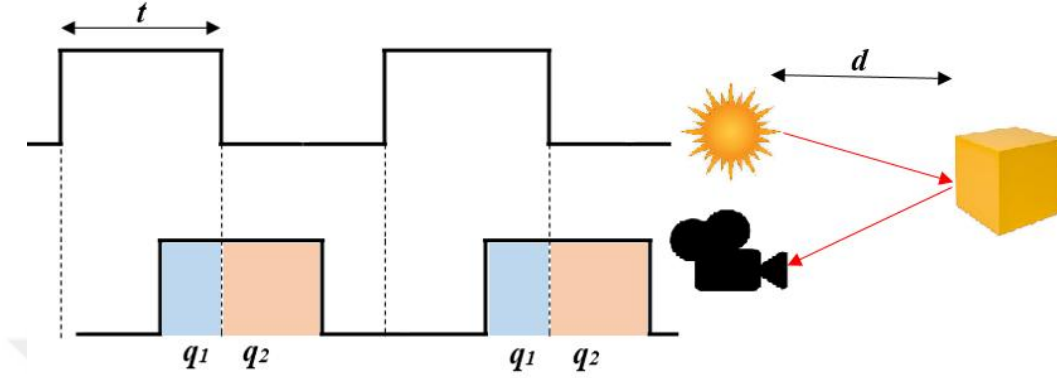
**Aydınlatma ünitesi:** Sahneyi aydınlatmada kullanılmaktadır. Işık, RF modülasyonlu ışık kaynakları için faz detektörü görüntüleyicileri ile 100 MHz'ye kadar yüksek hızlara ulaşabilmektedir, sadece LED'ler veya lazer diyotları gözle görülebilmektedir. ToF görüntüleyiciler için, doğrudan çerçeve başına tek bir darbe (ör. 30Hz) kullanılmaktadır. Aydınlatma, göze çarpmayacak hale getirilmesi için normalde kızılötesi ışık kullanılmaktadır.

**Optik:** Tarayıcı içerisindeki bir lens yansıyan ışığı toplamakta ve çevreyi görüntü sensörüne (odak düzlemi dizisi) aktarmaktadır. Bir optik bant geçiren filtre, ışığı sadece aydınlatma birimiyle aynı dalga boyunda geçirmektedir. Bu durum, uygun olmayan ışığı bastırmaya ve gürültüyü azaltmaya yardımcı olmaktadır.

**Görüntü sensörü:** Bu kısım, ToF kameranın merkezidir. Her pikselde, ışığın aydınlatma ünitesinden (lazer veya LED) nesneye ve odak düzlem dizisine geri dönme süresi ölçülmektedir. Zamanlama için farklı yaklaşımlar kullanılmaktadır.

**Sürücü elektroniği:** Hem aydınlatma ünitesi hem de görüntü sensörü yüksek hızlı sinyaller ile kontrol ve senkronize edilmek zorundadır. Yüksek bir çözünürlüğe sahip olmak için bu sinyallerin doğruluğunun yüksek olması gerekmektedir. Örneğin, aydınlatma birimi ile sensör arasındaki sinyaller sadece 10 pikosaniye değişirse, mesafe 1,5 mm'ye kadar değişebilmektedir.

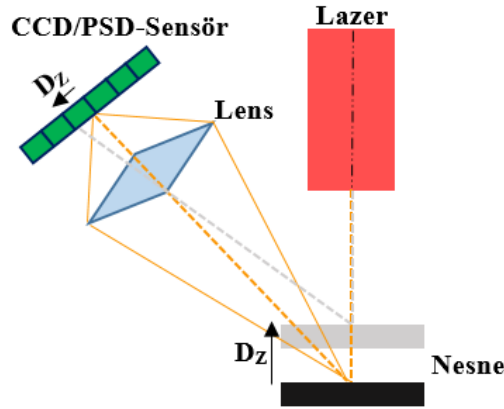
**Hesaplama ve Arayüz:** Mesafe doğrudan tarayıcı içerisinde hesaplanmaktadır. Yüksek performans elde etmek için bazı ölçümleme verileri kullanılmaktadır. Tarayıcı ile USB veya Ethernet gibi bazı arabirimler üzerinde bir mesafe görüntüsü alınabilmektedir.



Şekil 2.3. ToF Kameranın çalışma prensibi

Şekil 2.3'te grafik olarak gösterilen ToF kameraların çalışma prensibi olarak, mesafe,  $d = c \cdot t / 2$ ,  $q_2 / q_1 + q_2$ , dir. Burada  $c$  ışık hızını,  $t$  ise ışık uzunluğunu,  $q_1$  ışık yayıldığında pikselde biriken yük ve  $q_2$ , ışık olmadığı zaman biriken yüküdür.

### 2.2.2. Lazer Tarama Yöntemi



Şekil 2.4. Lazer üçgenleme sensörünün çalışma prensibi [15].

Üçgenleme tabanlı 3B lazer tarayıcılar, bulunduğu çevreyi araştırmak için lazer ışığı kullanan aktif tarayıcılardır. ToF'a nazaran, 3B lazer tarayıcılarda, üçgenleme lazeri üzerinde bulunan bir lazer ile ışın yayılmakta ve lazer noktasının yerini bulmak için bir kameradan yararlanmaktadır. Lazerin karşısındaki bir yüzeye ne kadar uzak olduğuna bağlı

olarak, lazer noktası kameranın görüş alanındaki farklı yerlerde görünebilmektedir. Bu tekniğe üçgenleme denilmektedir. Çünkü bu teknikte lazer nokta, kamera ve lazer yayıcı bir üçgen oluşturmaktadır. Kamera ile lazer yayıcı arasındaki mesafe olan üçgenin bir tarafının uzunluğu bilinmektedir. Lazer vericiye ait köşesinin açısı da bilinmektedir. Kamera köşesinin açısı, kameranın görüş alanındaki lazer noktasının konumuna bakılarak belirlenebilmektedir. Bu üç bilgi parçası, üçgenin şeklini ve boyutunu tam olarak belirlemekte ve üçgenin lazer nokta köşesinin yerini vermektedir. Çoğu durumda, tek bir lazer nokta yerine, bir lazer şeridi, nesnenin üzerinden geçirilerek veri toplama işlemi hızlandırılmaktadır. Şekil 2.4'te bir lazer üçgenleme sensörünün çalışma prensibi grafiksel olarak gösterilmektedir.

ToF ve üçgenleme lazer tarayıcılarının her birinin farklı durumlar için uygun olmalarını sağlayan güçlü ve zayıf yönleri bulunmaktadır. ToF tarayıcılarının en önemli avantajı, kilometrelerce uzaklıktaki mesafelerde çalışabilmeleridir. Bu tarayıcılar, binalar veya coğrafik özellikler gibi büyük yapıları taramak için uygundur. ToF tarayıcılarının en büyük dezavantajı doğruluklarıdır. Yüksek ışık hızı nedeniyle, gidiş-dönüş süresinin zamanlaması zordur, bu nedenle mesafe ölçümünün doğruluğu nispeten düşüktür. Mesafe ölçümünün doğruluğu milimetreler seviyesindedir.

Üçgenleme tarayıcılarında bu durum tam tersidir. Yani birkaç metre ile sınırlı bir menzile sahiptirler ancak doğrulukları nispeten yüksektir. Mesafe ölçümünün doğruluğu onlarca mikrometre seviyesindedir.

ToF tarayıcılarının doğruluğu, lazerin bir nesnenin kenar noktasına ulaşması gibi durumlarda kaybolabilir, çünkü tarayıcıya gönderilen bilgiler bir lazer darbesi için iki farklı konumdan gelmektedir. Bu tarayıcının pozisyonuna göre elde edilen gerçek koordinat değeri, nesnenin kenar noktasına çarpan bir nokta için bir ortalamaya göre hesaplanmaktadır. Bu bakımdan nokta yanlış bir konuma atanmış olmaktadır. Bir nesne taramasında yüksek çözünürlük kullanıldığında, ışının bir kenar noktaya çarpma ihtimali artmakta ve sonuçta elde edilen veriler nesnenin kenarlarının arkasındaki gürültüyü göstermektedir. Daha küçük bir giriş genişliğine sahip tarayıcılar, bu problemi çözmeye yardımcı olabilmektedir, ancak ışın genişliği mesafeden dolayı artacağından menzile ile sınırlandırılacaktır.

Düşük çözünürlüklü taramalar saniyede 10.000 örnekleme hızında, bir saniyeden daha kısa sürebilirken, milyonlarca örnek gerektiren yüksek çözünürlüklü taramalar, bazı ToF tarayıcılar için dakikalarca sürebilmektedir. Bu durumun yarattığı problem hareketten



kaynaklanan bozulmalardır. Her nokta farklı bir zamanda örneklendiğinden, nesne veya tarayıcıdaki herhangi bir hareket toplanan verileri bozabilmektedir. Bu nedenle, hem nesneyi hem de tarayıcıyı sabit platformlara monte etmek ve titreşimi en aza indirmek genellikle zorunluluktur. Hareket halindeki nesneleri taramak için bu tarayıcıları kullanmak neredeyse imkânsızdır.

Lazer tarama teknikleri son yıllarda 3B modelleme, bilimsel araştırma, inceleme ve uygulama alanlarında gelişme göstererek, oldukça popüler bir yöntem olarak kullanılmaya başlanmıştır [16]. Bu yöntemin popülerlik kazanmasında yatan temel neden, bu tarama tekniğinin geleneksel tekniklerden daha hızlı veri toplayarak sayısallaştırma işlemlerini kısaltmasıdır. Lazer ile tarama yapmanın en önemli avantajı, yapılacak olan işlemin hızlı ve dokunma olmadan gerçekleştirilmesidir. Lazer ışınlarının yüksek çözünürlük ve inceliği sayesinde, taranan nesnenin her bir yüzeyinin hassas bir şekilde detaylı olarak verilerin alınması sağlanabilir. Bu sayede mühendislik yapıları, otomotiv sektörü, tarihi binalar gibi alanlardaki objelerin hızlı ve etkin bir biçimde ölçülmesi sağlanmaktadır. Bunun dışında insan gözünün algılama sınırları dışına çıkan bazı durumların incelenmesi gerektiğinde kullanılan lazer ışığı sayesinde bu unsurlar belirginleştirilebilir. En önemlisi bir lazer ışık kaynağı ve sayısal kamera kullanılarak kaydedilmiş olan görüntülerden çok kısa bir süre içerisinde 3B modelleme yapılabilir [17].

### 2.2.3. Yapısal Işık Yöntemi

Yapısal ışık yöntemini kullanan 3B tarayıcılar, nesne üzerine yansıttığı bir ışık paterni ile nesne üzerindeki desenin deformasyonunu gözlemlemektedir. Desen, LCD projektör veya bir başka sabit ışık kaynağı kullanılarak nesne üzerine yansıtılmaktadır. Desen projektöründen biraz uzakta duran bir kamera, modelin şekline bakar ve görüş alanındaki her bir noktanın mesafesini hesaplamaktadır.



Şekil 2.5. Yapısal ışık yöntemi kullanılarak oluşturulan 3B model [18].

Yöntemin çalışma prensibinde, ışık üç-boyutlu şekilli bir yüzey üzerine, dar bir bant boyunca gönderilmektedir. Yansıyan ışık görünümü bir yansıma yüzeyi üzerinde farklı açılardan bozuk görünebilen bir aydınlatma çizgisi üretmektedir ve yüzeyin şekli (açık bölüm) ile tam bir geometrik yeniden yapılandırma elde edilebilmektedir. Bu yöntem, aynı zamanda örneklerin çok sayıda edinimine izin verdiğinden dolayı, tek seferde daha hızlı ve daha çok yönlü birçok çizgiden oluşan desen yansıtılmasına olanak sağlamaktadır. Ancak farklı açılardan bakıldığında, geometrik görüntü, nesnenin yüzey şekline bağlı olarak bozuk görünebilmektedir. Şekil 2.5'te yapısal ışık yöntemi kullanılarak oluşturulan örnek bir 3B model gösterilmektedir.

Renkli bir örüntü ve beyaz ışık arasında hızlı bir geçiş sağlanarak da yoğunluk resmi ile senkronize olmuş bir görüntü elde edilebilmektedir. Fakat elde edilen görüntülerde, bazı noktaların kaybolduğu delikler ile birlikte yansımanın fazla (gözlük v.b.) ve az (sakal, bıyık v.b.) olduğu bölümlerde küçük kalıntılar da oluşabilmektedir.

3B nesnelerin yüzey verilerini elde etmek için gerçek zamanlı veya yarı çevrimiçi şekilde çalışan, yapısal ışık kullanan sistemlerin birçoğu genellikle renkli örüntü olmak üzere sadece tek bir örüntü kullanmaktadır [19, 20]. Bu yöntemler, daha hızlı yakalama süresi için düşük doğrulukta çalışmaktadır. Diğer sistemler ise ikili olarak kodlanmış birçok örüntüyü hızlı bir şekilde değiştirerek kısa bir süre içerisinde yakalama işlemini gerçekleştirmektedir [21, 22]. Bu yaklaşımlar ile ilgili temel sorun, şerit genişliğinin bir pikselden büyük olması nedeniyle elde edilen görüntünün çözünürlüğünün düşük olmasıdır. Ayrıca örüntülerin hızlı bir şekilde projektöre yüklenmesiyle birlikte örüntülerin değişim süresi de yakalama sürecinin içine dâhil olmakta ve bu da birim saniyede elde edilen kare sayısını olumsuz etkilemektedir.

Yapısal ışıklı 3B tarayıcıların en önemli avantajlarından ikisi, hız ve kesinliktir. Başka yöntemlerde olduğu gibi her seferinde bir nokta taramak yerine, yapılandırılmış ışık tarayıcıları birden fazla noktayı ya da tüm görüş alanını bir defada tarayabilmektedir. Bütün bir görüş alanının tamamını saniyeden daha az bir sürede taramak, bozulmaların yayılmasını azaltmakta veya ortadan kaldırmaktadır. Bazı mevcut sistemler, hareketli nesneleri gerçek zamanlı olarak tarayabilmektedir. VisionMaster, 5 megapiksel kameralı bir 3B tarama sistemi oluşturmaktadır ve her bir karede 5 milyon veri noktası elde edebilmektedir.

Yapısal ışık yöntemi ile 3B tarama işlemi, her yıl yayınlanan birçok araştırma makalesi ile hala çok aktif bir araştırma konusudur. Yapısal ışık düzeninin, mükemmel

haritalar oluřturmada, hata tespiti ve dűzeltmelerinde olduka yararlı olduėuna alıřmalarda deėinilmiřtir [23].

Sayısal saak projeksiyonu ve faz-kaydırma tekniėi (farklı tűrde yapısal ıřık yűntemleri) kullanılarak gerek zamanlı bir tarayıcı geliřtirilmiřtir. Bu sistemle, dinamik olarak deforme olabilen 3B nesnelere ait, yűz ifadesi gibi yűksek yoėunluklu ayrıntılar, saniyede 40 kareye kadar yakalanabilmektedir [24]. Son dűnemlerde geliřtirilen bir bařka tarayıcı ile sisteme farklı desenler uygulanarak, yakalama ve veri iřleme iin her bir kare hızı saniyede 120 kareye ulařtırılmıřtır. Bu sistem ikili defocusing tekniėini kullanarak, saniyede yűzlerce hatta binlerce kareye ulařabilen hız atılımları yapılmıřtır [25].

Yapısal ıřık yűnteminin farklı bir modeli olan modűle ıřık yapısı kullanan 3B tarayıcılar nesne űzerine sűrekli deėiřen bir ıřık samaktadır. Genellikle ıřık kaynaėı, genliėini sadece sinűzoidal bir desende dűndűrmektedir. Tarayıcı űzerindeki bir kamera yansıyan ıřıėı algılamaktadır ve desenin kaydıėı miktar doėrultusunda, ıřıėın kat ettiėi mesafeyi belirlemektedir. Modűle ıřık, 3B tarayıcının lazer dıřındaki kaynaklardan gelen ıřıėı gűrmezden gelmesine de olanak saėlamaktadır, dolayısıyla bu konuda herhangi bir giriřimde bulunmamaktadır.

#### **2.2.4. Konoskopik Holografi Yűntemi**

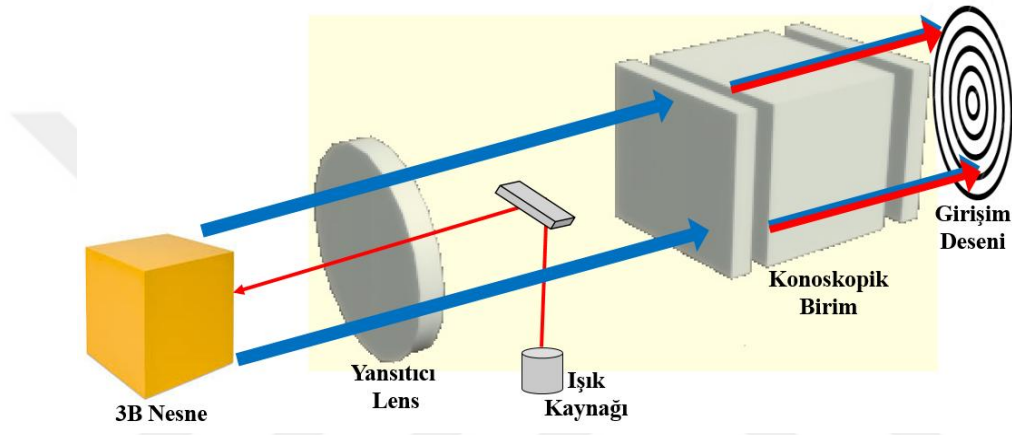
Klasik holografide, 3B nesne ıřını ile uyumlu bir ıřık kaynaėı kullanılarak referans kiriři arasında oluřan bir giriřim paterninin kaydedilmesiyle bir hologram oluřturulmaktadır. Aynı kırılma indisine sahip iki ıřın aynı hızda oėalmaktadır, ancak farklı geometrik yolları takip etmektedirler. Bu ıřınlar, űst űste geldiėinde, iki ıřın arasındaki faz farkının sadece geometrik yol farkına baėlı olduėu anlamına gelmektedir. Bu faz farkı, daha sonra orijinal ıřık alanını yeniden oluřturmak iin kullanılabilir, ۆlűlebilir bir giriřim modelinin yaratılmasından sorumlu olacaktır [26].

Konoskopik holografide ise anizotropik optik kristalinden geen bir ıřık ıřını, aynı geometrik yolu paylařan fakat farklı ortogonal polarizasyon modlarına sahip olan iki ıřına ayrılmaktadır. Bu iki kiriřin kırılma indisleri genellikle birbirinden farklıdır. Bu nedenle, iki ıřın kristalden ıktıktan sonra bir giriřim modeli oluřturmaktadır. Bu modelin ۆzellikleri ıřıėın kaynaėına olan mesafesine baėlıdır.

Konoskopik sistemde, bir lazer ıřını nesne yűzeyine yansıtılmaktadır ve hemen sonrasında aynı ıřın gűzergaėı boyunca yansıtılarak CCD (Charge-coupled device) űzerindeki bir Konoskopik kristal yardımıyla tahmin iřlemi gerekleřtirilmektedir. CCD

sensörlerin temel görevi ışık kaynağından gelen fotonları yakalamaktır. Sonuçta ölçülen yüzeye olan mesafeyi belirlemek için frekans analizi yapılabilen bir kırınım modeli tasarlanmıştır. Konoskopik holografinin ile temel avantajı sadece tek ışın yollu çalışarak ölçüm sağlamasıdır.

Konoskopik holografide her iki kırış aynı geometrik yol boyunca yayıldığı için interferometri bazlı ölçüm tekniklerine kıyasla oldukça stabildir. Ayrıca, bu teknikte tutarsız ışık kullanarak ölçüm yapmak da mümkündür.



Şekil 2.6. Konoskopik holografi sensörünün çalışma prensibi

Bu yöntemde sensör, objektif bir merceğe odaklanan ve ölçülmekte olan nesneye çarpan, göze zararsız bir lazer ışını yaymaktadır. Dağınık ışığın bir kısmı, nesneden sensöre geri döner ve optik olarak anizotropik kristal içeren konoskopik birime yönelir. Böylece ortaya çıkan girişim deseni tespit edilmektedir ve daha sonra ölçülen verilerden mesafe bilgisini almak için sinyal işleme algoritmaları kullanılmaktadır. Şekil 2.6'da konoskopik holografi sensörüne ait çalışma prensibi grafiksel olarak gösterilmektedir.

### 2.3. Temassız Pasif Yöntemler

Temassız pasif 3B görüntüleme çözümlerinde herhangi bir radyasyon yayılımı söz konusu değildir, bunun yerine yansıyan ortam radyasyonunu tespit etmeye dayanmaktadır. Bu mantığa sahip çoğu çözüm görünür ışığı algılamaktadır, çünkü ortamdaki mevcut radyasyona kolayca erişilebilmektedir. Ayrıca sistemde kızılötesi gibi diğer radyasyon türleri de kullanılabilir. Pasif yöntemler çok uygun maliyetle tedarik edilebilmektedir çünkü çoğu durumda basit dijital fotoğraf makineleri dışında özel donanımlara gereksinim duymamaktadırlar.

Genel olarak temassız pasif 3B tarayıcılar arasında stereoskopik, fotometrik, siluet tabanlı ve görüntü tabanlı modelleme tarayıcıları bulunmaktadır.

- **Stereoskopik:** Stereoskopik kameralar genellikle birbirine yakın şekilde kalibre edilmiş iki tane RGB kamera barındırmaktadır. Bu kameralar günümüzde sadece tüketici seviyesi mobil cihazlarda kullanılabilir halde olmasına rağmen, stereo görünümü bir derinlik haritasının nasıl oluşturulması gerektiği konusundaki araştırma alanında büyük ilgi görmektedir.

Stereoskopik sistemler, genellikle aynı sahneye bakan, ancak birbirinden bağımsız iki farklı video kamera kullanmaktadır. Bağımsız her bir video kamera tarafından yakalanan görüntüler arasındaki küçük farklılıklar analiz edilerek, görüntülerin her bir noktası üzerindeki mesafe belirlenmektedir. Bu yöntem, insan gözüne ait stereoskopik vizyonu taşıyan ve yönlendiren benzer prensiplere dayandırılmaktadır [27]. Çoğu stereoskopik yöntemde izleyicinin sağ ve sol gözü için iki ayrı ofset görüntü sunulmaktadır. Bu iki-boyutlu her bir görüntü daha sonra 3B derinlik algısı vermek için beyinde birleştirilmektedir. Bu teknikle gözlemcinin baş ve göz hareketleri ile görüntülen 3B nesneler hakkındaki bilgilerin artırılabilmesine ve tam üç boyutlu bir resim görüntülenmesine olanak sağlanmaktadır.

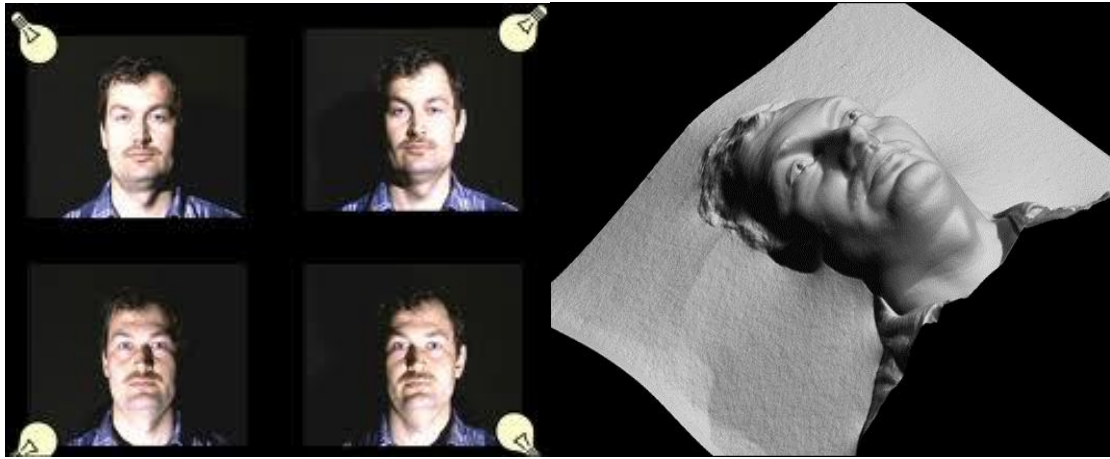
Bu yöntem ile yakalanacak sahnenin farklı açılardan alınan görüntülerinin yardımıyla ilişkili noktalar bulunmakta, nesne sınırları belirlenmekte ve bu veriler yardımıyla yeniden yapılandırılma işlemi gerçekleştirilmektedir. Stereoskopik tabanlı bir çalışmada yüksek kaliteli bir 3B nesne yüzey yakalama yönteminden bahsedilmektedir [28]. Bu gibi çalışmalarda sahnelerin yakalanması işlemi için hem sıradan hem de detaylı yüzey geometrisi elde etmek için son teknoloji kameralar kullanılmaktadır.

Stereoskopik yöntemlerin avantajı görüntü yakalama sırasında belirli aralıklarla patlayan flaşlara maruz kalınmaması ve düzenek üzerindeki kameraların sabit, tek bir ışık kaynağından faydalanılarak sahneyi eş zamanlı yakalayabilmeleridir. Ancak yöntemin birden fazla sayıda algılayıcı kameraya duyduğu ihtiyaç nedeniyle, bu yöntemi fotometrik stereo yönteminden daha maliyetli bir hale getirebilmektedir. Diğer yöntemlerde olduğu bu yöntemde de şeffaf yapılar ve ışık yansımalarından dolayı parlayan nesnelerin yakalanmasında sıkıntılar yaşanmaktadır. Ayrıca yüz gibi pürüzsüz ve düzlemsel yüzeylerin bu yöntem ile yeniden oluşturulabilmeleri zorlaşmakta ve kayıt esnasında yüzeyin yer değişiminin fazla miktarda olmaması gerekmektedir. Nesnelerin 3B yeniden yapılandırılma işlemlerinin çok fazla hesapsal karmaşıklık içermesi, bu tarz sistemlerin

çevrimiçi olarak kullanılmalarındaki en büyük engeldir. Gerçekleştirilmiş bazı algoritmalarda, girdilerden 3B nesne modellerinin yeniden yapılandırılması için yaklaşık 30 dakika gibi bir zamana ihtiyaç duyulmaktadır [28]. DI4D (Dimensional Imaging 4 Diemension) sistemi için ise bu süre 15 saniye/kare civarında gerçekleşmektedir.

Yöntemin kullanımının yaygınlaşması ile nesnelere ait 3B yüzey verilerinin elde edilmesi için çeşitli ticari yazılımlar da kullanılmıştır. Bunlardan birkaçı olan DI4D [29] dinamik yüzey verisi elde etme sistemi ve 3DMD [30] (3 Dimensional Multi Dynamic) çoklu dinamik 3B stereo sistemi, nesnelere ait 3B bilgileri için veri tabanlarını oluşturulmasında kullanılmışlardır [31, 32].

• **Fotometrik:** Fotometrik sistemler genellikle tek bir kamera kullanmaktadır, ancak değişen aydınlatma koşullarında birden fazla görüntü yakalayabilmektedir. Bu teknikle, her bir pikselde yüzey yönelimini geri kazanmak için görüntü oluşturma modelini tersine çevirmeye çalışılmaktadır. Yöntemin temel mantığı sabit bir şekilde duran bir nesnenin farklı açıdan yüzeylerine düşürülen ışıkların normallerinin hesaplanması şeklindedir [33]. Şekil 2.7’de insan silueti etrafına yerleştirilmiş dört adet standart ışık kaynağı ile oluşturulan fotometrik ve çoklu spektrumlu fotometrik stereo görüntüler gösterilmektedir.



Şekil 2.7. Fotometrik yöntem ile elde edilen 3B yüz modeli [33].

Bu yöntem nesnelerin içerisinde bulundukları ortamın ışık şiddetine, nesnelerin şeffaflık ve parlaklıklarına ve de düzensiz aydınlatmalara karşı oldukça hassas bir yapıdadır. Bu yöntem sonucunda elde edilen modeller, nesne yüzeyinin mesh yapısından ziyade o nesneye ait yüzeyin normallerinden oluşmaktadır. Bundan dolayı nesne yüzeyine ait bilgilerin çıkarılması için elde edilmiş yüzey normallerinin uygun mesh yapısına

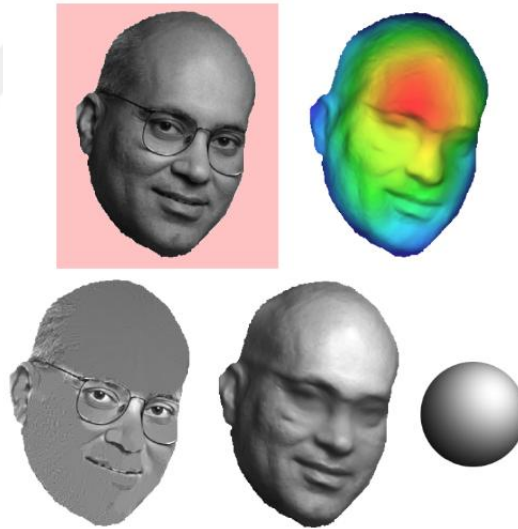
dönüştürülmesi gerekmektedir. Bu yüzden tüm bu işlemlerin gerçekleşme süreci artmakta ve tahmin edilemeyen hatalar ortaya çıkmaktadır [34-36].

Fotometrik yöntem kullanarak 3B modeller üretmek için oluşturulan sistemlerde çeşitli düzenekler oluşturulmaktadır. Kullanıcıların tespiti geçiş düzeneği üzerine konulmuş olan sensörler yardımıyla algılanmaktadır. Sistem üzerinde çalışan kayıt cihazları üzerlerindeki ışık kaynakları için her bir nesneye ait bir görüntüyü yaklaşık 20 ms kadar kısa bir sürede yakalamaktadır. Hassas bir değere sahip bu oran ile çok küçük değişikliklerin gerçekleşmesi kaçınılmazdır. Öte yandan ışık kaynakları her 20 ms de bir çalıştığı için kullanıcı sadece tek bir ışık kaynağına maruz kalmaktadır. Farklı çalışmalarda düzeneklerdeki aydınlatmalar için basit kaynakların dışında kızılötesi ışınlar da kullanılmaktadır [37]. Fotometrik stereo ile elde edilecek olan modellerin en büyük dezavantajı, görünür ışıkların belirli sürelerde patlamasından, kullanıcıların duyduğu rahatsızlıktır. Bu gibi durumların önüne geçmek için tarama sırasında sürekli olarak yansıtılan çoklu spektrum fotometrik yöntemi kullanılmaktadır [38]. Bu yöntemle patlayan flaşların engellenmesi için aynı sahne içerisindeki farklı pozisyonlara yerleştirilmiş olan kırmızı, yeşil ve mavi renkli ışıklarla ortam aydınlatılarak sahnenin eş zamanlı görüntüsü yakalanmaktadır. Ancak bu teknikte de bazı dezavantajlar bulunmaktadır. Nesnelerin tüm hatlarının net bir şekilde ortaya çıkarılması için nesne üzerinde şeffaf ya da parlayan bir kısım bulunmaması gerekmektedir. Bu gibi durumlarda görüntü üzerinde kalıntılar oluşmaktadır. Diğer taraftan yöntemin kullanımının gerektirdiği ışıklar ve yansımalar kullanıcılar için rahatsızlık verici düzeyde olabilmektedir. Ayrıca sistem, her bir farklı görüntü yakalama işlemi için farklı kalibre sürecine tabi tutulmaktadır.

- **Siluet Tabanlı:** Taranan nesnenin yakınlaştırılması için arka plana karşı alınan siluet yansımalarının kullanıldığı bu teknikte, nesnenin içbükey özellikleri tespit edilemese de, nesnenin şekli veya boyutu ile ilgili genel bir bakış yakalanabilmektedir. Genellikle, dönen bir tabla üzerinde, kamera tarafından çekilen farklı 2B görüntüler 3B bir model oluşturmak için birleştirilmektedir. Siluet tabanlı teknikler, iyi kontrastlı bir arka plana karşı üç boyutlu bir nesne etrafında bir dizi fotoğraftan oluşturulan ana hatları kullanmaktadır. Bu siluetler, nesnenin görsel gövde yaklaşımını oluşturmak için çıkarım yapılarak kesiştirilmektedir. Bu yaklaşımlarla, 3B bir nesnenin (bir kabın iç kısmı gibi) bazı kesişimleri tespit edilememektedir [39].

- **Görüntü Tabanlı:** Tarama yöntemleri arasında temassız pasif sınıfta kategorize edilen bu yöntemde bir nesnenin şeklinin belirlenmesi, görünür ışığın varlığına bağlıdır.

Görüntü temelli yöntemlerin araştırılması sırasında bazı kısıtlamaların ortaya çıktığı görülmektedir. Bu gibi teknolojiler her zaman derinlik tespiti için ek bir hesaplama adımı gerektirmektedir. Örneğin, bir döner tabla veya birden fazla kamera kullanımı gibi ek fiziksel yardımcılara ihtiyaç duymaktadır. Bu gibi yöntemlerden bazıları da sık sık gerçek nesne geometrisine uymayan sonuçların elde edilmesine neden olmaktadır. Bu kategoride çeşitli yöntemler öne sürülmüştür ve bu tarz yaklaşımlar, kontrolsüz bir ortamda yakalanan 2B görüntülerden faydalanılarak nesnelerin 3B yüzey verilerinin oluşturulmasını kapsamaktadır. Nesnelerin tek bir bakış açısından elde edilen yüzeyine ait görüntüsünden 3B görüntüsünün oluşturulması SIR (Single Image Reconstruction), bilgisayar görmesi alanında yeni bir konudur [40, 41]. SIR yöntemi, nesnelerin 3B yüzey verilerini elde etmede sağladığı avantaja rağmen çok yüksek çözünürlük ile oluşturulan mesh (çözüm ağı) yapısı bile çoğu zaman yüzey bilgilerini yeteri kadar keskinlikte yansıtamamakta ve ayrıştırılması zor olan yapılandırmalar ile tespit edilememektedir. Bu zorlukların sonucu olarak, bu alan üzerinde derinlemesine çalışmalar bulunmamaktadır.



**Şekil 2.8.** Görüntü tabanlı yöntem ile oluşturulmuş bir model [42].

Bilgisayar alanında, nesnelerin 2B görüntülerinden, 3B verilerinin elde edilmesinde en sık kullanılan ve nesne modellenmesi için yeniden yapılandırma üzerine son zamanlarındaki gelişmelere büyük katkı sağlayan en önemli yöntemlerden birisi 3DMM (3D Morphable Models), 3B dönüştürülebilir modellerdir [42, 43]. En çok bilinen ve en sık kullanılan 3DMM [44]'de verilmiştir. Şekil 2.8'de bu yöntem ile insan silüetine kaplanmış bir model verilmiştir. Bu 3DMM, 3B lazer tarayıcı ile elde edilen insan silüetinin, 3B şekil



bilgisi ve piksel değerlerinin birbirleri ile ilişkilendirilmesi vasıtasıyla oluşturulmuştur. Nesnelerin 2B yüzeyi ve şekilsiz dokusu ile ilgili bir takım parametreler derlenerek bu parametreler yardımıyla 3DMM modeller üzerine uydurulmaktadır. Bu algoritmaların düzgün ilerlemesi için bu parametrelerin başlangıçta doğru olarak verilmesi gerekmektedir ve bu konuda çok hassastırlar. Ayrıca uydurma işleminin çok fazla zaman alan bir işlem olması nedeni ile yöntemin video görüntülerine uygulanması, maliyeti yüksek donanımlar gerektirmektedir.

## **2.4. Sonuç**

Bu bölümde, 3B nesne tarama yöntemleri detaylarıyla incelenmiştir. Fiziksel nesne modellerinin oluşturulmasının ilk ancak en önemli adımı olan, veri elde etme safhasında kullanılacak yöntemin belirlenmesi büyük önem taşımaktadır. Kullanacağımız yöntem, tez çalışmasında incelediğimiz nesne tarama yöntemleri arasından, uygulamamızda kullandığımız donanımsal algılayıcı Kinect v1 kamera cihazının da desteklendiği, kısmen ToF yöntemini de içeren kızılötesi ışın projeksiyonu kullanan yapısal ışık yöntemidir. Bu yöntem ile kamera üzerinde bulunan RGB kamera ile birlikte yer alan kızılötesi algılayıcı sayesinde, ışınların nesne yüzeylerine ulaşma süreleri ile nesnelerin sahnelerdeki uzaklıkları birleştirilerek sanal ortamdaki konumları belirlenmektedir.

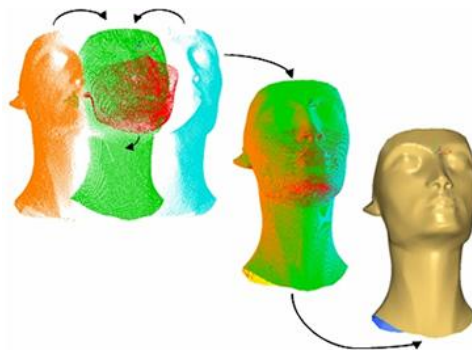
ToF yöntemi, yakalama yapılacak sahneye gönderilen ışık atımları (light pulse) ile ortamdan geri yansıyan ışınların kameranın merceğinde toplanarak, gönderilen ışının kendisine geri döndüğünde, bu gidiş-dönüş süresine bağlı olarak yüzeyin mesafesinin ölçülmesine dayanmaktadır. Microsoft Kinect v2 sensörü ToF yöntemiyle çalışmaktadır. Bu sistemde arka plan ışığının yoğun olduğu sahnelerde derinlik ölçümü yapılabilmesine olanak veren arka plan ışığını engelleme özelliği de bulunmaktadır. Oysaki Microsoft Kinect v1 ve cihazında kullanılan teknoloji, sahneye gönderdiği kızılötesi ışık huzmelerinin sahne içerisindeki obje yüzeylerinden yansıyan ışık demetleri ile nirengi oluşturarak o sahnedeki bir nesnenin derinlik bilgisinin ölçülmesi şeklindedir.

1 GHz frekansa sahip her bir saat çevriminde söz konusu ışık atımı ortalama 30 cm civarında mesafe kat etmektedir. Işık atımı ve gidiş-dönüş süreleri arasındaki zaman aralıkları ne kadar kısa bir zaman dilimi içerisinde gerçekleşirse ölçümler bir o kadar hassas şekilde yapılabilir. Bu yöntemde her pikselin birbirinden bağımsız olarak derinliği ölçülebilmektedir. Bu bakımdan tarama süreçleri, maliyet, uygulanabilirlik ve hız açısından değerlendirildiğinde en uygun yöntemin belirlendiği düşünülmektedir.

### 3. ÜÇ BOYUTLU (3B) NESNE TARAMA

Birçok 3B nesne tarama ve ayırt etme uygulamaları, çoklu araçlar, yeni teknolojiler ve donanımlar kullanan mevcut tarama sistemlerine, ihtiyaca göre yeni yöntemler eklenerek gerçekleştirilmektedir. Genellikle bu tarz uygulamalarda kullanılan yazılım kütüphaneleri, görüntüler ve uygulamalar, açık kaynak topluluğu (özgür yazılım) tarafından yönetilmektedir. Bu bakımdan ticari bir ürün niteliği taşıyan Microsoft Kinect algılayıcı kamera, resmi bir yazılım geliştirme kitine sahiptir. Kendi bünyesinde kapalı bir kaynak kod yazılımına sahip olmasının dışında özel bir açık kaynak kodlu SDK'sı (Software Development Kit) da mevcuttur. Tez çalışmasının uygulama kısmında, açık kaynak kodlu Kinect SDK yazılımı kullanılmaktadır.

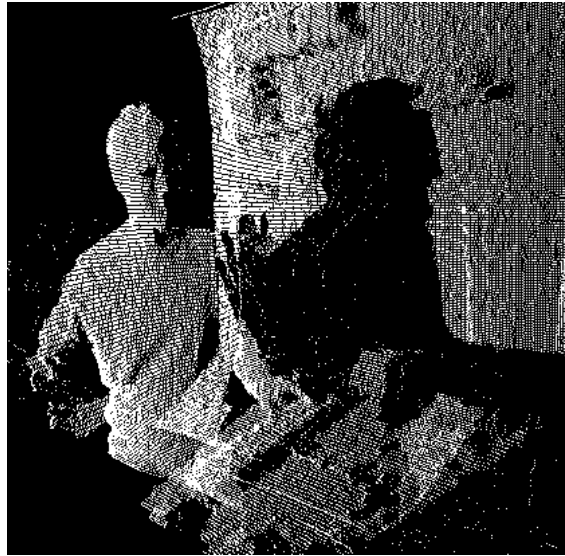
3B fiziksel nesnelerin tam bir 360 derece yeniden yapılandırılması, birden çok bakış açılı görüntülerinin, bu 3B nesnenin etrafından, farklı açılardan çekilmiş olmasını gerektirmektedir. Bunun temel nedeni tarayıcı kameranın herhangi bir açıdan yakalama işlemi yaptığı sırada, hangi pozisyonda olursa olsun nesnenin bir ya da daha fazla yüzünün her zaman kapalı kalacak olmasıdır. Bu durum 3B bir nesneye ait farklı çekimler arasındaki dönüşümün hizalanması ve birleştirilmesi yöntemlerini bir ihtiyaç haline getirmektedir [10]. Şekil 3.1'de aynı nesneden elde edilen birden fazla nokta bulutu verisinin birleştirilmesi gösterilmektedir. Dönüşümleri hizalamak veya birleştirmek için çeşitli yöntemler ortaya konulmuştur. Kullanılan kütüphane veya teknolojilerin bazıları bu tezin (örneğin PCL) uygulama tasarımı sırasında yapılan seçimlerdir. Uygulama yöntemleri, kamera iç yapısı, Kinect kalibrasyonu, algılayıcı yerleştirme yöntemleri, nokta bulutu kaydı, sayısal yöntemler ve 3B katı dönüşüm tahmini başlıkları altında incelenmiştir.



Şekil 3.1. Birden çok nokta bulutunun kayıtlarının birleştirilmesi [45].

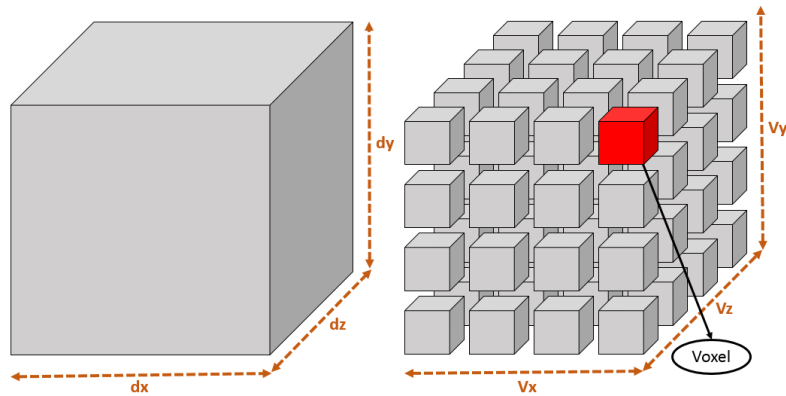
### 3.1. Nokta Bulutu

Nokta bulutu kümesi, (Point Cloud Library) çok boyutlu veriyi içerisinde barındıran bir veri yapısına sahiptir. Nokta bulutları genellikle üç-boyutlu verilerin gösterimi için kullanılmakta ve bu veriler  $x$ ,  $y$  ve  $z$  koordinatlarından oluşmaktadır. Özellikle  $z$  koordinatı sadece derinliğe göre kameranın nesne yüzeyine olan uzaklığını kaydetmektedir. Bunun dışında kalan diğer koordinat uzunlukları algılayıcı kameranın karakteristik özelliği olarak kameradan belirli bir mesafe (Örneğin: Kinect için 0,8 ile 4 metre) arasında kalan kısımların  $x$  ve  $y$  koordinatlarıdır. Bu aralıktan daha yakın ve daha uzak olan bütün noktalar algılayıcı kameranın hesaplama yeteneği dışında kalan bir mesafededir. Kinect algılayıcının asıl yakalanması istenilen nesneye göre minimum uzaklığı 0,8 ile 4 metreye ayarlanmalıdır. Bu bakımdan sadece 0,8 ile 4 metreye kadar olan uzaklıklardaki noktalar korunmuş olur. Bunların dışında kalan uzaklıklardaki veriler için delikli (tanımsız) yapılar ortaya çıkabilmektedir. Nokta verisi daha çok bazı nesnelerin dış yüzeyini göstermek için kullanılmaktadır. Nokta bulutundaki her bir nokta, geometrik koordinatların yanı sıra, normal yönü veya renk verileri de içerebilmektedir. Bu veriler noktalara eklendiğinde, bulut boyutları artmaktadır. Üç boyutlu bir nokta bulutuna RGB renk verisi de eklendiğinde, bu bulut artık altı boyutlu olabilmektedir. Bir nokta bulutu kümesi, 3B uzayda, farklı açı veya ışık koşullarında elde edilerek MeshLab sunucuları [46] gibi ortamlarda görüntüleyebileceğimiz, hatta kendi yüzeylerine renkler ve dokular uygulayabileceğimiz veri kümesidir. Şekil 3.2’de kapalı bir sahnenin, Kinect algılayıcı ile elde edilmiş nokta bulutu verisinin MeshLab ortamında gösterimi bulunmaktadır.



Şekil 3.2. Kinect ile yakalanan bir iç sahnenin nokta bulutu verisi

PCL, 3B geometri işlemi için yapılmış, çok boyutlu nokta bulutu barındıran bir BSD, (Brekel Software Development), lisanslı yazılım kitaplığıdır [47]. PCL, ROS (Robot Operating System) ile tamamen uyumlu olacak şekilde inşa edilebilmektedir, robotik, bilgisayarlı görme ve 3B geometri işlemleri gibi geniş bir kullanım alanına sahiptir. Nokta bulutu kütüphanesi, (PCL) nokta bulutu verilerini oluştururken, 3B fiziksel bir sahne içerisindeki her bir küp piksel değerinin nesne yüzeyi ile kamera arasında bulunan mesafesine göre hesaplanarak, koordinatlarının ölçülmesi mantığıyla çalışmaktadır. 2 boyutlu (x, y) piksel değerlerinin üç boyutlu (x, y, z) şeklindeki yapısına voksel adı verilmektedir. Her bir voksel bulunduğu koordinat noktasına göre içerisinde üç boyutlu değer taşımaktadır. Vokseller bilinen piksel özelliklerinin yanında fazladan bir derinlik verisi içermektedir. Bunun için de vokseller üç boyutlu kübik bir yapı şeklini almaktadır. Küp halinde eşit büyüklükteki vokseller bir dizi şeklinde bölünmüştür. Voksel bütün bir küp içerisinde, (sahne) ızgara şeklinde bölünmüş fiziksel alanlara ayrılan belirli sayıdaki küplerin her birine verilen isimdir. Her bir vokselin boyutları birbirine eşittir. Sahneleri oluşturan varsayılan bütün bir küpün metre cinsinden boyutu eksen başına üç metredir. Varsayılan voksel boyutu eksen başına 512'dir. Veri çekme sırasında, ızgarada önden arkaya geçilirken değerler her voksel için kontrol edilmektedir. Şekil 3.3'te, her bir vokselin aldığı değerlerin -1 ile 1 arasında değişen farklı ölçülerde olduğu gösterilmektedir. Bu değer her zaman sabit aralıkta olmak zorunda değildir. Ancak z koordinatı dışında, koordinatların verilen ölçü aralıklarında olduğu görülmektedir.



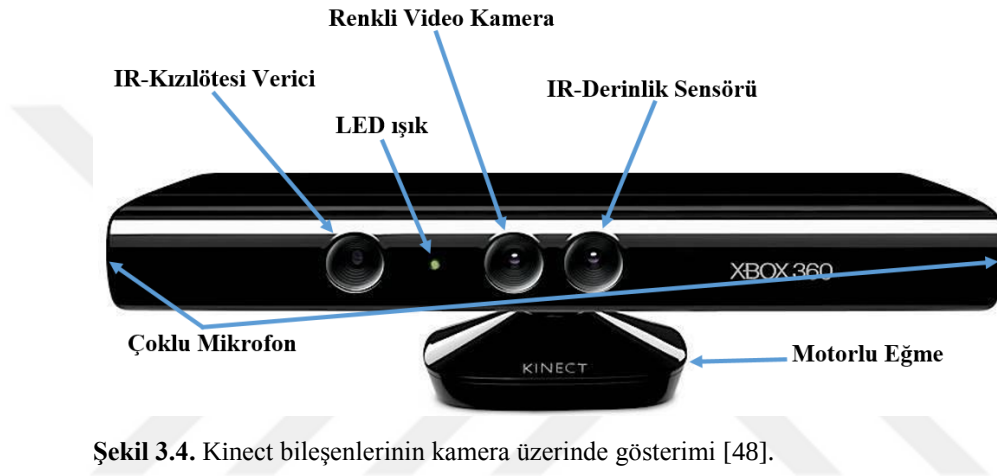
Şekil 3.3. Nokta bulutu voksel yapısı

Microsoft Kinect, bir derinlik işlemcisi ile birlikte bir kızılötesi sensörü birleştirmektedir. Algılayıcı kameranın hâkim olduğu aç “görüş alanı” olarak

adlandırılmaktadır. Derinlik işlemcisi derinlik çerçeveleri üretmektedir. Her derinlik çerçevesi  $512 \times 424$  nokta barındırabilen bir ızgara yapısıdır.

Her bir nokta x, y ve z kartezyen koordinat sistemi üzerinde yansıtılmaktadır. (0, 0, 0) noktası sensörün merkez pozisyonudur. Diğer tüm noktalar sensörün pozisyon açısına göre hesaplanmaktadır. Nokta bulutu veri yapısına göre x koordinatı yatay ekseninde, y koordinatı dikey ekseninde ve z koordinatı ise derinlik ekseninde konumlanmıştır.

### 3.2. Kamera İç Yapısı ve Kalibrasyonu

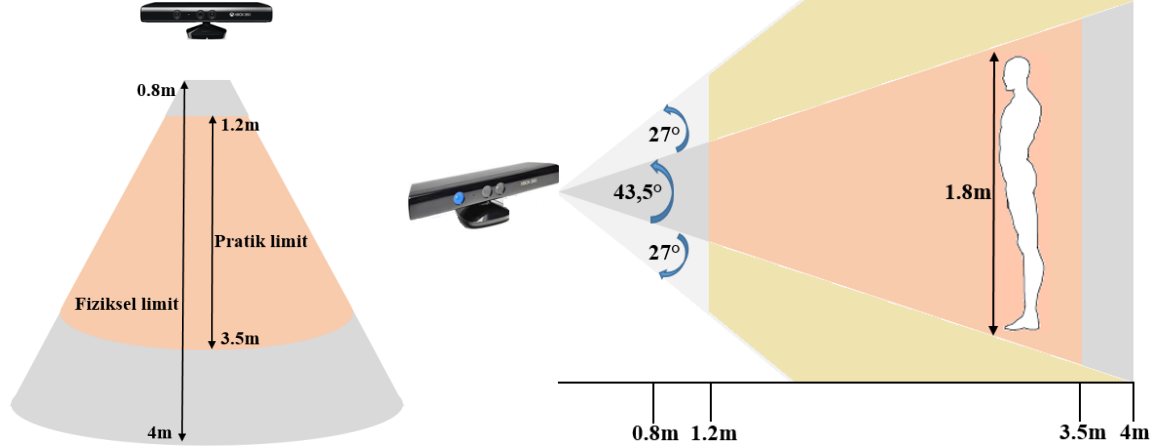


Şekil 3.4. Kinect bileşenlerinin kamera üzerinde gösterimi [48].

Kinect algılayıcı, Microsoft tarafından geliştirilmiş derinlik bilgisi çıkarımı ve hareket algılama özelliklerine sahip özel bir cihazdır. Kinect kameralar, derinlik bilgisinin yanında, renk ve iskelet verilerini içerebilmektedir. Kinect SDK kullanıcı ara yüzü vasıtasıyla bu tür verilere erişebilmekte ve bu verileri işleyebilen uygulamalar geliştirilebilmektedir. Alınan veriler sayesinde nesne ayırt etme, hareket algılama, ses algılama, yüz tanıma ve iskelet takip gibi pek çok uygulamalar geliştirilmesine imkân sunan Kinect algılayıcı bünyesinde barındırdığı dört adet donanım bileşeninden oluşmaktadır. Bunlar IR-kızılötesi projektör, RGB kamera, derinlik algılayıcı sensör ve mikrofonlardır [48]. Kameranın ilk etapdaki piyasaya çıkış versiyonu, bir oyun konsolu olan Xbox 360 için üretilmiş sürüm olan “Kinect for Xbox 360’dır”. Daha sonra popülerliğinin artmasıyla, ticari satış amaçlı üretilen “Kinect for Windows” sürümü piyasada yerini almıştır. Kinect sensörün, oyun konsolunun yeni bir sürümü olan Xbox One ile kullanıma sunulan ikinci modeli, "Kinect for Xbox One" ismiyle 2014 yılından itibaren piyasada yerini almıştır. Bununla birlikte, ticari sektörler için üretilen Kinect’in ikinci sürümü, “Kinect for Windows v2” ismiyle pazara çıkarılmıştır. Ancak bu sürümün üretimi bir süre sonra durdurulmuş, Xbox One için

üretilen Kinect sensörün, USB üzerinden bilgisayarlara bağlanmasına izin verilerek, ticari amaçlı olarak kullanılmasına izin verilmiştir. Şekil 3.4'te Xbox 360 için üretilen bir Kinect algılayıcının üzerinde taşıdığı bileşenleri gösterilmektedir. Buna göre, bir IR-projektör (kızılötesi), IR-derinlik algılayıcı ve renkli RGB kameradan oluşan sensör bileşenleri aşağıda detaylandırılmaktadır.

- **Renkli (RGB) Video Kamera:** Bu video kamera; kırmızı, yeşil ve mavi olmak üzere üç renk bileşenini algılayarak yüz tanıma ve diğer algılama özelliklerine destek vermektedir. RGB Görüntü algılayıcısı olarak adlandırılan bu algılayıcı, istenilen çözünürlükte, normal bir fotoğraf makinesi gibi çalışarak uygulamalarda renkli görüntü kullanabilmemize olanak sağlamaktadır. Video kamera saniyede 30 Hz'lik yani 30 FPS ile çalışmakta ve  $640 \times 480$  piksel çözünürlükte, kanal başına 8-bit görüntü oluşturmaktadır. Ayrıca 24-bit RGB renk verisi üretebilmektedir. Görüntü hızı azaltılarak video çözünürlüğü artırılabilir. Örneğin,  $1280 \times 960$  piksel çözünürlüğünde bir görüntü elde edilmesi istenirse, kameranın saniyede 10 Hz görüntü karesi ile çalıştırılması gerekmektedir.



Şekil 3.5. Kinect algılayıcının derinlik ölçüm aralığı

- **IR-Derinlik Algılayıcı:** Aydınlatma koşullarından bağımsız olarak üç boyutlu alanı görmek için bir kızılötesi yansıtıcısı ve tek renkli CMOS (Complimentary Metal-Oxide Semiconductor) ile birlikte çalışmaktadır. IR derinlik sensörü ve kızılötesi verici, birlikte derinlik algılama sistemini oluşturmaktadır. Bu iki bileşenin ortak çalışmasıyla bir sahnedeki 3B nesne yüzeyinin, kameraya olan uzaklığını içeren derinlik haritası çıkarılabilmektedir. Algılayıcılar, 30 Hz frekansta  $640 \times 480$  veya  $320 \times 240$  çözünürlükte 0,4

ila 4 metre arasındaki mesafeyi algılayabilmektedir. Derinlik algılayıcı, yakın mod çekiminde 0,4-3m, normal mod çekiminde 0,8-4m, kamera derinlik mesafesi aralıklarında güvenilir veriler sağlayabilmektedir. Görüş açıları, yatay ekseninde  $57,5^\circ$ , dikey ekseninde  $43,5^\circ$  olmak üzere, görüntüleme alanına sahiptir. En iyi derinlik verisi sonuçları 1,5 ila 3 metre arasında elde edilmektedir. Şekil 3.5'te Kinect algılayıcının derinlik ölçüm aralığı grafiksel olarak gösterilmektedir. Ayrıca derinlik verileri 12 bit derinlikte tutulmaktadır.

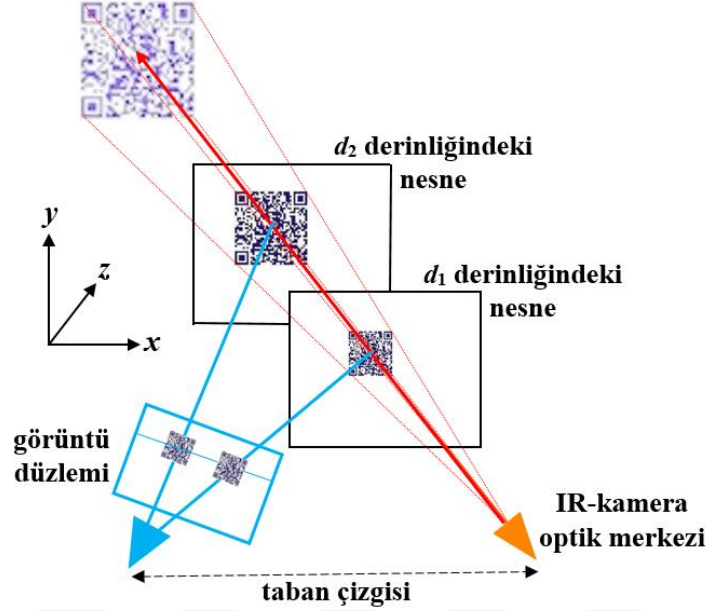
- **Çoklu Mikrofon:** Odadaki gürültüden, kullanıcı seslerini izole edebilen dört adet mikrofon dizisinden oluşan bileşendir.

- **Motorlu Eğme:** Kinect algılayıcının otomatik olarak aşağı ve yukarı yönde, (y-ekseninde)  $27^\circ$  hareket etmesini sağlayan bileşendir.

Kinect algılayıcı kamera, üzerindeki IR-projektör yardımıyla, kamera görüş alanı boyunca sahneye kızılötesi ışın demeti gönderirken, IR-derinlik algılayıcı, sahneden yansıyan kızılötesi demetlerini yakalamaktadır. Bu mantıkla Kinect yapısal ışık derinlik sensörüne bir örnektir. IR-projektör ve IR-derinlik algılayıcı kamera arasındaki geometrik ilişki, çevrimdışı bir kalibrasyon prosedürü ile elde edilmektedir. IR projektörü bilinen bir kızılötesi ışın demetini üç-boyutlu sahneye yansıtmaktadır. Bu ışın demeti, (RGB) video kamera tarafından görünemez ancak IR-derinlik algılayıcı kamera tarafından görüntülenebilmektedir. Yansıtılan noktaların yerel modelleri benzersiz olduğundan, kalibre edilmiş projektör nokta desenleriyle, elde edilen görüntüde gözlenen yerel nokta modelleri arasındaki uyumu yakalamak mümkündür. Bir noktaya ait derinlik bilgisi, nokta örüntüsünün göreceli sol-sağ dönüşümünden çıkarılabilmektedir. Bu dönüşüm, nesne ile kamera projektör düzlemi arasındaki mesafeye bağlı olarak değişmektedir [49].

Kinect geliştiricileri, algılayıcının yaptığı derinlik ölçümünü, üçgenleme işlemi olarak tanımlamaktadır [50]. Kızılötesi ışın kaynağı, sahneye yansıtılan sabit bir benek deseni oluşturmak için bir kırınım ızgarası ile çoklu ışınlara ayrılan tek bir ışın yaymaktadır. Bu model, IR-kamera tarafından yakalanmakta ve bir referans modeliyle ilişkilendirilmektedir. Referans modeli, sensörden bilinen bir mesafede bir düzlemi yakalayarak elde edilmektedir ve sensörün belleğinde saklanmaktadır. Sensöre olan uzaklığı referans düzleminden daha küçük veya daha büyük olan bir nesneye bir ışın yansıtıldığında, kızılötesi görüntüdeki ışınların konumu, lazer projektörü ve perspektif merkezi arasındaki taban çizgisi yönünde kaydırılmaktadır. Bu kaymalar, tüm ışınlar için, değişken bir görüntü veren basit bir görüntü korelasyon prosedürüyle ölçülmektedir. Her bir piksel için, sensöre

olan mesafe, karşılık gelen eşitsizlikten geri alınabilmektedir. Şekil 3.6'da Kinect algılayıcının derinlik ölçümü grafiksel olarak gösterilmektedir.



Şekil 3.6. Kinect algılayıcının derinlik ölçümü

Kinect kamerayı ölçümleme sırasında metrik derinliği ham derinliğe dönüştürmek gereklidir. IR-derinlik algılayıcı kamera açısı ile her pikselin derinlik pozisyonunu, aşağıdaki formülü kullanarak metrik derinlik ölçüleri sayesinde ham derinlik mesafesini hesaplamaktadır;

$$X_{ir} = \frac{f_{xir}}{(x - c_{xir})dm}, \quad Y_{ir} = \frac{f_{yir}}{(y - c_{yir})dm}, \quad Z_{ir} = dm \quad (3.1)$$

$$\begin{pmatrix} X_{rgb} \\ Y_{rgb} \\ Z_{rgb} \end{pmatrix} = \begin{pmatrix} X_{ir} \\ Y_{ir} \\ Z_{ir} \end{pmatrix} R + T \quad (3.2)$$

$$X_{rgb} = \frac{X_{ir} \cdot f_{xrgb}}{Z_{ir}} + c_{xrgb}$$

$$Y_{rgb} = \frac{Y_{ir} \cdot f_{yrgb}}{Z_{ir}} + c_{yrgb} \quad (3.3)$$

Yukarıdaki formüllerde,  $X$  ve  $Y$  derinlik görüntüsünün piksel pozisyonlarını,  $f_{xir}$ ,  $f_{yir}$  odak uzaklıklarını,  $c_{xir}$ ,  $c_{yir}$  IR algılayıcının kamera temel nokta pozisyonlarını ve  $dm$  metre cinsinden derinlik mesafesini ifade etmektedir.  $f_{ir}$  ve  $c_{ir}$  ölçümleme tarafından tahmin



edilmektedir. Renkli (RGB) görüntü ve derinlik resim arasındaki haritalamanın RGB ve IR kameralarının döndürülmesi  $R$  ve çevrilmesi  $T$  şeklinde ifade edilmektedir [51].

Kinect'in derinlik ve RGB kameraları, şekil 3.4'te gösterildiği gibi birbirinden bağımsızdır. Bu nedenle, RGB kamera koordinatları, derinlik kameralarıyla hizalanamamaktadır. Bu problemi çözmek için tablo 3.1'de gösterilen, derinlik kamera kalibrasyon parametreleri kullanılmaktadır [52]. Buna göre, 3B nokta bulutu koordinatlarında, RGB çerçeve tanımlayıcı nokta koordinatları tahmin edilmektedir.

**Tablo 3.1.** Kinect kalibrasyonunun döndürme ( $R$ ) ve dönüştürme ( $T$ ) parametreleri

$$\begin{aligned}
 f_x_{\text{rgb}} &= 5.2921508098293293e+02 & f_x_{\text{depth}} &= 5.9421434211923247e+02 \\
 f_y_{\text{rgb}} &= 5.2556393630057437e+02 & f_y_{\text{depth}} &= 5.9104053696870778e+02 \\
 c_x_{\text{rgb}} &= 3.2894272028759258e+02 & c_x_{\text{depth}} &= 3.3930780975300314e+02 \\
 c_y_{\text{rgb}} &= 2.6748068171871557e+02 & c_y_{\text{depth}} &= 2.4273913761751615e+02 \\
 R &= \begin{bmatrix} 9.9984628826577793e-01 & 1.2635359098409581e-03 & -1.7487233004436643e-02 \\ -1.47790961008364480e-03 & 9.9992385683542895e-01 & -1.2251380107679535e-02 \\ 1.7470421412464927e-02 & 1.2275341476520762e-02 & 9.9977202419716948e-01 \end{bmatrix} \\
 T &= [1.9985242312092553e-02 \quad -7.4423738761617583e-04 \quad -1.0916736334336222e-02]
 \end{aligned}$$



**Şekil 3.7.** Kinect ile yakalanmış RGB, derinlik ve nokta bulutu görüntüleri

Kinect algılayıcı kamera kullanarak üretilen derinlik ve RGB görüntülerine birer örnek Şekil 3.7'de gösterilmektedir. Bu görüntülerdeki RGB kamera saniyede 30 çerçeve değişmektedir ve her karede 640x480 piksel çözünürlükte görüntü yakalanmaktadır. Derinlik ölçümleri parametreleri kamera ölçümleme yöntemleri kullanılarak hesaplanabilmektedir. Kinect algılayıcı kamera, fiziksel bir sahnenin, aralıklı ölçümünü içeren bir derinlik haritasını oluşturmak için ışık tekniğini kullanmaktadır [53]. Bu veriler

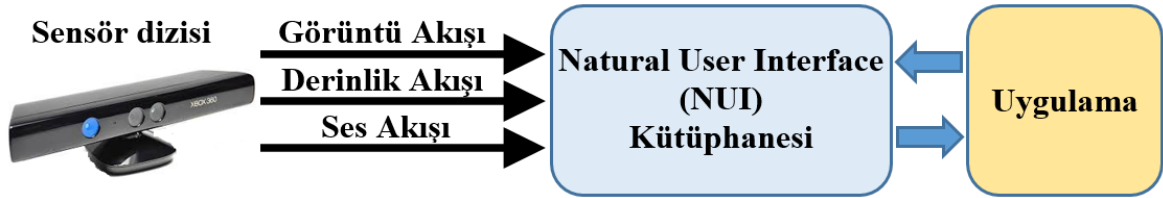
ayrık 3B nokta bulutu kümesi olarak toplanmaktadır. Kinect ile alınmış derinlik verisi gürültü, dalgalanma ve çok sayıda boşluk ‘delik’ içerebilmektedir [54].

Bazı araştırmacılar, Kinect algılayıcıları tıpkı 3B tarayıcılar gibi kapalı ortamların yoğun 3B haritalarını oluşturmak için piksel başına derinlik bilgileriyle birlikte RGB görüntülerini kullanmışlardır. Ancak, birçok çalışmadaki temel problem Kinect’in 3B taramalar için nispeten düşük  $X / Y$  derinlik çözünürlüğüne sahip olmasıdır. Bu sorunu gidermek için, [55]’de derinliği süper çözünürlük ile veri kalitesini artırmak için bir yöntem tarif edilmektedir. Ancak önerilen birçok yöntemde Kinect algılayıcılar sadece katı cisimlerin taranmasında kullanılabilmektedir.

### **3.3. Yazılım Bileşenleri ve Algılayıcı Yerleştirme**

Kinect yazılımı, belirli kısıtlar çerçevesinde algılayıcı donanımını kontrol eden ve elde edilen veriler ile çeşitli işlemler yürütülmesine olanak veren bir yazılım geliştirme kütüphanesi (SDK) olarak tanımlanmaktadır. Bu yazılım, kütüphanenin barındırdığı algoritmik bileşenlerin yanı sıra Kinect geliştirme kütüphanesine (araç) de atıfta bulunmaktadır. Son yıllarda, OpenNI (Açık Doğal Etkileşim) [56], Microsoft Kinect SDK [57] ve OpenKinect (LibFreeNect) [58] dâhil olmak üzere Kinect donanımıyla kullanılabilen birçok mevcut araç bulunmaktadır. Occipital tarafından yönetilen OpenNI, her zaman NITE adında uyumlu bir orta yazılım ile birlikte çalışmaktadır ve Mayıs 2018’e kadar en yüksek sürüm 2.2’dir. Microsoft Kinect SDK, Microsoft tarafından piyasaya sürülmüştür ve güncel son sürümü 2.0’dır. OpenKinect, yazılım geliştiriciler tarafından topluma açık platformlarda kullanıma izin verilmiş ücretsiz bir açık kaynak kütüphanesidir. Xbox oyun platformu dışında geliştirilen projeler, özellikle 2010 yılından itibaren popüler bir şekilde artmıştır. Artış gösteren OpenKinect ve OpenNI projelerinin yaygınlaşmasıyla, Microsoft Kinect SDK ismiyle kendi ürettiği yazılım geliştirme aracını, yazılımcıların hizmetine sunmuştur. Microsoft yazılımcıları, Kinect yazılım geliştirme kiti SDK’yı .NET platformu üzerinde geliştirdiklerinden sadece Windows işletim sistemi versiyonları üzerinde çalışmaktadır. Buna karşın, OpenNI ve OpenKinect, Linux ve MacOSX platformları üzerinde de çalışabilmektedir. Sonuç olarak geliştiricilerin büyük çoğunluğu, genellikle OpenNI ve Microsoft SDK kütüphanelerini kullanmaktadırlar. Bunun en önemli nedeni, OpenNI’nin yeni sürümünün (2.2), kullanıcıların Microsoft Kinect SDK ile aynı makineye yüklemelerine ve Microsoft Kinect sürücüsünü kullanarak her iki paketi çalıştırmasına izin vermesidir. Bu bakımdan, OpenNI’nin artık Kinect

sürücüsü ile birlikte uyumlu şekilde kullanılabileceği anlamına gelmektedir. Böylece, her iki paketten de yararlanmak isteyen kullanıcılar, bu iki sürücü arasında geçiş yapmak zorunda kalmayacaktır. Kinect SDK, 2014 yılı öncesinde geliştirilen 1.0 ile 1.8 arasındaki tüm sürümleri “Kinect for Windows” modeli olan v1 cihazı için geliştirilmiş olup, 2014 Ekim ayından itibaren “Kinect for Windows” modeli olan v2 cihazı için 2.0 son sürümü geliştirilmiştir. Güncel Kinect SDK araçları, 2014 yılından itibaren Microsoft denetiminde yayınlanmaktadır. Bunun en önemli sebebi, sunucularında çeşitli OpenNI projeleri bulunduran Occipital firmasının, kendi ürünlerini “Structure Sensor” üzerinde yoğunlaştırmasıdır.



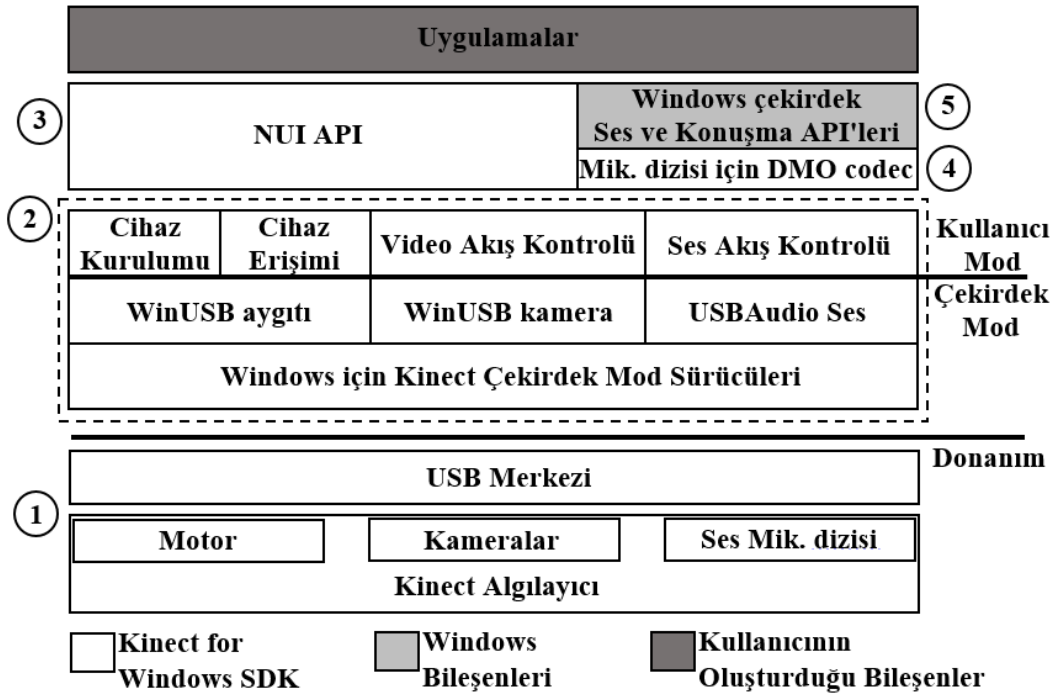
Şekil 3.8. Kinect uygulamasında yazılım-donanım etkileşimi

Şekil 3.8’de gösterildiği üzere, Kinect uygulamaları, elde edilen verileri işleyebilen, kendi başlarına ya da başka yazılımlarla ilişkili çalışabilen bir grafik kullanıcı arabirimine sahip yazılım olarak tanımlanabilir. Doğal kullanıcı arabirimi (NUI), Windows, API’ler (Application Programming Interface) için Kinect’in çekirdeğini oluşturmaktadır. Bununla birlikte, uygulamalarda, ses verisi akışı, renkli görüntü akışı, renk ve derinlik akışlarına erişilebilmektedir. Microsoft, geliştirilen örnek uygulamalar arasına C#, C++ ve Visual Basic gibi farklı dillerden üretilmiş kapsamlı uygulamalar eklemiştir. Kinect for Windows SDK, Kinect kullanarak gerçekleştirilen yazılım geliştirmesi için gerekli sürücüdür, referans API’ler sürücüler için teknik dokümantasyon sağlamaktadır.

Şekil 3.9, tez çalışmasında, yazılım mimarisi olarak kullanılan Kinect for Windows SDK 1.8 geliştirme kitinin çalışma prosedürünü, grafiksel olarak göstermektedir. Buna göre kullanıcıların geliştirdiği uygulamalarda, Windows ve NUI (Doğal Kullanıcı Arabirimi) API (Application Programming Interface) çekirdek, ses/konuşma API’leri üzerinden Kinect algılayıcı donanımıyla iletişime geçmektedir.

Yeniden yapılandırılacak olan üç boyutlu gerçek dünya nesnelerinin bulundukları sahnelerin verilerinin düzenli bir şekilde oluşturulması aşamasında, algılayıcı kamera ve donanımların pozisyonlarının en uygun bir şekilde seçilmesi gerekmektedir. Öncelikle

uygun bir açı veya gerekli pozisyonların oluşturulması için stratejiler geliştirilmelidir. Uygulanacak olan yönteme göre, sıralı kamera görüntülerinden alınan veriler üzerinde işleme veya bir döner tabla üzerine yerleştirilmiş nesneden farklı açılardan alınan görüntülerin verileri üzerinde bir dizi işlem gerçekleştirilebilmektedir. Bu tarz yöntemlerde karşılaşılan başlıca güçlükler, farklı taramalardan elde edilen verilerin daha sonraki süreçler için hizalanması ve birleştirilmesidir.



Şekil 3.10. Algılayıcı ve tarama nesnesinin yerleştirme pozisyonları

Mevcut birçok çalışmada, yeniden yapılandırılacak olan gerçek dünya nesnesinin, üzerine konulacağı bir döner tabanla çözüm üretilmektedir. Buna göre tek tek görüntülerin dönüşümü izlenerek ve çekilen her bir kare görüntüyle birlikte tablonun döndürme değerleri kaydedilerek işlemler yürütülmektedir [54]. Bu yaklaşımın dezavantajı ise nesneyi döndürmek için harici bir cihaza ihtiyaç duyulmasıdır. Birden fazla algılayıcı kamera mevcut olduğu durumlarda, ön ölçümlemeli kamera yöntemi kullanılabilir. Buna göre kameralar bilinen pozisyonlarda yerleştirilmekte ve bunlar arasında dönüşüm gerçek çekim işleminden önce çözülmemektedir [59]. Kameralar yeniden yapılandırılacak olan nesnenin etrafına yerleştirileceği ve birbirlerine göre pozisyonları sabit kalmak zorunda olduğu için bu yöntem taşınabilir amaçlar için uygun değildir ve çok sayıda kamera kullanımını gerektirmesinden dolayı da maliyet açısından dezavantajlıdır. Şekil 3.10'da, tek bir sabit Kinect algılayıcı kullanarak, kapalı bir sahne içerisinde, döner tabla üzerine monte edilmiş modelin, nokta bulutu verisinin çıkarılması gösterilmektedir.

### 3.4. Sayısal Yöntemler

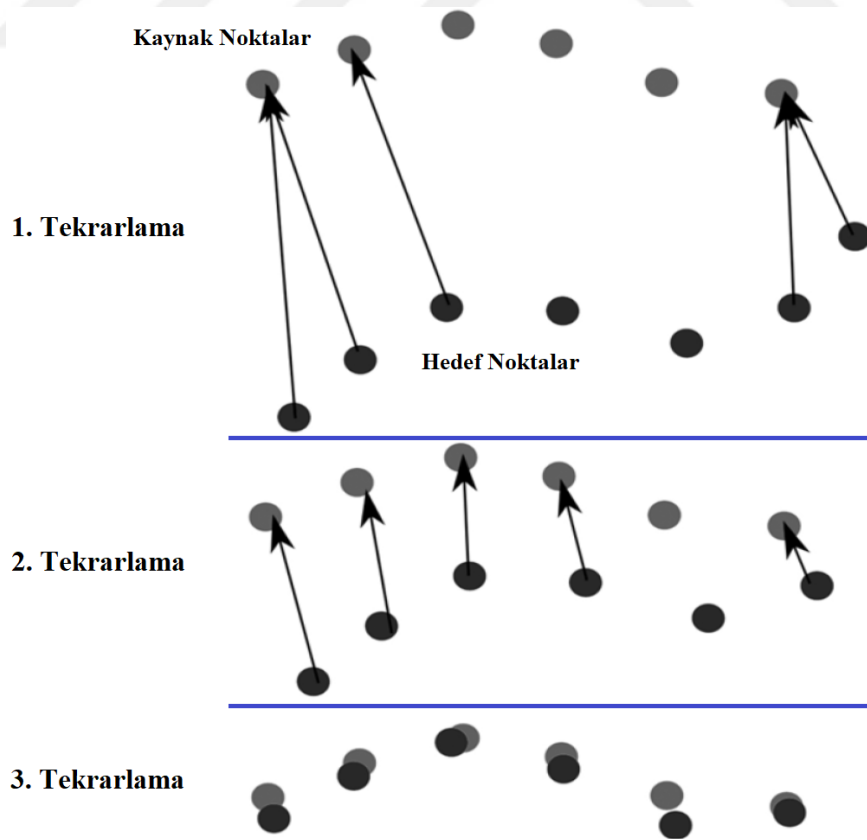
Çeşitli özelleştirilmiş algoritmalarla oluşan bu yöntemler genellikle kapalı ortamlarda yoğun 3B haritalar oluşturmak için kullanılmaktadır. Yeniden yapılandırma için çoklu Kinect derinlik bilgilerinin, doğru kamera pozisyonlarının ve görüntü dönüşlerinin birleştirilmesiyle gerçekleştirilmektedir. Etkili küçük çalışma alanlarının 3B haritalarının üretilmesinde kullanılan bu yöntemlerin ana fikrini, ardışık çerçeveler arasındaki mekânsal bilgilerin birleştirilmesi oluşturmaktadır. ICP, (En yakın nokta tekrarlama) 3B şekil hizalama hesaplamasında kullanılan bir algoritmadır. ICP, görüntülerdeki üst üste olan 3B noktaları karşılaştırarak her 3B görüntü için kamera dönüşümlerini hesaplamak için kullanılabilir. Bu yöntemi kullanarak gerçek zamandaki bütün görüntülerde ICP (Iterative Closest Point) tahminini kullanmanın sadece GPU (Graphics processing unit) uygulamaları sayesinde mümkün olduğu öne sürülmektedir [54].

Özellik çıkarımı ve eşleştirme, belirli bir alan içerisindeki herhangi bir nesneye ait mekânsal bilgilerinin, kamera pozisyonuna göre üç boyutlu olarak sanal ortamdaki koordinat sistemine göre konumunun belirlenmesidir. Bunlardan birinci adım olan (SURF) algoritmasında, farklı açı ve mesafelerden özellik noktaları ile ilgili ilişkilendirme yapılmaktadır. Kamera açısı hesaplanması sırasındaki farklı tanımlamalar neticesinde yanlış nokta bulutu eşleştirilmesi oluşabilmektedir [60]. Tıkanıklık, örtüşme gibi yanlış eşleştirmelerinde ikinci adım (RANSAC) algoritmasına başvurulmaktadır. Algoritma,

görüntü noktaları ile bir alt görüntü noktaları arasındaki ilişkilendirmeler için kullanılmaktadır. Kamera pozu tahmin etmek için üç boyutlu noktalar arasında bir dizi denklemler kümesi oluşturmaktadır. Bu denklemler en az görüntüleme hatası için içsel parametreler arasındaki karesel mesafelerin toplamıdır. Bu algoritma aynı hizadaki 2B veri setlerini takip ederek, olabildiğince büyük düzlemler çıkararak model olması istenen asıl nesneden ayırma işlemidir.

Aynı nesneye ait alınmış, birden fazla nokta bulutu kümesi verisi birleştirilmek istendiğinde, nesnenin bir kesitinin benzer noktalarına ait konumlarında, mesafe farkı meydana gelmiş ise, bu iki nokta arasındaki fark en aza indirilmelidir. Bunun için ise üçüncü adımdaki dönüşüm işlemini gerçekleştiren (ICP) algoritması uygulanmaktadır. Bu algoritma ile benzer bu iki nokta bulutu çiftine komşu olan ortak en yakın bir nokta belirlenmektedir. Dördüncü adımda verilerin görselleştirilmesi için uygun yapıya dönüştürülmesi sağlanmaktadır. Görselleştirilecek olan sanal ortamın formatına uygun dönüşümler gerçekleştirilmektedir.

#### 3.4.1. ICP (Yinelemeli En Yakın Nokta)



Şekil 3.11. ICP algoritmasında örtüşen nokta bulutu setleri

ICP algoritması, ilgili nesnelere ait çerçeveler arasındaki her bir kare hatalarını en aza indirerek, bir nokta bulutu ile bazı referans yüzeyler veya başka bir nokta bulutu arasındaki dönüşümü bulmayı amaçlamaktadır. ICP'nin “yinelemesi”, çözüm sırasındaki, yerel hatanın minimuma yaklaştıkça yazışmaların yeniden ele alınması gerçeğinden kaynaklanmaktadır. ICP'yi herhangi bir gradyan iniş yöntemi olarak, nispeten iyi bir başlangıç noktasına sahip olduğunda uygulanabilmektedir. Aksi takdirde, ilk yerel minimumda süreç kilitlenecek ve çözüm işe yaramayacaktır.

ICP algoritması [61], temel olarak ardışık nokta bulutlarını hizalamak için kullanılan bir algoritmadır. Tekrarlı olarak en yakın noktaya ulaşmak için çeşitli yaklaşımlar bulunmaktadır. Şekil 3.11'de gösterildiği gibi algoritmanın ana prensibi, en yakın iki noktadaki bulut setleri arasındaki Öklid mesafesinin en aza indirilmeye çalışılmasıdır. Bu algortmada bu işlemler tekrarlı bir şekilde yapılmaktadır. Algoritmada, iki nokta bulutu (kaynak ve hedef) ve yineleme sayısı girdi olarak alınmaktadır ve aralarında bir dönüşüm matrisi döndürülmektedir. Hizalanması için girdi olarak verilen iki nokta seti;

$$X = \{x_1, \dots, x_{N_x}\} \text{ ve } P = \{p_1, \dots, p_{N_p}\}'\text{dir.} \quad (3.4)$$

Hedeflenen hesaplama, rotasyon ve çeviri kombinasyonu kullanarak ortalama karesel hata maliyetini her tekrarda en aza indirmektir.

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - R p_i - t\|^2 \quad (3.5)$$

Burada  $x_i$  ve  $p_i$  birbirine karşılık gelen noktalardır. Buna göre algoritma aşağıdaki adımları sırayla gerçekleştirmektedir;

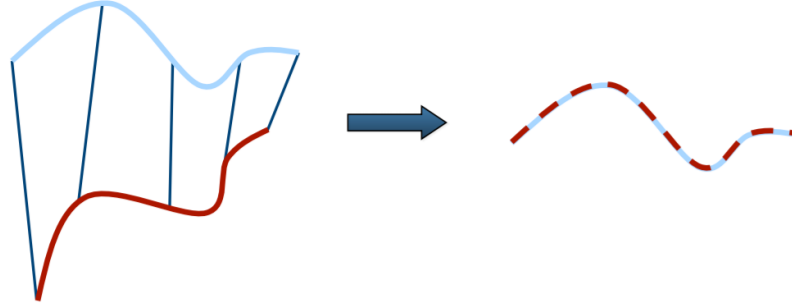
1. Kaynak nokta bulutundaki her bir nokta için, hedef nokta bulutundaki en yakın nokta bulunmaktadır.

2. Her kaynak nokta bulutunun bir noktasının önceki adımda bulunan eşleştirmeye en iyi şekilde hizalayacak bir ortalama karesel hata maliyet fonksiyonu kullanarak rotasyon ve çeviri kombinasyonu tahmin edilmektedir.

3. Elde edilen dönüşümü kullanarak hedef noktalar dönüştürülmektedir.

4. Tekrarlama, (noktalar yeniden ilişkilendirilmekte ve yineleme sayısı sona erece kadar önceki adımlar tekrarlanmaktadır).

Yazışmalar doğru bir şekilde oluşturulduğu takdirde, nizami rotasyon ve çeviri işlemleri kapalı formda hesaplanarak tekrarlanmaktadır. Şekil 3.12’de örtüşen nokta bulutları için temsili bir grafik gösterilmektedir. Böylece iki farklı çerçeve arası nokta bulutu hizalama işlemi tam örtüşme sağlanarak yerine getirilmektedir.



Şekil 3.12. ICP algoritmasında kapalı formda tam örtüşen nokta bulutu setleri

### 3.4.2. RANSAC (Rastgele Örnek Konsensüsü)

RANSAC algoritmasının ismi, RANdom SAMple Consensus kelimelerinin kısaltılmış ilk harflerinden oluşmaktadır [62]. Bu algoritma, sınırlayıcıları ortadan kaldıran ve verileri geometrik olarak tutarlı ve kısıtlı bir ortamda tutan klasik ancak sağlam bir tahmin tekniğine dayanmaktadır. Hedef parametrik modeli tanımlamak için minimum sayıda veri elemanını örnekleyerek, kararlı bir çözüme ulaşmaktır. RANSAC algoritmasındaki temel fikir, bir dizi kaynak verinin içerisinde barındırdığı gürültülü verileri ortadan kaldırmaktır. Başka bir deyişle, aykırı değerleri ve varoluşları, duruma bağlı olarak belirlenen veya verilen bir eşik değeri ile belirlemektir. Klasik analiz, belirli bir olasılıkla iyi bir çözümü garantilemek için gereken sayıda yinelemenin veri setinin dâhili oranına bağlı olduğunu belirtmektedir.

RANSAC algoritmasının ana hatları şu şekildedir;

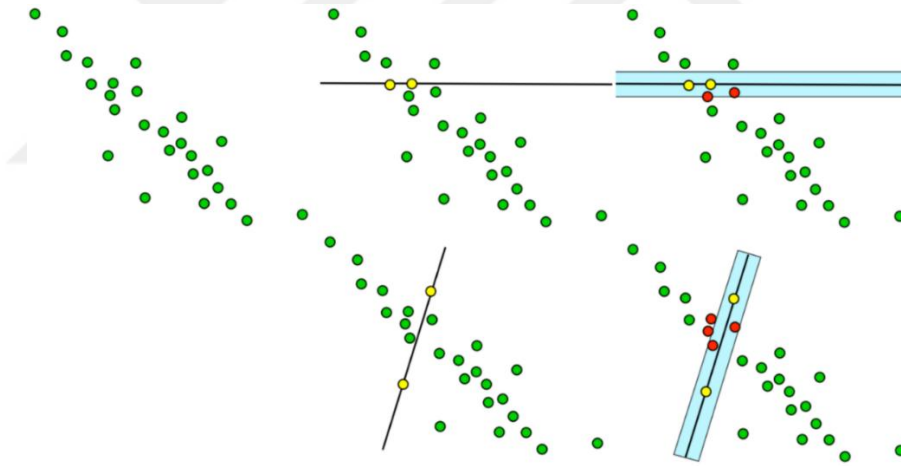
- 1-  $N$  kere tekrarla,
- 2- Rastgele, küçük bir nokta alt kümesi seç ( $s$ ),
- 3- Bu  $s$  adet noktaya / alt kümeye, bir çizgi / model oturt,
- 4- Kalan noktalar arasında, doğruya mesafesi  $t$  'den küçük olan noktaları / modele yakın olanları bul, gerisini aykırı nokta olarak reddet,
- 5-  $d$  ya da daha çok böyle nokta varsa, doğruyu bu noktalar ile birlikte tekrar oluştur ve işlemi bitir, aksi halde bir sonraki tekrara geç.

RANSAC algoritmasının parametre seçiminde, seçilen ilk nokta sayısı  $s$  modeli oturtmak için gereken minimum sayıdır (çizgi için iki). Uzaklık eşiği  $t$ , noktaların aykırı



olmama olasılığı  $p$  olacak şekilde seçilir (örn. 0.95). Standart sapması  $\sigma$  olan sıfır ortalamalı Gauss gürültüsü:  $t^2=3.84 \sigma^2$ . Örnek sayısı  $N$ ,  $p$  olasılıkla, en az bir rastgele örnek, aykırı değerlerden arınmış şekilde seçilmektedir. Örnek sayısının adaptif olarak belirlenmesi, aykırı olmayan oranı  $(1 - e)$  önceden bilinmediği için, en kötü durum seçilmektedir, örneğin %50. Daha çok aykırı olmayan nokta bulundukça bu oran artırılmaktadır, örneğin %80  $\Rightarrow e = 0.2$ . Süreç;

- $N = \infty$ , örnek sayısı = 0,
  - $N >$  örnek sayısı olduğu sürece,
  - Bir örnek seç ve aykırı olmayanların sayısını belirle,
  - $e = 1 - (\text{aykırı olmayan sayısı}) / (\text{toplam sayı})$ ,
  - $N$ 'yi  $e$ 'den tekrar hesapla,
- $$N = \log(1 - p) / \log(1 - (1 - e)^s).$$
- Örnek sayısını 1 arttır, şeklinde ilerlemektedir.



Şekil 3.13. RANSAC algoritmasının iki adım uygulanması

RANSAC algoritması için iki farklı örnek için uygulanan iki adımı, şekil 3.13'de gösterilmiştir. Buna göre rastgele,  $k$  ögesi seçilmekte daha sonra, eşikler ve aykırı değerler, eşik değerine göre süreçte belirtilen sırayla belirlenmektedir.

### 3.4.3. Derinlik-Ölçeği Yöntemi

Derinlik ölçeği yönteminde Lowe'nin ölçümüne benzer şekilde, bir eşleşme setindeki yanlış eşleşmeleri hesaplamak için basit bir yöntem kullanılmaktadır [63]. Aynı sensörden alındığı varsayılan 3B taramalarda, RGB kameraların dahili kalibrasyon matrisinin etkileri görmezden gelinebilmektedir, buna göre aşağıdaki değerler belirlenmektedir;

$$DS = Depth * Scale$$

(3.6)

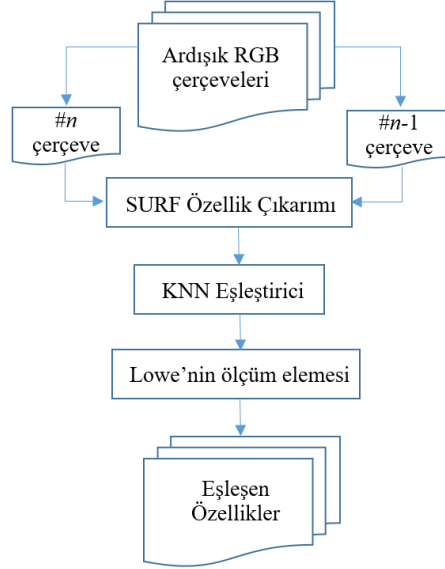
Denklem (3.6), 3B dünyadan gelen derinlik özellik bilgisinin ve ölçeğin yoğunluk görüntüsündeki özelliğin destek alanının boyutunun olduğunu ve 3B dünyadaki destek alanının ölçeğini temsil etmektedir. Denklem, doğrudan perspektif projeksiyonunun ters uygulamasından kaynaklanmaktadır. Aynı 3B noktasına ait özelliklerin aynı  $DS$ 'ye sahip olduğu düşünülebilir. Bu ölçü, değişmez bir destek alanını döndüren birçok özellik detektörü için uygulanabilmektedir. Uzun özellik tanımlayıcılarının tersine, (örn. SIFT 128 bayt bir vektördür), Depthscale basit bir tam sayı ile temsil edilebilmektedir ve bu da karşılaştırmanın daha verimli olmasını sağlamaktadır. Ayrıca, kapsüllenmiş ölçek bilgisinin, yoğunluk temelli özellik tanımlayıcısı ile ilgili olmaması nedeniyle, eşleştirme sırasında kullanılabilecek ekstra bilgi vermektedir. Ancak sadeliği nedeniyle yoğunluk özellikleri kadar açıklayıcı olmayacaktır.

#### 3.4.4. Lowe'nin SNN Yöntemi

Lowe'nin 2004 yılında yaptığı bir çalışmaya göre yinelenen desenler ve çeşitli geometrik deformasyonlar nedeniyle, farklı özellikler birbirlerine benzeyebilir, dolayısıyla en yakın eşleşme her zaman doğru eşleşme olmayabilir [64]. Bu tür aykırı sayıları azaltmak için, KNN (K-Nearest Neighbors), ( $k=2$ ) komşularını hesapladıktan sonra, ilk en yakın ikinci en yakın komşunun benzerliğini kontrol etmeyi önermiştir. SNN, (Second Nearest Neighbour) ikinci en yakın komşuluk olarak ifade edilmektedir. En yakın ikinci komşunun doğru eşleşmeler için uzak olması beklenmiştir. Bu "yakınlık" ölçüsü, 0 ile 1 arasındaki bir eşik değer ile gerçekleştirilmektedir. 1'den biraz daha az olan eşik değerler, sistemin neredeyse her zaman bir eşleşme olarak en yakın özelliği alması anlamına gelmektedir, bu da pratik olarak tedbiri etkisiz kılmaktadır. Öte yandan 0'a yakın olan eşikler, neredeyse hiçbir eşleşme bulunmayan paranoyak bir ölçü ile sonuçlanmaktadır.

Derinlik-ölçeği gibi, Lowe'nin ölçüsü de eşleşmenin doğru bir eşleşme olup olmadığını belirlemek için bir eleme yöntemi kullanmaktadır. Lowe'nin ölçüsü,  $knn$ 'den sonra gerçekleştirilmektedir ( $k$ , 2'ye eşittir).  $Knn$ , bir dizi eğitim ve test özelliği setini almaktadır. Daha sonra iki en yakın test özelliği ile her bir eğitim özelliği birbiri ile eşleştirilir, bunlardan bir tanesi en yakın olan diğeri ise ikinci olanıdır. Lowe'nin ölçüm fikrinden sonra eşlemenin bir doğruluk eşleşmesi olup olmadığı belirlenmek istenebilir. Bu eşlemeler yerel özellikler taşıdığından, ilk test özelliğine benzer ikinci bir test özelliği de

barındırabilir. Burada tek bir eşik değeri ile birinci ve ikinci özellik benzerliklerini kontrol edilebilmektedir.



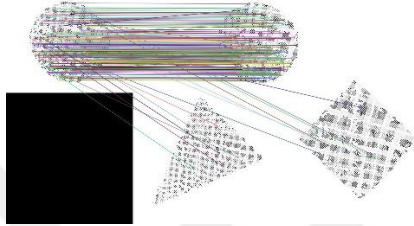
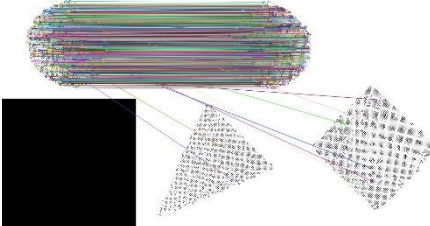
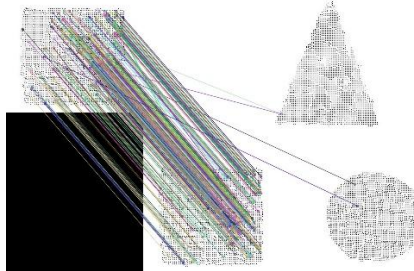
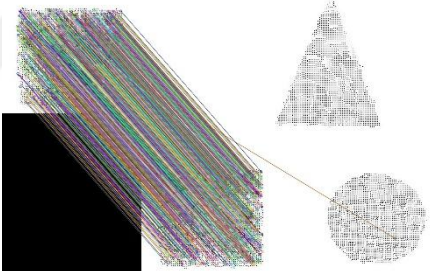
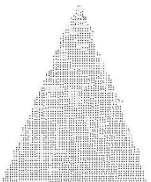
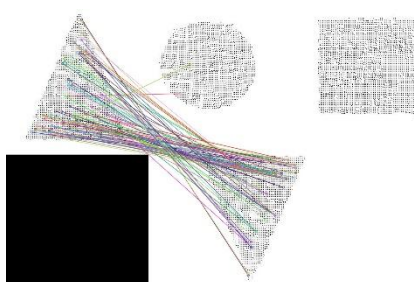
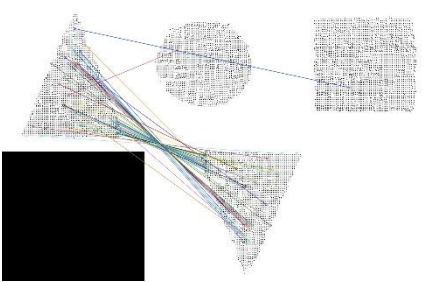
Şekil 3.14. Lowe'nin ölçütü ile KNN eşleşme grafiği [63].

Şekil 3.14'te gösterildiği gibi, bu yöntemin ana fikri, gözlemlenen her özellik için iki aday özelliği döndüren *knn* eşlemesinden sonra, bu iki aday özelliği çok benzerse aday özelliklerini kontrol etmek, daha sonra Lowe'nin ölçüm etiketleri ile yanlış yani diğer bir deyişle aykırı eşleme olarak eşleştirmektir. Eşleştirilen özellikler, ardışık nokta bulutları arasında 3B dönüştürme matrisi oluşturmak kullanılabilmektedir. Lowe'nin ölçütünde adayların birbirlerine olan benzerliklerine bakılarak bir eleme süreci gerçekleştirildiği için aynı kaynaktan edinilen verilerdeki karşılaştırmalarda dahi açı ve uzaklık değişimlerinden ziyadesiyle etkilenmektedir.

Tablo 3.2'de görüldüğü gibi SIFT ve SURF eşleştirme algoritmaları değerlendirme sonuçları verilmiştir. Buna göre, farklı konumlara ait olan çoklu nesne sahnesinde üç farklı geometrik nesne için eşleştirme işlemini nokta bulutları üzerine uygulanmıştır. SIFT ve SURF algoritmaları ile anahtar nokta sayısı ve eşleşen anahtar nokta sayısı değerlendirilmiştir. Üç nesneye ait RGB görüntüler, nokta bulutu eşleştirilmesini değerlendirmek için seçilmiştir. Toplamda, farklı açılardan ele geçirilen üç farklı geometrik nesne için 144 ayrı eşleştirme işlemi gerçekleştirilmiştir. Bu algoritmaların uygulanması özellik anahtar noktalarının eşleştirilmesi tablo 3.2'de gösterilmiştir. SURF algoritması ortalama 227 özellik bulmuştur. Ancak SIFT algoritması ortalama olarak

sadece 124 özellik bulabilmiştir. Dolayısıyla SURF algoritmasının SIFT algoritmasından daha fazla özellik bulduğu gözlenmiştir. Ayrıca tablo 3.3'te gösterildiği gibi özellik eşleştirmesi için ortalama çalışma süreleri saniye cinsinden verilmiştir [60].

**Tablo 3.2.** SIFT, SURF eşleştirme algoritması sonuçları

| Test Nesneleri                                                                                      | SIFT Eşleştirme                                                                     | SURF Eşleştirme                                                                      |
|-----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <br><b>Daire</b>   |    |    |
| <br><b>Kare</b>   |   |   |
| <br><b>Üçgen</b> |  |  |

**Tablo 3.3.** SIFT, SURF özellik eşleştirme sayısı ve çalışma süreleri

|             | Ortalama çalışma süresi |                |        | Ortalama Özellik Eşleme Sayısı |
|-------------|-------------------------|----------------|--------|--------------------------------|
|             | Test Nesnesi            | Test Görüntüsü | Toplam |                                |
| <b>SIFT</b> | 0.159                   | 0.353          | 0.512  | 124.3                          |
| <b>SURF</b> | 0.037                   | 0.135          | 0.172  | 227.7                          |

### 3.4.5. 3B Katı Dönüşüm Tahmini

Kayıt problemi olarak bilinen, bir sahnenin iki farklı görünümü arasındaki katı dönüşümünü bulmak, bilgisayar görme adımlarında gerekli bir aşamadır. Örneğin, farklı görünümlerden verileri birleştirebilmek için dönüşümler ortak bir referans çerçevesinde yapılmalıdır. Matematikte, bir Öklid uzayının katı bir dönüşümü veya Öklid izometrisi, her bir nokta çifti arasındaki mesafeleri korumaktadır.  $R^2$  düzlemi,  $R^3$  uzayı veya gerçekte  $n$  boyutlu  $R^n$  uzayının katı dönüşümleri, Öklid geometrisinin temelini oluşturdıkları için bir Öklid dönüşümü olarak adlandırılmaktadır [65].

Katı dönüşümler, rotasyonlar, çeviriler, yansımalar veya bunların birleşimlerini içermektedir. Tüm katı dönüşümler Affine dönüşümlerinin birer türevleridir. Katı dönüşümlerin hepsinin kümesi,  $n$  boyutlu Öklid uzayları için  $E(n)$  olarak bilinen yani Öklid grubu olarak adlandırılan gruplardır. Genel olarak, bir katı dönüşüm, herhangi bir  $v$  vektörü üzerinde hareket ederken, formun dönüştürülmüş bir  $T(v)$  vektörünü üreten bir dönüşüm olarak tanımlanmaktadır. Bu katı dönüşüm denklem 3.7’de verilmiştir.

$$T(v) = R \times v + t \quad (3.7)$$

Verilen denklemde  $R^T = R^{-1}$  ifade eder.  $R$  bir ortogonal dönüşümdür.  $t$  ise vektör için verilen kökenin dönüşümüdür.

Bir dönüşümün katı olduğunu teyit etmek için, noktalar veya metrikler arasındaki mesafenin bir ölçüsü gereklidir. Eşleştirilen tanımlayıcı iki dizi nokta arasındaki ilişki,  $R$  matrisi ve  $t$  çeviri vektöründen oluşan Öklid dönüşüm matrisi ile sağlanmaktadır. Öklid uzaklık formülü  $R^n$  Pisagor teoreminin genelleştirilmiş biçimidir. Formül,  $X$  ve  $Y$  gibi iki nokta arasındaki mesafeyi, koordinat eksenleri boyunca mesafelerin karelerinin toplamı olarak vermektedir. Denklem 3.8’de uzaklık formülü verilmiştir.

$$d(X, Y)^2 = (X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \dots + (X_n - Y_n)^2 = (X - Y) \cdot (X - Y) \quad (3.8)$$

Verilen denklemde  $X = (X_1, X_2, \dots, X_n)$  ve  $Y = (Y_1, Y_2, \dots, Y_n)$  olarak ifade edilir. “.” ise sayısal çarpımı göstermektedir.

$$d_i = R \cdot d_i + T + V_i \quad (3.9)$$

Denklemden  $d_i$ ,  $m_i$  eşleşen tanımlayıcı noktaları sunar. Rotasyon matrisi için  $R$ , dönüşüm matrisi  $T$  ve gürültü matrisi için  $V$  kullanılmıştır. Dönüşüm matrisinin bulunma süreci denklem 3.10'da verilmiştir. Bu denklemde  $N$  küme sayısını temsil etmektedir.

$$\bar{d} = \frac{1}{N} \sum_{i=0}^N d_i \quad (3.10)$$

### 3.5. Sonuç

Bu bölümde, 3B nesne tarama sürecinde gerçekleştirilen adımlardan, uygulanabilecek yöntemlerden, taranacak nesnenin yerleştirileceği düzeneğin ayarlanması ve tarama için kullanılacak aygıtın seçimi ve yerleştirilmesinden bahsedilmiştir. Maliyet açısından tek bir Kinect ile tarama işleminin yerine getirilmesi amaçlanmıştır. Kinect kameranın çalışma prensibi ve işlevselliği hakkında bilgiler verilmiştir. 360 derecelik fiziksel nesnelerin birden fazla alınmış taramalarının eşleştirilmesi için nokta bulutu verilerinin 3B kayıt yöntemlerine değinilmiştir. Genellikle metrik mantığıyla işleyen yöntemlerde nokta bulutu verisindeki artış ve nokta arasındaki hassasiyetin artması yöntem sürecinin uzamasına ve doğruluğun azalmasına neden oldukları fark edilmiştir. Bu bakımdan bilinen 3B kayıt algoritmaları kullanmanın yerine bir sonraki bölümde detaylandırdığımız, önerilen kayıt işlemi yerine getirilmiştir.

#### 4. 3B TARAMA SİSTEMİ

Bir sahne içerisindeki taranan nesnenin duruşu ve hareketi, bununla birlikte tarama sırasında kullanılan kameranın sayısı ve işleyişi 3B tarama sistemlerini birbirinden ayıran temel özelliklerdir. Tez çalışmasında, tarama işlemlerinde kullanılması düşünülen teknolojinin Kinect algılayıcı olarak hedeflendiyse de nasıl kullanılması gerektiği konusunda farklı yöntemler değerlendirilmiştir. Değerlendirmeler algılayıcı kamera donanımının fiziksel özellikleri doğrultusunda ele alınmıştır. Nokta bulutu kümesi elde edilmesini sağlayan yazılım kütüphanelerinin el verdiği ölçüde algılayıcı donanımın tüm yeteneklerinden faydalanılmıştır.

Günümüzde yapısal ışık kaynağına dayalı tarama ve hızlı lazer tarama teknikleriyle nitelikli olarak gerçeğe oldukça yakın, hassas ölçümler yapılabilmektedir [66]. Örneğin; yüksek hassasiyette karasal lazer tarayıcılar kullanılarak, kültürel miras'a ait yapılar üzerinde gerçekleştirilen 3B modelleme ve yapısal analiz işlemleri, değerli çalışmalar arasındadır [67]. Bir başka çalışmada, orman haritalarının uzaktan algılanması ve doğruluk hesaplamaları için 3B nokta bulutlarının işlenmesinden faydalanılmaktadır [68]. Bu tekniklerin uygulanmasında yaşanan zorluklar ve sistemlerin yüksek maliyeti yaygın olarak kullanılmasında birer engel teşkil etmektedir. Bunun yanında, sistemlerde kullanılan cihazların yönetilmesi için çoğu kez bir uzman bilgisine başvurulmaktadır. Bu tekniklerin temel problemi tam 3B model oluştururken, kamera veya nesneden birinin, bazen her ikisinin de hareketli olması gerekmesidir. Benzer sıkıntılar, insana ait 3B modelin çıkarılması sırasında, insanın sabit bir şekilde poz verememesine bağlı olarak oluşabilmektedir. Bu gibi olumsuzlukların üstesinden gelmek adına farklı yaklaşımlar ortaya konmuştur. Bunların başında görüntü tabanlı yöntemler bulunmaktadır. Hem 3B nesne modellemede hem de insan vücuduna ait modelleme işlemlerinde görüntü tabanlı yöntemler sıkça kullanılmaktadır. "State-of-the-art" çoklu bakış yöntemi ile etkili ve doğruluğu yüksek sonuçlar elde edilebilmektedir [69]. SFS (Shape-From-Silhouette) yönteminde, birden fazla sensörün senkronize kullanılmasıyla etkili sonuçlar elde edilebilmektedir [70, 71]. Bu yöntemin en önemli avantajı yalnızca dokular üzerindeki silüet bilgilerine bakılmasının yeterli olmasıdır. Fakat bu yöntemlerin hesaplama maliyetleri yüksektir ve senkronize kamera sayısına bağlı olarak farklı açılardan elde edilen seyrek veya karmaşık dokulardaki tıkanıklıklar, problemlere neden olmaktadır [72].

Bir başka çalışmada tek renkli bir görüntü ve buna karşılık gelen derinlik haritasından yüksek kalitede kişiselleştirilmiş bir avatar hesaplamak için Kinect kullanılmaktadır [73]. Gerçek zamanlı yakalanan yüz ifadeleri sayısallaştırılarak belirli karakterler ile eşleştirilmektedir. Özellikle son yıllarda yine tek bir Kinect kullanarak insan vücudunu tahmin etmek için siluet görüntüleri ve derinlik verilerini kullanan SCAPE (Shape Completion and Animation of People) modeli tanıtılmıştır [74]. Ancak bu yöntem poz ve şekil deformasyonlarını ayrı olarak ele almaktadır, bu durum, kişiye bağlı olduğundan dolayı doğru değildir. Buna karşın tensor-tabanlı model yaklaşımı şekil ve poz deformasyonlarını birlikte ele almaktadır [75].



Şekil 4.1. Tarama sistemi kurulumu

Tez çalışmasında, şekil 4.1’de gösterildiği üzere, tek bir sabit Kinect algılayıcı kullanarak, kapalı bir mekandaki 3B nesnenin tam taramasını gerçekleştiren bir sistem tanıtılmaktadır. Bunun yanında 360 derecelik tarama için kullanılan geometrik hizalama ile birlikte nokta bulutlarına bir azaltım ve düzeltme işlemi uygulanmıştır. Böylece makul bir kaliteye sahip ve normal tarama verilerine göre çok daha az sayıda, düzenli veri içeren kapalı bir 3B nesne oluşturulmaktadır.

Uygulamada, ilk olarak sabit bir Kinect karşısındaki nesnenin bir döner tabla yardımıyla, ihtiyaca göre farklı açılardan tarama işlemi yapılmaktadır. Tarama sayısı nesne taramasındaki tıkanıklık durumuna göre değişmektedir. Taramalar toplamda 360 dereceye tamamlanan açılardan elde edilmektedir. Birleştirildiklerinde farklı taramaların ortak bölgelerinde veri sayısının çok olmasında bir sakınca yoktur. Çünkü örtüşen bölgeler için daha sonra bir düzeltme ve arıtma işlemi uygulanmaktadır. Geometrik dönüştürme işlemi için her bir tarama verisi, merkezi (0,0,0) olacak şekilde ötelenmektedir. Bu işlem doğru



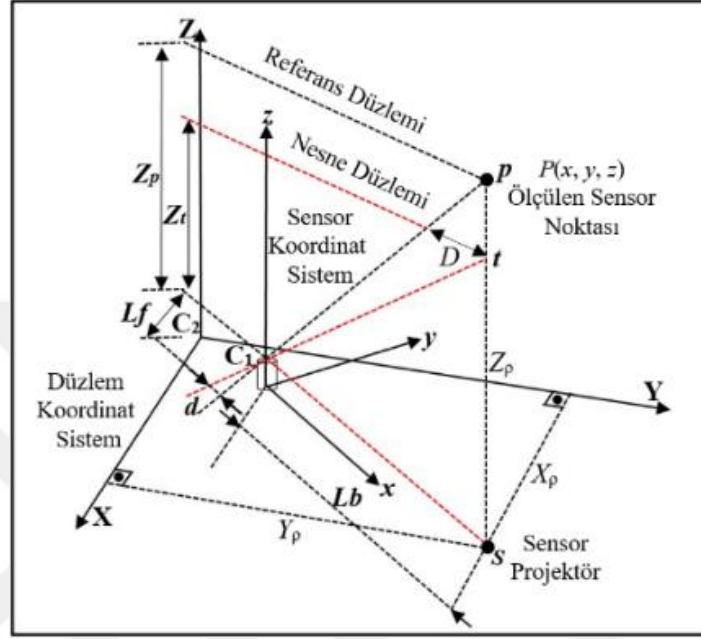
nokta dönüşümünü sağlayacaktır. Farklı açılardan elde edilen bir dizi anahtar pozları, her birinin üzerinden seçilen hizalama anahtar noktaları aracılığıyla eşleştirilerek, birbirine hizalanmaktadır. Hizalamanın ardından, elde edilen tam 3B taramaya bir dizi düzeltme ve azaltım işlemi uygulanarak, veriler daha doğal bir yapı haline getirilmektedir.

#### 4.1. 3B Kayıt İşlemi

3B nesne tarama yöntemlerinde kullanılan teknolojiler genellikle ya lazer ya da yapısal ışık tarayıcıları gibi aktif sistemleri veya doğrudan görüntüler üzerinde çalışan pasif sistemleri kullanmaktadır [76]. Yeni tarama yaklaşımlarında, video oranında derinlik görüntülerinin elde edilmesi sırasında kullanılan 3B ya da derinlik algılayıcıların yetenekleri, ön plana çıkmaktadır. Kinect algılayıcılar, ToF kameralara nazaran daha düzenli veri ve nispeten daha az gürültülü bir tarama sunmaktadır. Bu yapıdaki taramaları birbirine kaydetmek için ICP, RANSAC ve türevleri veya global katı hizalama teknikleri kullanılmaktadır [77, 78]. Katı hizalamanın ardından tek bir 3B yapı oluşturmak için birleştirme yöntemi uygulanabilir [79, 80]. Bazı yaklaşımlarda, elle taşınabilen kameraların, sabit duran 3B nesne etrafında serbestçe dönmesiyle bir prosedür uygulanmıştır. Bazılarında ise tam tersine, sabit duran algılayıcılar karşısında örneğin bir döner tabla ile döndürülen 3B nesnelerin, farklı açılardan yakalanan taramaları ile çalışılmıştır [81].

Bu bölümde, derinlik bilgileri yardımıyla gerçek nesnelerin geometrik yüzeylerine ait nokta bulutu kümesinin oluşturulması yer almaktadır. Nokta bulutu yakalama işlemi Visual Studio .Net ortamında, C# programı ile yürütülmüştür. Nokta bulutu verilerinin üretilmesinde açık kodlu bir kütüphane olan nokta bulutu kütüphanesinden (PCL) yararlanılmıştır. PCL ile filtreleme, yüzey yeniden yapılandırma, model oluşturma, sınıflandırma, 3B kayıt, nesne algılama ve tahminleme uygulamalarını gerçekleştirecek algoritmalar kullanılabilmektedir. Şekil 4.2’de,  $t$  noktasındaki bir nesnenin sensöre olan uzaklığı ve referans düzlemiyle olan ilişkisinde ölçülen dengesizlik durumu  $d$  gösterilmektedir. Referans düzlemi üzerindeki bir  $P$  noktasında, sensöre belirli bir mesafe  $Z_p$  uzaklıktaki nesnenin bulunduğunu farz edelim. Eğer bu nesne sensör kameradan belirli bir mesafe kadar uzaklaşır ya da yaklaşır, görüntü düzlemi üzerindeki konumu  $x$  düzlemi boyunca yer değiştirecektir. Nesne düzlemi üzerindeki bir  $t$  noktasının görüntü düzlemi üzerinde ölçülen aykırılığı  $d$  kadardır ve nesne düzlemi üzerinde  $D$  kadar mesafe yer değiştirmektedir. 3B derinlik bilgileri hesaplanırken kullanılan sistem üzerindeki

parametreleri açıklamak gerekirse;  $Z_p$ , referans düzlemine olan uzaklık,  $Z_t$ , nesnenin derinlik mesafesinin  $t$  noktasına olan uzaklığını göstermektedir,  $L_f$ , kızılotesi kameranın odak uzaklığı,  $L_b$ , taban uzunluğu,  $\delta x$ ,  $\delta y$ , lens bozulma katsayıları ve  $x_p$ ,  $y_p$ , temel ofset noktalarıdır [82].



Şekil 4.2. Tarayıcı ölçümlerinde koordinat eksenleri ve sensör arasındaki ilişki

$$\frac{D}{L_b} = \frac{Z_p - Z_t}{Z_p}, \quad \frac{d}{L_f} = \frac{D}{Z_t} \quad (4.1)$$

Denklem 4.1 kullanılarak,  $D$  verisi,  $Z_t$  gibi farklı parametreler ile ifade edilirse;

$$Z_t = \frac{Z_p \cdot L_f \cdot L_b}{L_f \cdot L_b + Z_p \cdot d} \quad (4.2)$$

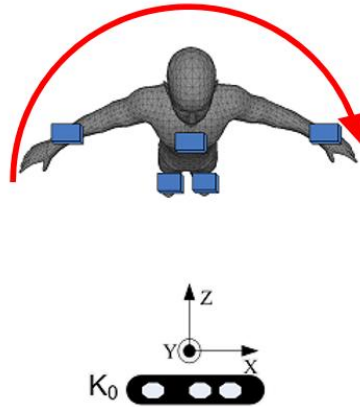
Denklem 4.2’de, sensör ile zemine ait matematiksel model koordinat sistemi arasında gözlenen derinlik eşitsizliği için kalibrasyon ile tespit edilebilir sabit parametreler  $Z_p$ ,  $L_f$  ve  $L_b$  türetilmiştir. Böylece her bir noktanın basit geometrik şekil koordinatları ile nesnelerin görüntü koordinatları, ölçek değerleri üzerinden hesaplanabilir;

$$X_t = -\frac{Z_t}{L_f} (x_t - x_p + \delta x), \quad Y_t = -\frac{Z_t}{L_f} (y_t - y_p + \delta y) \quad (4.3)$$

Denklem 4.3’de  $X_t$  ve  $Y_t$  bir nokta bulutunun (PCL) görüntü koordinatları,  $x_p$  ve  $y_p$ , ana noktanın koordinatları,  $\delta_x$  ve  $\delta_y$ , kamera objektifi bozulmaları için yapılan düzeltmelerdir. Bu hesaplamalar dikkate alınarak bir sahne içerisindeki geometrik şekillere sahip olan farklı nesnelere ait veri setleri elde edilmektedir.

3B nesne tarama ve modelleme çalışmalarında, kullanılan kamera sayısına ve nesnenin hareket durumuna göre, kabaca dört farklı senaryo ile sınıflandırılmaktadır. Bu farklı yaklaşımlar çalışma alanlarının gelişerek farklı araştırmaların doğmasına yol açmaktadır. Bu senaryolar şu şekilde kategorileştirilebilir; 1) Tek bir sabit sensör ile hareketli nesne, 2) Tek bir hareketli sensör ile sabit nesne, 3) Çoklu sabit sensörler ile sabit nesne ve 4) Çoklu sabit sensörler ile hareketli nesne.

• **Tek bir Sabit Sensör ile Hareketli Nesne:** Bu tarz yaklaşımlarda genellikle derinlik tabanlı istatistiksel öğrenme yöntemleri uygulanmaktadır. Bu yöntemlerde derinlik verileri görüntü silüetleri ile eşleştirilerek bir SCAPE modele biçimlendirilmeye uydurularak şekil tahmin edilmektedir [74]. En iyi model eşleştirmelerinin öğrenilerek tahmin edildiği bir plan yürütülmektedir. Nihai model verinin kendisinden değil bir aday alt uzay eşleştirmesinden türetilmektedir. Bu yaklaşıma ait optimizasyon süreçleri nispeten yavaştır. Şekil 4.3’te tek bir sabit sensör kullanarak hareketli bir nesneyi tarama sistemi gösterilmektedir.



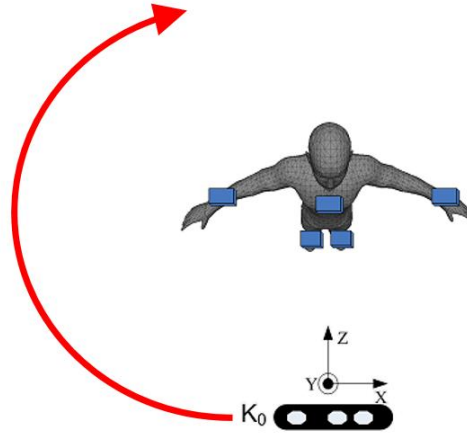
Şekil 4.3. Tek bir sabit sensör ile hareketli nesne tarama sistemi

Basit bir 3B tarayıcı öneren bir çalışmada, nesne üzerindeki hareketli bir kaleme ait gölge ile nesne şekli çıkarılmaktadır. Bu yaklaşım yalnızca durağan kurulum için uygundur. Nesne etrafında herhangi bir hareket işlemi söz konusu değildir [83]. Aktif nesne tarama çalışmalarına birer alternatif olarak SFS yeniden yapılandırma ve stereo gibi

görüntü tabanlı pasif yaklaşımlar mevcuttur. Ancak bu çalışmaların tarama eşleştirmelerindeki olumsuzlukları ve iç bükey tıkanıklıkları çoğu kez hatalara neden olmaktadır [81, 84].

Tez çalışmasında, model eşleştirme gibi öğrenme süreçli, zahmetli bir dizi işlem yerine getirmektense, hareketini bir döner tabla ile kontrol altına alabildiğimiz bir tarama sistemi gerçekleştirilmiştir. Bu yaklaşım benzerlerine göre daha hızlı ve pratiktir.

- **Tek bir Hareketli Sensör ile Sabit Nesne:** SFS tabanlı yöntemlere ait tarama yaklaşımlarında, sensör bir robot koluna monte edilerek, kolun nesne etrafında dairesel hareket etmesi sağlanmaktadır. Bu dairesel hareket esnasında takip edilen ardışık çerçeveler belirlenen bir DoF (Degree of Freedom) temel matrisi ile sınırlandırılmaktadır [85]. Temel matris parametrelerinin tümünün tahmin edilebilmesi için ilgili epipolar teğetlerin yapılandırma hatalarının en aza indirilmesi gerekmektedir [86]. Böylece 3B nesne modelleri bu temel matrislerden hesaplanan hareketler vasıtasıyla yeniden oluşturulabilir. Her ne kadar bu sistemle etkileyici ve doğal sonuçlar elde edilebilse de kurulumu halen pratik değildir.

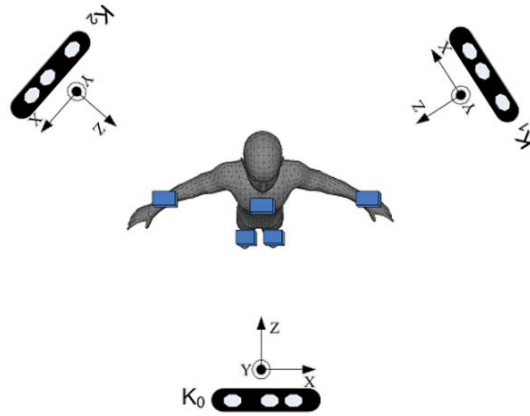


Şekil 4.4. Tek bir hareketli sensör ile sabit nesne tarama sistemi

Son dönemlerde, derinlik tabanlı yöntemlerden biri olarak ortaya çıkan KinectFusion yaklaşımı ile oldukça başarılı işler yapılmıştır [87]. Bu yaklaşımda kamera poz tahmini, hacimsel şekil gösterimi ve tekrar eden en yakın noktaya tarama hizalanması gibi işlemler yürütülmektedir [88]. Gerçek zamanlı olarak durağan bir sahnenin 3B yeniden yapılandırılması ve hacimsel gösterimine dayalı yoğun takibi gerçekleştirilebilir. Her ne kadar durağan bir sahnede çalışılsa da hareketli bir taramada noktaların birleştirilmesi sırasında başarısızlıklar meydana gelebilmektedir.

Gerçek zamanlı çalışmalarda performans sağlamak için uygulamanın GPU tarafından desteklenmesi gerekmektedir. Kinect sensör ile ToF kameralara kıyasla daha hızlı hizalama yapmak mümkündür. Çünkü sistematik bozulmalara karşı toleransı yüksektir. Bu bakımdan ToF kameralara KinectFusion yaklaşımını doğrudan uygulamak mümkün değildir. Bozulmaların önüne geçmek için süper çözünürlük gibi bazı yaklaşımlar ortaya konmuştur [78].

• **Çoklu Sabit Sensörler ile Sabit Nesne:** Araştırmacılar Kinect sensörü yaygın olarak kullanmaya başlamadan önce çeşitli sabit yoğunluklu kameralar kullanarak 3B modeller elde etmişlerdir. SFS tabanlı yaklaşımlarda, yüzey temelli yapılandırma kullanarak, bir enerji küçültme ve iyileştirme süreci sonunda, sonuçlar etkili hale getirilebilmektedir [89]. Ayrıca stereo 3B sahne yapılandırma ve SLAM (Simultaneous Localization and Mapping) tabanlı kamera izleme için yapılan çalışmalarda yoğun derinlikli haritalar eşleştirilerek önerilen yaklaşımlar bulunmaktadır [90, 91]. Fakat yaklaşım, bozuklukların giderilip yeniden düzenlenebilmesinde bir sınıra sahiptir. Benzer şekilde SFS yaklaşımlarında, nesnelere ait içbükeylerin yakalanmasında yaşanan problemler, şekil silueti ve görüntüye dayalı iyileştirmenin birleşimi ile giderilmeye çalışılmaktadır [92].

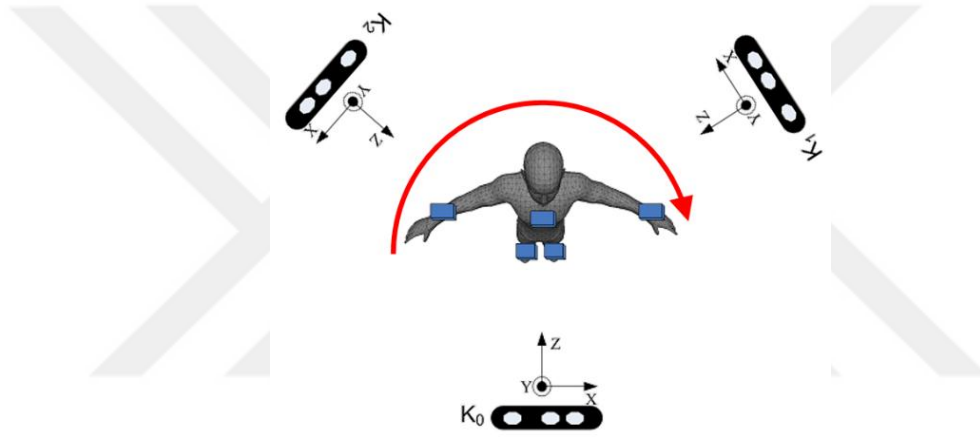


Şekil 4.5. Çoklu sabit sensörler ile sabit nesne tarama sistemi

Mesafe sensörlerinin piyasada yaygın olarak kullanılmaya başlanmasıyla araştırmacılar 3B modellemede nesne şeklini doğrudan üretebildiği çalışmalar yapmıştır. Ticari olarak üretilen, çoklu kalibre edilmiş derinlik mesafe sensörleri kullanılarak 3B nesne ve insan modelleme yapılmaktadır. Nesnelerin farklı kameralardan elde edilen nokta bulutlarının hizalanması ve insan vücudu için kişinin kamera karşısında sabit durması gerekmektedir. Her iki yöntemde şimdiye kadar doğruluğu yüksek sonuçlar elde edilmiş

olsa da birden fazla kameranın senkronize edilmesi ve bunlar arasındaki parazitlerin giderilmesi gibi etkenler bu sistemi ev ortamı uygulamalarından uzaklaştırmaktadır [93].

• **Çoklu Sabit Sensörler ile Hareketli Nesne:** Bu yöntemlerde kullanılan kamera sayısı, nesnenin hareketi ile ters orantılıdır. Hareket açısının artması sabit kamera sayısının azaltılmasını sağlar. Nesnelerin birden fazla taramalarını birleştirmek için iyi bir kayıt sağlanabilirse, kamera sayısının artırılmasında sakınca yoktur. SFS yaklaşımlarına ait bazı yöntemlerde nesneye ait renkli yüzey noktalarının (CSP-Colored Surface Point) bulunarak bunların hizalanmasıyla kayıt yapılmaktadır [94]. 6-DOF katı hareketi, her bir ardışık iki çerçeve arasındaki CSP ye ait bilgiler içermektedir.



Şekil 4.6. Çoklu sabit sensörler ile hareketli nesne tarama sistemi

Derinliğe dayalı yaklaşımlarda ICP algoritmasının artikülasyonlu versiyonu kullanılarak iyi bir kayıt elde edilebilmektedir [95]. Bununla birlikte çift yönlü katı olmayan geometrik kayıt yada global kayıt tekrarlı bir şekilde uygulanabilir [96]. Yine de sonuç olarak tüm bu yöntemler en az üç veya dört senkronize kurulmuş algılayıcı kamera kullanmayı gerektirmektedir. Ayrıca tarama sırasında, kamera karşısındaki nesnenin sabit kalması veya bir döner tabla üzerinde hareket ettirilmesi gerekmektedir.

Bilgisayar görmesi ve bilgisayar grafiği alanlarına ait çalışmalarda gerçek fiziksel 3B nesnelere ait geometrik içerik elde etmek temel bir ihtiyaçtır. Günümüzde maalesef ki 3B sahnelerle ilgili mesafe bilgilerini, gerçek zamanlı olarak, yüksek kalite ve çözünürlükte sunan, düşük maliyetli hazır bir sistem bulunmamaktadır [97]. Yüksek kalitede tarama üretebilen lazer ve yapısal ışık gibi cihazlar yüksek maliyetli ve kullanımı için özel bilgi gerektirmektedir. Örneğin Cyberware firmasına ait 3B tarama fiyatları ortalama 220.000\$'dır. Daha az maliyetli, yeni bir teknoloji sınıfı olan derinlik kameraları

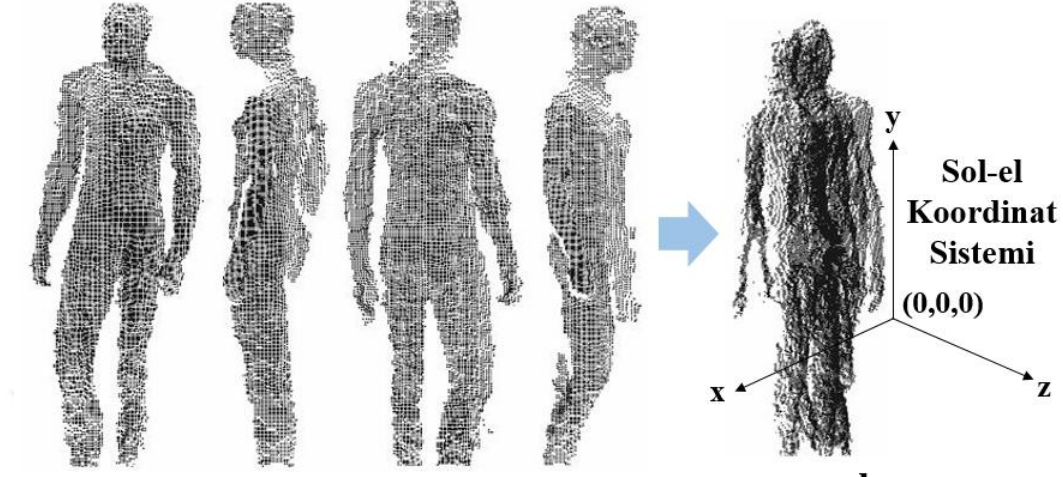
kullanılarak da tarama üretilebilmektedir. Bu sensörler derinlik ölçümü için iki farklı yaklaşım ile çalışmaktadır. Birincisi ışık atımının iletimi arasındaki zaman gecikmesini hesaplayan, uçuş süresi ToF olarak adlandırılmaktadır. SwissRanger, gerçek zamanlı olarak yüksek çözünürlüklü 3B görüntü verisi sağlayan endüstriyel bir ToF kameradır. SR4000 / SR4500 3D fiyatı yaklaşık 10.000\$'dır. İkincisi bir kızılötesi örüntü kullanmaktadır. Sahneye yansıtılan bir kızılötesi ışın örüntü deformasyonu boyunca kodlanarak örüntü derinliği hesaplanmaktadır. Bu kameralar ToF algılayıcılara göre daha düşük maliyetlidir. Çünkü CMOS (Complementary Metal-Oxide Semiconductor) veya CCD sadece kızılötesi görüntüleyici kullanmaktadır. Bu tür kameralardan olan Kinect, 2017 yılına göre yaklaşık 100\$'dır.

Bu çalışmada, tek bir sabit Kinect kullanarak, 3B tam bir tarama gerçekleştiren ev uygulamaları sistemi tanıtılmaktadır. Uygulama günlük kullanıcılar tarafından kolayca uygulanabilmektedir. Güvenilir bir şekil yakalama için sadece Kinect kullanılmaktadır. Her ne kadar basit kullanım ve uygun maliyeti olsa da Kinect ile çalışmak yüksek gürültüler ile mücadeleyi gerektirmektedir. Bu nedenle düzenli bir 3B yapı oluşturmak ve hizalamaları doğallaştırmak için bir azaltım ve düzgünleştirme algoritması uygulanmıştır. Önerilen yöntem yüksek kalitede bir veri girişi gerektirmez, sistem kurulumu basit ve kolaydır ve maliyeti düşüktür ayrıca yürütülen hesaplamaların karmaşıklığı düşüktür.

Çalışmamızın hedefi manuel hareket ettirilen bir el tipi döner tabla kullanarak, gerçek nesnelere ait 360 derece 3B nokta bulutu üreten, ToF kameraya dayalı bir 3B şekil tarayıcısı oluşturmaktır [98]. Sistem, aktif kızılötesi sistemle çalışan bir Kinect kamera ve karşısında döner tabla üzerinde duran 3B nesneden oluşmaktadır [55]. Döner tabla yardımıyla aynı nesneye ait farklı açılardan bir dizi tarama işlemi yapılmaktadır. Bu işlem sırasında, kamera sabit kalırken, döner tabla üzerindeki nesne manuel olarak döndürülmektedir. Tarama sayısı ve döndürme açısı, 3B nesnenin kapalı şekline bağlıdır. Gözle görülür bir tıkanıklık olmadığı kanaatine varılarak, tarama derecesi 360'a tamamlanır. Tabi ki bu durumda mükemmel bir tam tarama yapılması beklenmemektedir. Ancak eksiklik veya kayıp noktalar dönüşüm ve hizalama işlemlerinden sonra artırım ve düzelme işlemleri ile giderilmektedir [99].

Şekil 4.7'de insan vücuduna ait farklı açılardan oluşturulmuş ham veriler gösterilmektedir. Kamera karşısındaki 3B nesne her tarama için döner tabla kullanarak 90 derecelik açılar ile döndürülmüştür. Neticede herhangi bir tıkanıklığa maruz kalmadan,

tarama verileri 360'a tamamlanmıştır. Aynı koordinat düzlemi üzerinde, elde edilen dört farklı tarama görüntülendiğinde, ortaya çıkan çakışma gösterilmiştir.



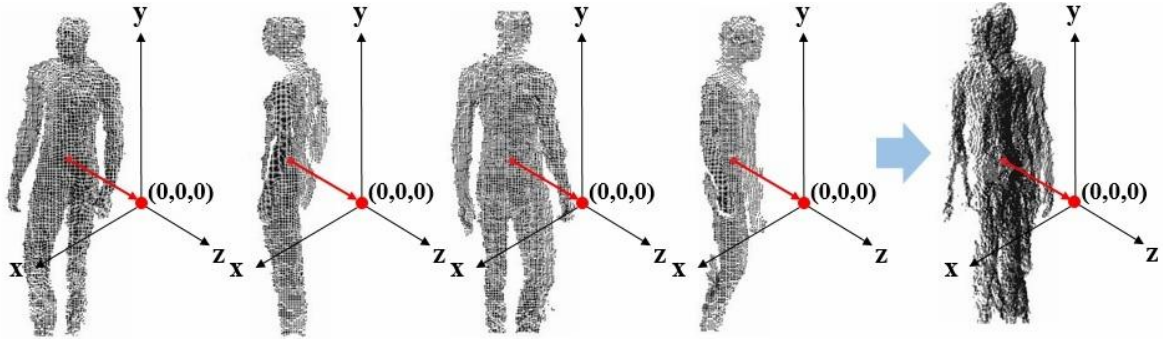
Şekil 4.7. Tarama sistemimizdeki insan vücuduna ait ham verilerin farklı açılardan tek bir Kinect ile yakalanması

#### 4.2. 3B Geometrik Dönüştürme

Katı cisim dönüşümlerinde, sadece döndürme ve çevirme işlemleri ele alınmaktadır. Bunların dışında, büyüklük ve kesme gibi düzenlemeler değiştirilmemektedir. Örneğin Affine 3B haritalama ile veriye ait her bir nokta için  $(x_1, y_1, z_1)$ , başka bir boşlukta tanımlı bir koordinat  $(x_2, y_2, z_2)$  belirlenmektedir. Çünkü bu işlemler genellikle Affine 3B dönüşümlerinde birer alt küme olarak kullanılmaktadır.

Aynı nesne için gerçekleştirilen birden fazla taramalarda nokta bulutlarının çakışma fazlalıklarının bir sakıncası yoktur. Ancak çakışmalar arasında boşluklar oluşması problemlere neden olabilmektedir. Çünkü çakışma fazlalıkları, bir dizi azaltım işlemleriyle düzenli hale getirilmektedir. Ancak çakışmalar arasında bir eksiklik olması durumunda düzeltme işlemi yapılamayacaktır. Kinect kamera ile ortalama 120 derecelik üç farklı tarama ile tam bir tarama gerçekleştirilebildiği halde, taramalar genellikle daha çok sayıda ve daha düşük açılarla gerçekleştirilmektedir.





Şekil 4.8. Ağırlık merkezi noktası orijine kaydırılan farklı taramalar

#### Algoritma-1

**Giriş:** Her çerçeve için başlangıç noktası bulutu alınır,  $P_i^{ref}, \{i = 1 \dots n\}$ .

**Çıkış:** Tüm bir çift çerçeve için 3D döndürülmüş tarama elde edilir.

$RP_i^{out}, \{i = 1 \dots n\}$ .

- Her bir çerçevenin ağırlık merkezi hesaplanır,  
Başla-  $x_{max}, y_{max}, z_{max}, x_{avg}, y_{avg}, z_{avg}$  and  $count_{max}$ .
- while**  $i \leq n$  **do**  
 $count \rightarrow count + 1$   
 $x_{max} \rightarrow x_{max} + x_i, y_{max} \rightarrow y_{max} + y_i, z_{max} \rightarrow z_{max} + z_i$   
**end while**  
 $x_{avg} \rightarrow x_{max} / count_{max}, y_{avg} \rightarrow y_{max} / count_{max}, z_{avg} \rightarrow z_{max} / count_{max}$
- Tüm noktalar ağırlık merkezi orijin olacak şekilde ötelenmektedir,  
**for**  $i = 1 \rightarrow n$  **do**  
 $\sum P_i^{ref} x_i = x_i - x_{avg}, y_i = y_i - y_{avg}, z_i = z_i - z_{avg}$   
**end for**
- Her kareye ait 3D geometrik dönüşümü hesaplanır, Formül (3).  
**return**  $RP_i^{out}, \{i = 1 \dots n\}$ .

Kinect ile gerçekleştirilen her bir taramada çok sayıda voksele sahip birer veri kümesi oluşturulmaktadır. Bu kümelerin birleştirilmesinde, yüksek seviyeli enterpolasyon çözümleri, çok sayıda voksel hesaplandığı için yavaş çalışmaktadır. Katı nokta bulutu dönüşümleri buna kıyasla daha hızlı enterpolasyon sağlamaktadır. Çünkü sadece dönüşüm işlemleri için her bir voksel belirlenen oranlarda çevrilmektedir [100].

Algoritmamızda Kinect yardımıyla elde ettiğimiz tüm verilerdeki nokta bulutu düzeni, sol el koordinat sistemine göre ayarlanmaktadır. Dönüşüm işlemlerini yerine getirmeden önce aynı 3B nesne için yakalanan her bir taramaya ait ağırlık merkezleri hesaplanmaktadır. Elde edilen bu ağırlık merkezi, koordinat sisteminin merkezinde yani orijin üzerinde olacak şekilde kaydırılmaktadır. Şekil 4.8’de gösterildiği gibi, diğer tüm

veriler de aynı doğrultuda kaydırılan miktar kadar birlikte ötelenmektedir. 3B nokta bulutu çerçevelerini hizalamaya yönelik çalışma prensibi Algoritma-1’de sözde kod olarak verilmiştir.

Merkez noktası, orijine kaydırılan veriler için orijine göre bir 3B geometrik döndürme işlemi gerçekleştirilmektedir. Geometrik dönüşüm işlemleri bir koordinat sisteminden kendisine doğru olan eşleşmelerdir. Genellikle dönüştürülen geometrik modeller Model Koordinat Sistemine (MCS) göre yer değiştirmektedir. Dönüşüm işlemleri 3B katı bir nesne için taranan ham nokta kümesi üzerine uygulanmaktadır. Dönüşüm işlemleri sonucunda nesne parçacıkları ve bunlar arasındaki bağıl mesafeler korunmaktadır. 3B homojen koordinatlar dört boyutlu konum vektörleri ile ifade edilmektedir. Bir 3B nokta olan  $P$  homojen koordinatlarda dört boyutlu bir vektör ile temsil edilmektedir;

$$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \alpha x \\ \alpha y \\ \alpha z \\ \alpha \end{bmatrix} \quad (4.4)$$

Dönüştürülen nokta kümesi  $X^*=f(P, \text{dönüşüm parametreleri})$ . Homojen koordinatlarda, 3B dönüşümler 4x4 matrisler ile temsil edilmektedir ve bir nokta dönüşümü gerçekleştirilir;

$$\begin{bmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4.5)$$

Nokta bulutlarının uzay eksenini etrafındaki keyfi döndürülme sürecinden bahsetmek gerekirse;

- Rasgele eksen, koordinat eksenlerinden biriyle aynı hizaya getirilmektedir.
- Yön kosinüsleri ( $c_x, c_y, c_z$ ) olan bir noktadan ( $x_0, y_0, z_0$ ) geçen keyfi bir eksen ele alınmaktadır.
- Dönüştürme ( $x_0, y_0, z_0$ ) koordinat merkezine (orijin) göre yapılmaktadır.
- Eksen,  $y$  eksenini ile çakışacak şekilde uygun bir şekilde döndürülmektedir.
- Nokta bulutları  $y$ -ekseni etrafında döndürülür.

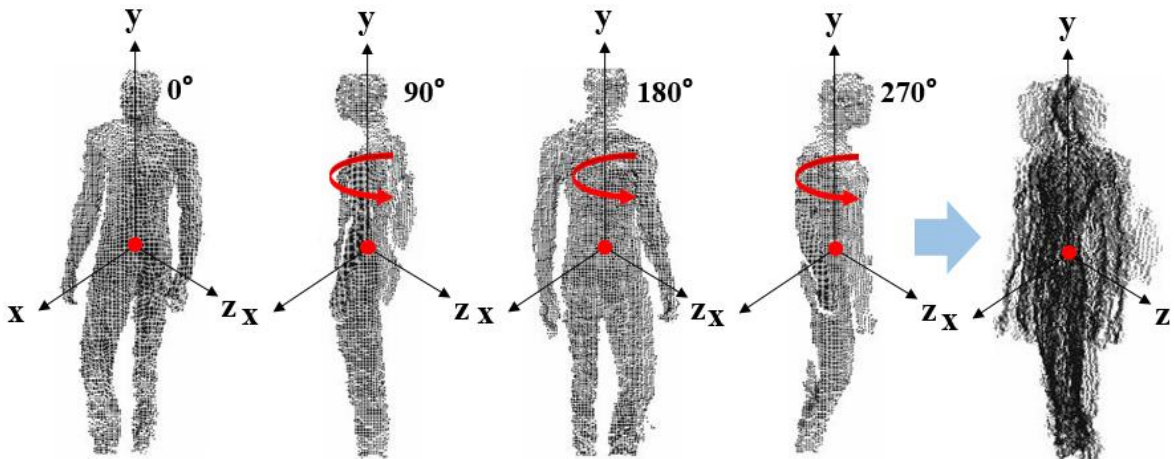
3B dönüşümlerde bir nesnenin döndürülebileceği üç dikey düzlem bulunmaktadır. Bunlar, eksenler etrafındaki dönüş düzlemleri olarak tabir edilmektedir. Radyanların bir eksen etrafındaki döndürülmesinin ilk adımı, aşağıda formüle edilmektedir;

$$[T_{Ry}] = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4.6)$$

Sol el koordinat sistemine göre ayarlanmış y-ekseni etrafında saat yönünde döndürülme işlemi;

$$p' = R_y(\phi)p \quad (4.7)$$

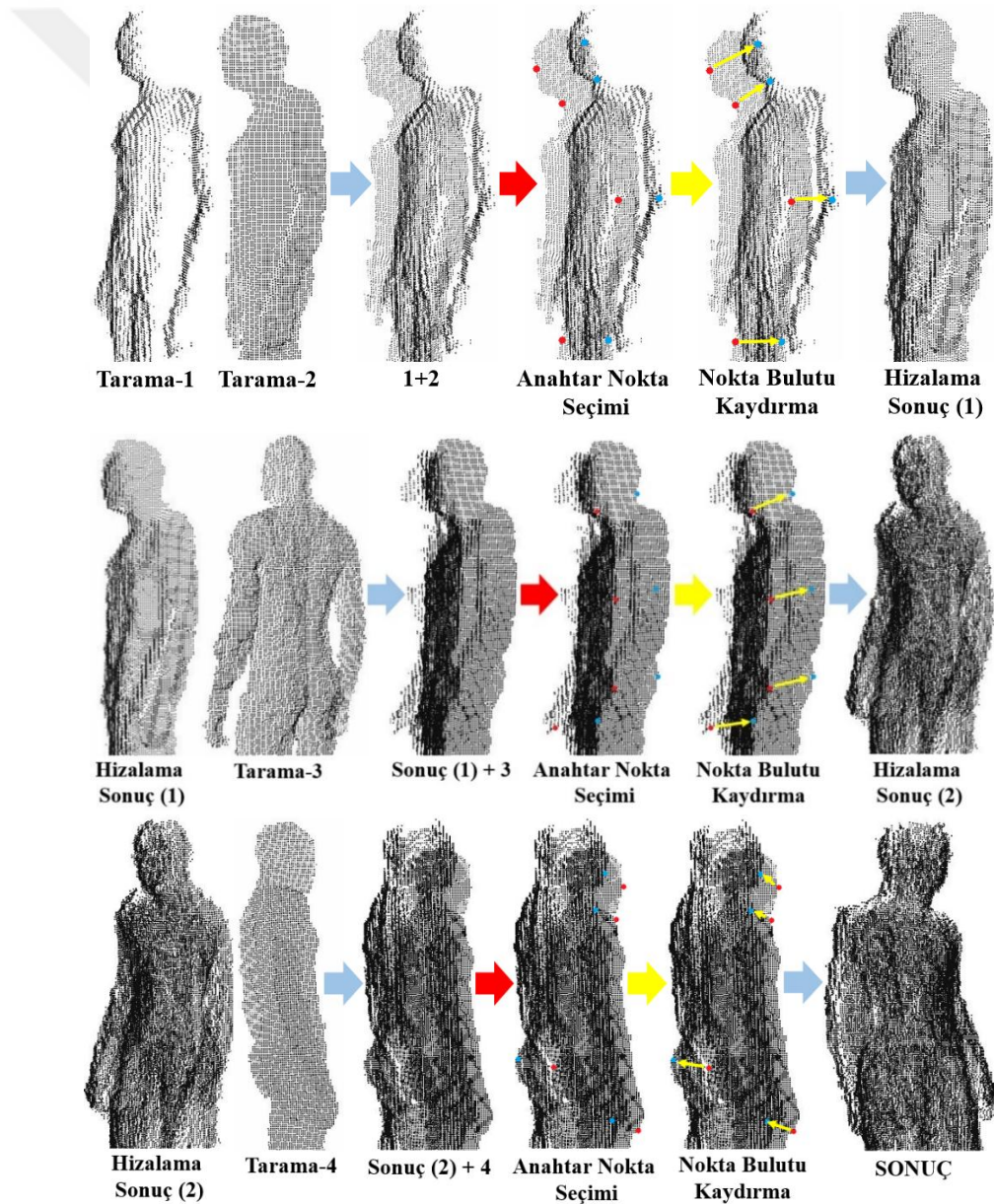
Bu matrisler uygun sırayla çoğaltılarak işlemler yerine getirilmektedir. Bu işlemlerin sırası önemlidir. Buradaki  $\phi$  açısı, nokta bulutunu tarama sırasında, tabla üzerinde döndürülen nesnenin dönüş açısıdır. Dönüş açıları programa  $\pi$  katsayısı cinsinden tanıtılmıştır. 3B nesnenin yalnızca ilk taramasına herhangi bir dönüş işlemi uygulanmaz. Şekil 4.9'da görüldüğü üzere daha sonraki taramalar için her birinin dönüş miktarı oranında dönüşüm işlemi yapılmaktadır. Birinci tarama sabit kalmaktadır. Sonraki her tarama için dönüş açısına göre, programda kullanılan açı değerleri,  $\pi/2$ ,  $\pi$ , ve  $3\pi/2$ 'dir.



Şekil 4.9. Orijin merkezine taşınan taramalarda 3B geometrik dönüşüm işlemi

### 4.3. Hizalama

Global hizalama, katı taramalar için döngü kapanma problemlerine uygulanabilen yöntemlerden birisidir [101]. Diğer bir kaba-kuvvet çözümünde, tüm taramaların ICP algoritması ile yinelenmesi söz konusudur [102]. Bu tarz sistemlere ait metotlarda genellikle büyük denklem çözümlerine başvurulmaktadır. Bir başka çözümde ise açgözlü bir yaklaşımla her yeni tarama bir önceki taramaya hizalanmaktadır. Ancak bu yöntemlerde önceki taramalara ait hatalar indirgenemeyebilir ve tutarlı bir döngü üretilmesi garanti edilememektedir. Bu nedenle taramalar arasında ikili hizalama fikri ortaya çıkmaktadır.



Şekil 4.10. Çoklu tarama için gerçekleştirilen hizalama adımları

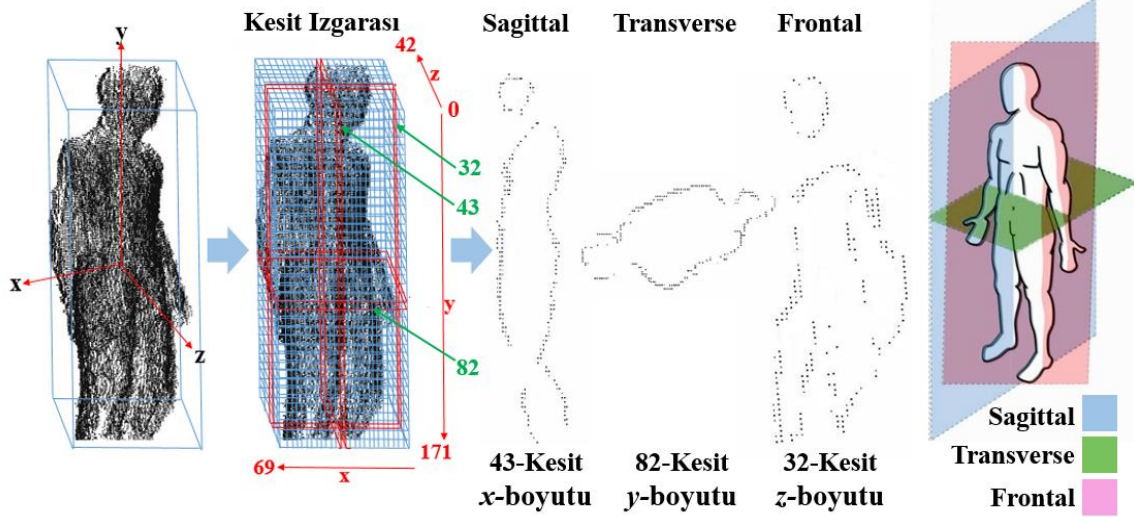
Katı hizalama için öncelikle hizalanacak çerçeveler için geometrik dönüştürme işlemleri yerine getirilmektedir. Daha sonra ardışık her iki çerçeve için hizalama anahtar noktaları belirlenmektedir. Bu anahtar noktalar kullanıcı tarafından nesnenin aykırı özellikleri arasından belirlenmektedir. Örneğin; köşe, kenar, uç nokta vb. Bu noktalar her iki çerçeve için ortak olduğu düşünülen noktalardan seçilmektedir. Belirlenen noktalar için doğrudan derinlik bilgileri işleme alınmaktadır. Dönüştürme sürecinden geçmiş olan çerçevenin anahtar noktaları, ilk çerçevenin anahtar noktaları ile eşleştirilecek şekilde, hizalama işlemi yapılmaktadır. Hizalama anahtar noktaları, ilk çerçevede ortak konumda olacak şekilde, çerçeveler birleştirilmektedir. Dönüştürülen çerçevedeki tüm nokta bulutu verisi, anahtar noktalar ile birlikte, aynı sapma miktarı kadar ötelenerek, ilk çerçeveye taşınmaktadır. Böylece ilk iki çerçeve için hizalama işlemi yapılmış bir nokta bulutu elde edilmektedir. Sonraki her bir çerçeve için aynı hizalama süreci tekrar edilmektedir. Şekil 4.10'da gösterildiği gibi, birden fazla tarama işlemi, 3B dönüştürme işleminin ardından gerçekleştirilen hizalama tekniği ile birleştirilmektedir. Hizalama için önceki taramada seçilen kırmızı renkteki noktalar, sırasıyla bir sonraki taramada seçilen mavi renkteki noktalar ile hizalanarak birleştirilmiştir. Birleşim bölgelerinde meydana gelen veri fazlalıkları herhangi bir sorun oluşturmaz. Çünkü hizalama işlemi sonrasındaki azaltım ve iyileştirme işlemleri ile veriler düzenli hale getirilmektedir. Bu yöntem çok taramalı nesne modelleme işlemleri için pratiktir ve kısa sürede etkili sonuçlar üretmektedir.

#### 4.4. İyileştirme

3B nesnelerin doğru bir şekilde kaydedilmesinden sonra, bir nokta bulutu iyileştirme ve arıtma süreci takip edilmektedir. Şüphesiz elde edilen kayıtlar mükemmel bir sonuca sahip değildir. Çünkü her ne kadar hassas ölçümler alınsa da, döner tablanın döndürülmesi manuel olarak yapılmaktadır. Bu da hizalama bölgelerindeki nokta yoğunluğun diğer bölgelere oranla daha fazla olmasına neden olmaktadır. Hedefimiz her bir voksele bir nokta bulutu yerleşecek yoğunlukta bir çerçeve oluşturmaktır. Bunun için öncelikle kayıt işlemi yapılmış tüm nokta bulutunu kapsayan bir voksel çerçevesi oluşturulmaktadır. Daha sonra, her bir kesit bir birim voksel alanını kapsayacak şekilde, ızgara üç boyutlu olarak kesitlere ayrılmaktadır. Şekil 4.11'de gösterildiği gibi kayıt yapılan nokta bulutu verileri sol el koordinat sistemine göre kesitlere ayrılmıştır. Şekilde verilen nokta bulutları  $x$  koordinatına göre 69,  $y$  koordinatına göre 171 ve  $z$  koordinatına göre 42 birim kesite bölünmüştür. Bu modele ait kayıt yapılan toplam nokta bulutu sayısı 34181'dir. Çıkarılan



kesitler üzerinde yürütülen algoritma ile hem modeli oluşturan nokta bulutu sayısı azaltılmakta, hem de noktalar daha düzenli bir yapıya kavuşmaktadır.

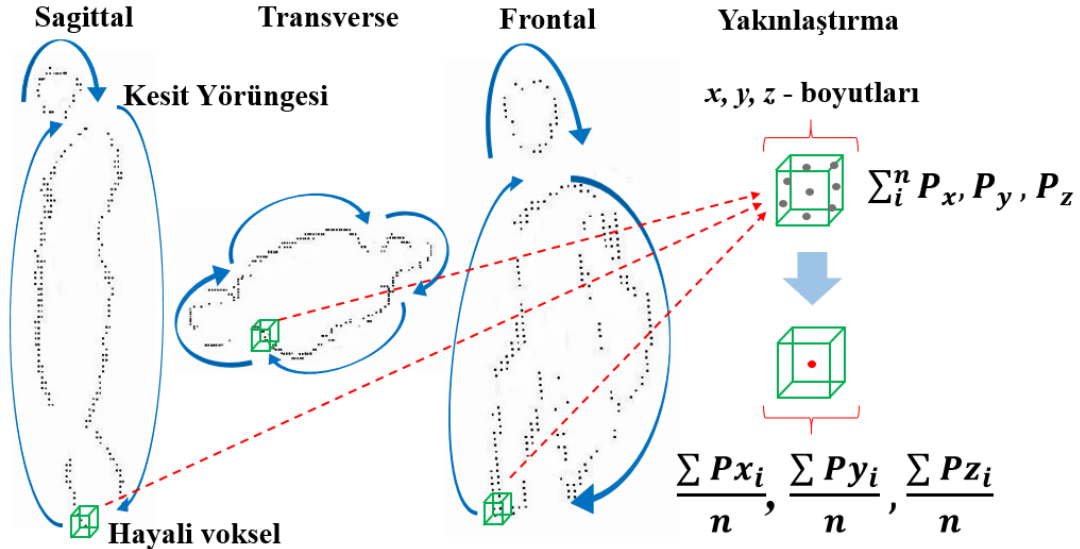


Şekil 4.11. Kesit çıkarım senaryosu ve üç-boyutlu çıkarılan kesitler

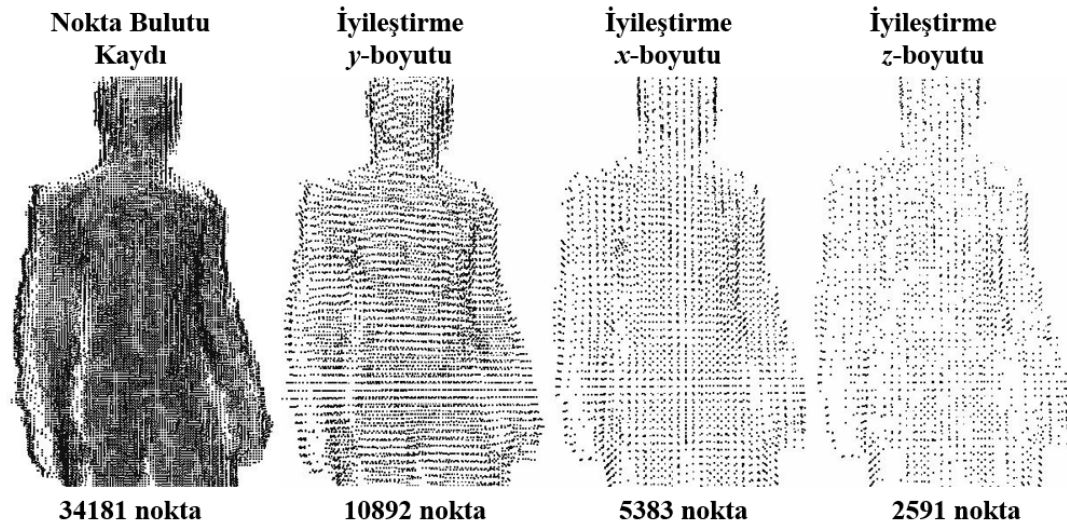
Algoritma her bir boyut için çıkarılan bütün kesitler üzerindeki nokta bulutlarına doğrudan uygulanmaktadır. Kesit çıkarma işlemi, sagittal frontal ve transverse olmak üzere üç farklı boyutta ele alınmaktadır. İyileştirme süreci bu üç farklı boyut için ayrı ayrı işleme alınmaktadır. Öncelikle her bir kesit üzerinde gezinerek iyileştirme yapacak hayali bir voksel oluşturulmaktadır. Bu voksel tüm kesiti bir tur dolaşmaktadır. Algoritma her bir birim voksel hareketinde, içine aldığı noktalarda iyileştirme yapmaktadır. İyileştirme hangi boyut için ele alınmış ise o boyuta ait koordinat değeri, kesitin ortalama değeri olarak atanmaktadır. Diğer koordinat değerleri, voksel içerisinde kalan tüm noktaların koordinat değerleri her bir boyutun ortalaması hesaplanarak atanmaktadır. Örneğin, şekil 4.10'daki gibi x-boyutu üzerindeki 43. kesitte yapılan bir iyileştirme için, x koordinatına kesitler arası ortalama değer olan 43,5 değeri atanmaktadır. Buradaki y ve z değerleri voksel içerisindeki tüm noktaların y ve z değerlerinin ortalaması olarak atanmaktadır. Benzer şekilde y boyutunda 82. kesit için yapılan iyileştirmede, y değeri 82,5, x ve z değerleri için voksel içerisindeki tüm noktaların x ve z değerlerinin ortalaması atanmaktadır. Aynı uygulama, z-boyutunda yapılan iyileştirme için de geçerlidir.

Şekil 4.12'de üç farklı boyut için ayrılan her bir kesit üzerinde yapılan iyileştirme süreci gösterilmektedir. Buna göre her bir kesit için hayali voksel tüm noktalardan geçerek voksel içerisindeki noktaların ortalama bir değerini ilgili voksele kaydetmektedir. Kayıt

işlemi sırasında  $x$ -boyutuna ait kesitin  $x$  koordinat değerine, bulunduğu kesitin ortalama sabit değeri atanmaktadır. Geriye kalan  $y$  ve  $z$  değerleri hayali vöksel üzerinden hesaplanarak atanmaktadır. Aynı şekilde  $y$ -boyutuna ait kesitin  $y$  koordinat değerine, bulunduğu kesitin ortalama sabit değeri atanmaktadır. Geriye kalan  $x$  ve  $z$  değerleri hayali vöksel üzerinden hesaplanarak atanmaktadır. Bu durum  $z$ -boyutu için de aynıdır.  $Z$ -boyutuna ait kesitin  $z$  koordinat değerine, bulunduğu kesitin ortalama sabit değeri atanmaktadır. Geriye kalan  $x$  ve  $y$  değerleri hayali vöksel üzerinden hesaplanarak atanmaktadır. Kayıt yapılan 3B nokta bulutu üzerine uygulanan iyileştirme ve azaltma işlemine ait süreç hattı, sözde kod olarak Algoritma-2’de verilmiştir.



Şekil 4.12. Kesitlerin hayali vöksel ile iyileştirme prosedürü



Şekil 4.13. Nokta bulutu kaydı ve farklı boyutlardaki iyileştirme sonuçları

### Algoritma-2

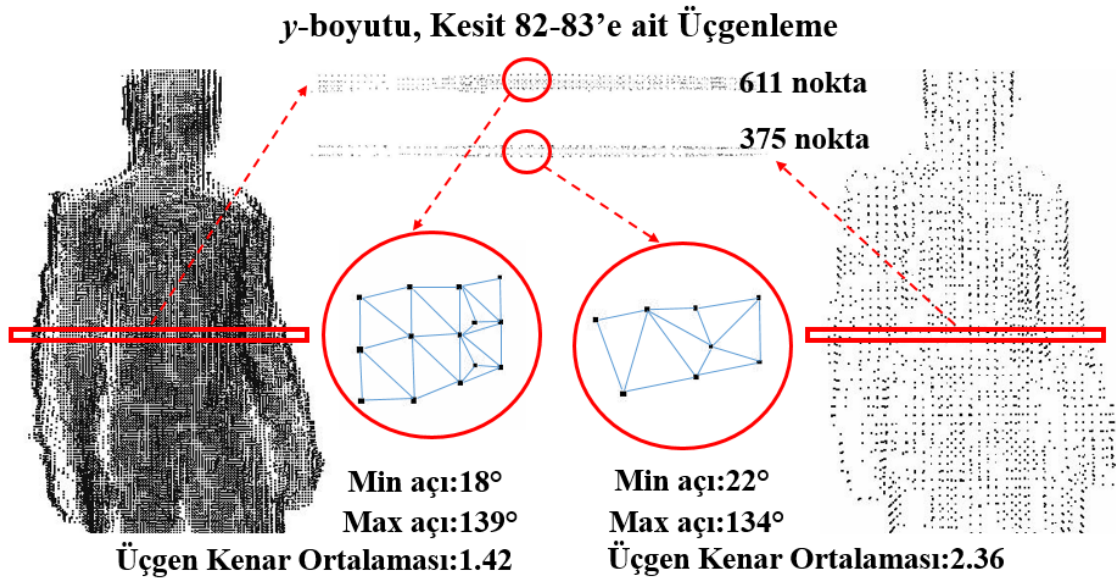
- **Giriş:** Her bir kare için referans noktası,  $P_i^{ref}, \{i = 1 \dots n\}$ .
- **Çıkış:** Geliştirilmiş nokta bulutu kaydı,  $P_i^{out}, \{i = 1 \dots n\}$ .
- Kesit Hesaplamaları,  
Sagittal ( $v_{max}, v_{min}, x_{max}, x_{min}$ ) / (x-boyutu)  
**while**  $i \leq n$  **do**  
    **if**  $x_i > v_{max}$  **then**  
         $v_{max} \rightarrow x_i$ .  
    **if**  $x_i < v_{min}$  **then**  
         $v_{min} \rightarrow x_i$ .  
**end while**  
Transverse ( $v_{max}, v_{min}, y_{max}, y_{min}$ ) / (y-boyutu)  
**while**  $i \leq n$  **do**  
    **if**  $y_i > v_{max}$  **then**  
         $v_{max} \rightarrow y_i$ .  
    **if**  $y_i < v_{min}$  **then**  
         $v_{min} \rightarrow y_i$ .  
**end while**  
Frontal ( $v_{max}, v_{min}, z_{max}, z_{min}$ ) / (z-boyutu)  
**while**  $i \leq n$  **do**  
    **if**  $z_i > v_{max}$  **then**  
         $v_{max} \rightarrow z_i$ .  
    **if**  $z_i < v_{min}$  **then**  
         $v_{min} \rightarrow z_i$ .  
**end while**
- Ayrılan her bir Sagittal kesit için nokta bulutu iyileştirme işlemi  
 $P_j^{sec}, \{j = 1 \dots m\}$ .  
Başla  $x_t, y_t, z_t$ ,  
**for**  $j = v_{min} \rightarrow v_{max}, +1$  **do**  
    **while**  $j \leq m$  **do**  
        **if**  $y_t \geq v_{min}$  and  $y_t \leq v_{min} + 1$  **then**  
             $y_t \rightarrow y_t + Py_j$ .  
        **if**  $z_t \geq v_{min}$  and  $z_t \leq v_{min} + 1$  **then**  
             $z_t \rightarrow z_t + Pz_j$ .  
    **end while**  
     $x_{avg} \rightarrow (v_{min} + 1)/2, y_{avg} \rightarrow y_t/m, z_{avg} \rightarrow z_t/m$ .  
**end for**  
 $P_i^{out}, \{i = 1 \dots n\}$ .  
return  $\{x_{avg}, y_{avg}, z_{avg}\}$ .



Şekil 4.13’de kayıtlı nokta bulutlarına uygulanan iyileştirme algoritması sonuçları sıralanmıştır. Buna göre öncelikle y-boyutu üzerinden çıkarılan kesitlere bir iyileştirme sağlanmıştır. İyileştirme adımı sonucunda kaba nokta bulutlarının sayısında ciddi oranda azalma meydana gelmiştir. Bu sayı 34181 adet noktaya sahip veri sayısının 10892’ye düşürülerek değer yaklaşık olarak 3’te 1’ine indirgenmiştir. Aynı süreç, x ve z boyutları için de uygulanarak bu değer sırasıyla 5383’e ve sonuç olarak 2591’e düşürülmüştür. Noktalar arasındaki hizalar daha tutarlı ve düzenli bir yapıya kavuşturulmuş olmakla birlikte nokta sayısında da ciddi bir azaltım sağlanmıştır. Böylece, hizalama sonucu oluşan çakışmalar ve nokta bulutu karmaşası giderilmiştir.

#### 4.5. Sonuç

Şimdiye kadar yapılan birçok çalışmada yüzey kafeslerinin genel amaçlı olarak değerlendirilebilmesi için birçok kıstas üzerinde çalışılmıştır. Bilinen yöntemlerde kullanılan kıstaslar geometrik özellikler üzerinden çıkarılmaktadır [103]. Bu özellikler için teorik yönler tanımlanarak geometrik yorumlar incelenmektedir. Yüzey kafes üzerindeki nokta bulutunun düzenlilik halini tespit etmek için noktalar arası mesafe ve üçgen kafesler arasındaki açı ortalamaları ele alınmıştır. Şekil 4.14’de kaydedilen ham nokta bulutu ile iyileştirme yapılan nokta bulutunun ikişer kesitleri arasındaki üçgelemeye ait açı değişimleri ve üçgenleme kenar ortalaması verilmektedir.



Şekil 4.14. Kayıtlı ve iyileştirilmiş nokta bulutu yüzey örgü özellikleri değişimi

Buna göre kesitlere uygulanan iyileştirme neticesinde üçgenlemelerin açıları arasındaki değişimler gösterilmiştir. Bu iki kesit için ortaya çıkan sonuçta üçgenlemenin en küçük açısı  $18^{\circ}$ 'den  $22^{\circ}$ 'ye yükselmiştir. Buna ek olarak üçgenlerin ortalama kenar uzunlukları toplamı 1.42'den 2.36'ya yükselmiştir.

Kinect ile veri alma, 3B döndürme, hizalama ve iyileştirme gibi uygulamalar için geliştirilen tüm algoritmalarda C# programlama dili kullanılmıştır. Programlama için kullanılan bilgisayar donanımına ait özellikler, 8 GB RAM, 2.40 Ghz Intel Core i7-3610QM CPU, GT650M ekran kartıdır.

Bu bölümde kapalı ortamlarda tek bir sabit Kinect kamera kullanarak 3B nesne taraması için saf kullanıcıların yönetebileceği pratik bir sistem sunulmaktadır. Nesne taramalarında kullanılan yöntemler için bir değerlendirme yapılmıştır. Sunulan yöntem doğrudan taranan nokta bulutları üzerinde çalışmaktadır. Gerçekleştirilen sistemde, mevcut yöntemlerde kullanılan karmaşık algoritmalara ve yüksek maliyetli sistemlere olan bağıllık, ortadan kaldırılmıştır. Taramadaki kayıt sonucu, iyileştirme ve arıtma işlemlerinin ardından, noktalar arası açı değişimlerine göre değerlendirilmiştir. Kayıt işlemi yapılan nokta bulutu ile iyileştirme uygulanan nokta bulutları arasındaki ortalama açı değişimleri bizlere düzenlilik ile ilgili fikir vermektedir. Bunun yanında nesne ayırt etme sürecinde, iskelet çıkarımı algoritmalarında kullanılan, tam ( $360^{\circ}$ ) bir 3B nokta bulutu kaydı kullanımına hazır bir sonuç elde edilmiştir.

## 5. İSKELET ÇIKARMA

İskeletler, nesneleri bağılı bulunduğu şekillerinden soyutlayarak tanımak ve çıkarım yapmak üzere, çeşitli uygulamalarda kullanılan yapılardır. İskelet yapısı çıkarımı 1970'li yıllardan beri, bu alanda yapılan çalışmalarda kullanılan popüler bir teknik olmuştur. Şimdiye kadarki çalışmalarda kullanılan algoritmalar genellikle model uygulamalarını ve buna ait özellik çıkarımlarını hedeflemiştir. Genel olarak çıkarım prosedürleri, geometrik ve hacimsel tabanlı olmak üzere iki temel metoda ayrılmıştır [104].

Hacimsel yöntemler genellikle girdi modelinin vokseleştirilmesine ihtiyaç duymaktadır [105]. Bu yöntemde hacimsel alanlar, Öklid mesafe ya da kapalı bir potansiyel alana karşılık gelen iç hacimler ile vokseleştirilmektedir. Özellikle girdinin geometrik şekle ait iç kısımları hakkında bilgi edinmesi istenmektedir. Bu talep eksik verilere sahip nesneler üzerinde çalışma aşamasında sıkıntı oluşturmaktadır. Bazı çalışmalarda, potansiyel alan sınırları boyunca gelen kuvvet hesaplanarak, [5] bazılarında ise modellerin sırt noktalarının tekrarlı bir şekilde budanıp, eğri iskelet yapısı ortaya çıkıncaya kadar inceltilmesi uygulanmıştır [106]. Bu yöntemde su geçirmez kapalılık özelliğinin tam olmasının bir önemi yoktur aynı zamanda düşük oranda eksik verilerin olması başarılı bir sonuca etki etmemektedir.

Geometrik yöntemler girdiyi vokseleştirmek yerine, işlemleri doğrudan ham nokta bulutları ya da çokgen kafesler üzerinden gerçekleştirir. Bunlar arasında en bilinen yöntemlerden biri olan Voronoi diyagramları eğri iskeleti oluştururken medial ekseni devre dışı bırakmaktadır. Ayrıca yapısı gereği gerçek nesnelerden taranmış olan verilerdeki gürültülere karşı toleransı yüksektir. Buna karşın nesnelerin topolojik şekillerine ait iskelet yapısını doğrudan oluşturan metotları barındırır. Birbirinden bağımsız farklı uygulamalarda çeşitli fonksiyonlar kullanarak kullanıcıdan hem doğru sınır komşuluklarını ayırt etmesi istenir hem de fonksiyonu modelin yönelimine duyarlı hale getirmesini tercih edebilmektedir. Eğri iskeleti çıkarmada kullanılan geometrik tabanlı yöntemler, nesnelere ait tüm topolojik özellikleri taşımamakta ve çok fazla işlem yapmaya neden olmaktadır [107]. Bununla birlikte bazı çalışmalarda ise eğri iskeleti oluşturmada başarılı bir geometrik yöntem olan, genelleştirilmiş dönel simetrik analizi yöntemi tüm şekiller için bir genelleştirmeye olanak tanımamaktadır [108]. Kapalı sahneler içerisindeki 3B nesnelerin ayırt edilmesi için nokta bulutu verilerinden faydalanılarak uygulanan tekniklerin başında,

nesnelerin iskelet yapısının çıkarımı gelmektedir. Çünkü iskelet yapıları, bir nesnenin en basit şekilde tanımlanabilecek en temel özelliğidir. Bu bakımdan, özellikle son yıllarda doğrudan iskelet bilgisi üzerinden nesne tanımlamanın yapıldığı mevcut birçok değerli çalışma bulunmaktadır.

Bilgisayarlı grafik alanındaki bazı uygulamalar ile karmaşık 3B fiziksel nesnelerin algılanması, ayırt edilmesi ve tanınması için çalışmalar yürütülmektedir. Nesnelerin temsil edildiği modellere ait veriler üzerinden çıkarımlar üretmek, son yıllardaki popüler yöntemler arasındadır. Bu yöntemler içerisinde en popüler olanlardan bir tanesi de iskelet yapısı ile çalışmaktır. İskelet yapıları varlıkların fiziksel dış görünümünün birebir karşılığını yansıtmaktadır. Birçok çalışmada canlılara ait olan iskelet yapılarından bunların hangi tür canlı sınıfına ait olduğunu hatta hangi coğrafyada yaşam sürdürdüklerine dair çıkarımlar yapılabilmektedir. 3B modellere ait nokta bulutu yapısı üzerinden elde edilen eğri iskelet yapısı ile bu çıkarımlar sayesinde modellerin topolojik ve karakteristik özellikleri temsil edilmektedir [3]. 3B nokta bulutları gerçek nesnelerin uzaysal konumlara karşılık gelen sayısal verilerin birleşiminden meydana gelmektedir. Nokta bulutları üzerinden elde edilecek eğri iskelet yapısı yine bir nokta kümesi ile temsil edilmektedir. Topolojik açıdan sade ve anlaşılır bir şekilde komple 1B iskelet elde etmek için nesne eklem bölgeleri birleştirilerek tamamlanmaktadır. Nokta bulutu iskelet eğrisi bir nesnenin geometrik ve topolojik olarak soyutlanmış üç boyutlu şeklini temsil etmektedir. Eğri iskelet yapısı 3B nesneleri temsil etmek için yeterince özlü ve etkileyicidir. Ayrıca, nesnelerin topolojik yapısı bu yolla verimli bir şekilde temsil edilirken yüzey ayrıntılarından kaçınılmış ve veri işlenmesi için daha kolay bir ortam sağlanmış olmaktadır [4]. Günümüzde gerçekleştirilen nesne tanıma ve algılama uygulamalarının veri elde etme aşamasında üç boyutlu tarayıcılar yaygın bir şekilde kullanılmaya başlanmıştır. Özellikle düşük maliyeti ve kolay uygulanabilir olması gerekçeleriyle, LIDAR (Laser Imaging Detection and Ranging) ve Microsoft Kinect gibi tarayıcılar vasıtasıyla nesnelere ait 3B nokta bulutları, farklı amaçlar için kolaylıkla elde edilebilmektedir [109].

Bu bölümde, nesnelerin nokta bulutu verilerini kullanarak, mesh yapısına gerek duymadan, doğrudan eğri iskeleti oluşturmak için metriksel bir nokta bulutu işleme algoritması sunulmaktadır. Düzenli bir iskelet yapısı oluşturulması ve metriksel işlemlerin sağlıklı sürdürülebilmesi için nesnelerin nokta bulutları elde edilmesi sırasında meydana gelebilen eksiklik, tıkanıklık ve pürüz gibi durumların giderilmesinde, Junjie'nin çalışmalarından esinlenerek bir yaklaşım uygulanmıştır [110]. Bu yaklaşımda kaba nokta

bulutuna bir tekrarlama daraltma işlemi uygulanarak, ham nokta bulutuna ait iskelet yapısının düzenli bir şekilde çıkarılması sağlanmıştır. Nokta bulutu bir adım indirgenerek topolojik olarak inceltirilmiş olmaktadır. Bu aşama büyük oranda eksik veriye sahip nokta bulutu yapılarındaki iskelet çıkarımlarında imkânsız hale gelmektedir. Bu adımdan sonra düzenli iskelet yapısı üretilmesi için Huang'ın Medial eğri iskelet çıkarım algoritması hibrit bir şekilde uygulanmaktadır [111]. Böylelikle kısmen eksik veriye sahip olan nesnelerin dahi düzenli eğri iskeletinin hızlı ve etkili bir şekilde üretilmesi sağlanmaktadır. Tek açıdan elde edilmiş nokta bulutu kayıtlarına uygulanan Medial iskelet algoritması tek başına (Laplace daraltma olmaksızın) uygulandığında bazen ilk tekrarlamlarda bazı uzuvlara ait iskelet yapısının yok olmasına bazen de son tekrarlamlara doğru uzuvlardaki iskelet yapısının oluşturulamamasına neden olmaktadır. Laplace yöntemi ile nispeten noktaların Medial algoritmasının uygulanabileceği aralığa çekilmesine yardımcı olmaktadır.

### 5.1. Laplace Yöntemi

Microsoft Kinect algılayıcı yardımıyla kamera donanımının yeteneklerinden faydalanarak gerçek fiziksel nesnelerin nokta bulutu ve mesh yapısı üretilebilmektedir [112]. Bu çalışmada yürütülen algoritmalar Kinect ile elde edilecek veriler üzerinde koşturulabilmektedir. Bu açıdan her bir nesne için bir nokta bulutu seti girdi olarak algoritmaya eklenmektedir. Laplace daraltma sürecinde nesnelerin dış anahtar noktaları giriş modelin genel görünümüne etki etmeden inceltilmektedir [4]. Bunun için algoritmaya girdi olarak verilen nokta bulutu,  $P = \{p_i\}$  ilk olarak sıfır hacim değerine sahip nokta bulutu setine dönüştürülmektedir. Bu daraltmakta olan nokta bulutu ile eğri iskeletin en küçük hacimli geometrik yapısını üretmek için modelin karakteristiği bozulmadan iskelet konumunun ilk tahmini gerçekleştirilir.

Nokta bulutlarındaki daraltma aşağıdaki doğrusal sistemin çözümünün tekrar edilmesi ile hesaplanmaktadır [110];

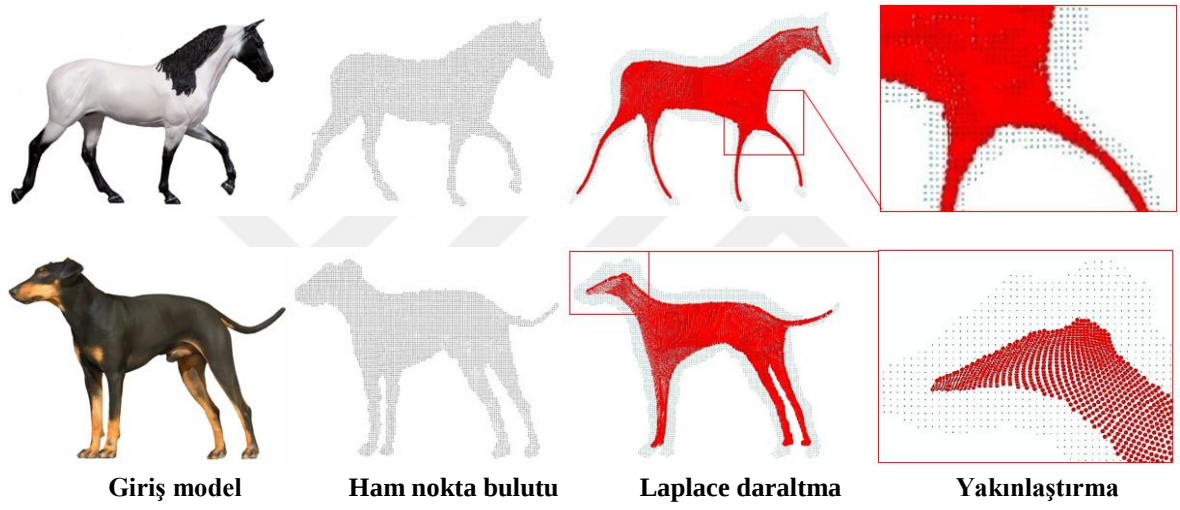
$$\begin{bmatrix} W_L L \\ W_H P \end{bmatrix} P' = \begin{bmatrix} 0 \\ W_H P \end{bmatrix} \quad (5.1)$$

Denklem 5.1'deki  $L$ ,  $n \times n$  boyutlarında bir Laplace matrisinin contangent ağırlığıdır,  $P'$  daralan nokta bulutu.  $W_L$  ve  $W_H$ , daralan ve çekim kısıtlamalarını dengeleyen çapraz

ağırlık matrisleridir.  $W_L$ 'nin  $i$ 'inci diyagonal elementi  $W_{L,i}$  olarak adlandırılır. Bu durum  $W_H$  için de  $W_{H,i}$  şeklindedir.

$$\|W_L L P'\|^2 + \sum_i W_{H,i}^2 \|p'_i - p_i\|^2 \quad (5.2)$$

Denklem 5.2 ile öncelikle Laplace daraltması kullanılarak modelin geometrik ayrıntıları ortadan kaldırılmaktadır devamında ise küçültme sırasındaki şekil geometrisi korunmaktadır.



Şekil 5.1. Ham nokta bulutuna uygulanan Laplace daraltma adımı

Modelin ayrık nokta bulutu ile tahmin edilen teğetsel düzlem alanı oluşturulması için gerekli olan tek Laplace parametresi en yakın komşuların ( $k$ ) sayısıdır. Kinect algılayıcı kamera ile yakalanmış nokta bulutu veri seti ile edinilen tecrübeye göre, yapılan test değerleri  $k=2$  sayısına göre alınmaktadır ve değerler  $[0,5..50]$  aralığındadır. Nokta bulutu olan  $P$ , iskelet eğrisi haline getirilirken, yinelemeli olarak denklem 5.1 içerisindeki  $W_L$  ve  $W_H$  güncellenerek çözüme ulaştırılır. İlk yineleme ile birlikte  $P'$ 'de önemli ölçüde daralma meydana gelmektedir. Her iterasyonda daha fazla daralma meydana gelmesi için  $W_L$  ağırlıkları komşuluk oranına göre artırılmaktadır. Ancak aşırı daralmalar yaşanmaması için çekim ağırlıkları olan  $W_{H,i}$ ,  $p_i$ 'nin çöküş derecesine göre güncellenmesi gerekmektedir. Komşuluk derecelerine göre hesaplanmakta olan bu değer yaklaşık olarak  $\min_{q \in N_k(p_i)} \|p_i - q\|$  şeklinde bulunur. Bununla birlikte daha yakın komşuluğa sahip noktalar mevcut konumlarına daha fazla çekilmektedir. Şekil 5.1'de gösterildiği üzere

farklı iki giriş için uygulanan Laplace iterasyon adımı ( $t$ ) aşağıdaki gibi uygulanmaktadır [110]; Şekilde gösterildiği gibi nesnelerin kaba nokta bulutları üzerindeki inceltme miktarı her bir iterasyon adımı için kırmızı noktalar ile ifade edilmektedir.

1-  $P^{t+1}$  için,  $\begin{bmatrix} W_L^t L^t \\ W_H^t \end{bmatrix} P^{t+1} = \begin{bmatrix} 0 \\ W_H^t P^t \end{bmatrix}$  çözümü hesaplanmaktadır,

2-  $W_L^{t+1} = s_L W_L^t$  ve  $W_{H,i}^{t+1} = W_{H,i}^0 S_i^0 / S_i^t$  güncellenmektedir, buradaki  $S_i^0$  ve  $S_i^t$  sırasıyla  $i$  noktasının orijinal ve mevcut komşuluk kapsamlarıdır.

3- Mevcut nokta bulutu  $P^{t+1}$  ile yeni Laplace operatörü  $L^{t+1}$  hesaplanmaktadır.

Hesaplama adımları kullanıcı tarafından daha önceden belirlenmiş  $t$  eşik değerine ait  $(S_i^{t+1} - S_i^t) / S_i^0 < t$ , bağıntısı gerçekleştiğinde durmaktadır. Bu çalışmada Kinect kamera ile elde edilen tüm modeller için  $t=0.1$  değerine ayarlanmıştır. Kullanılan birçok model için ortalama altı tekrarlama iskelet bütünlüğünün  $s_L = 10$  ile sağlandığı görülmektedir. Ancak iterasyon adımlarının tamamlanması zaman açısından avantajlı değildir. Bu nedenle medial iskelet çıkarma algoritmasının zamansal olarak optimum çalışması için bu tekrarın yalnızca bir kez uygulanmasının yeterli olduğu anlaşılmaktadır. Buradaki Laplace'ın koordinat ölçeği bir noktanın komşuluk ölçüsü ile orantılıdır. Bu nedenle çalışmadaki sabit çözünürlükteki Kinect modelleri için  $W_L^0 = 2/(S^0)$  ve  $W_H^0 = 5.0$  olarak ayarlanmaktadır. Buradaki  $S^0$  nokta bulutunun ortalama komşuluk kapsamıdır.

Medial iskelet algoritmasının uygulanması için bir daraltma adımı olarak uygulanan bu yöntem iterasyon adımlarının tekrar tekrar uygulanması ile 1B eğri iskeletine yaklaşan ince bir yapı oluşturmaktadır. Ancak bu adımlar sonucunda tam bir iskelet yapısı çıkarımı için iterasyon adımları artırılmalıdır. İterasyonlar ilerledikçe  $W_H$ 'ın tekrar artması sistem matrisini çapraz olarak etkili hale getirmektedir. Bu durum daraltma için istikrar sağlasa da iterasyon adımları, süreci zamansal olarak olumsuz etkilemektedir. Bu nedenle bir iterasyon sonrasında Medial iskelet daraltma algoritması yürütülmektedir. Bir iterasyon Laplace daraltma uygulanmış nokta bulutu üzerine Medial iskelet çıkarma algoritması uygulanmaktadır.

## 5.2. Medial Yöntemi

Medial iskelet çıkarma algoritması tek ve çok değişkenli ayarlanabilen basit ancak güçlü bir araçtır. Sağlamlığı, nokta bulutlarının bünyesinde gürültüler barındırmasına rağmen sağlam sonuçlar üretilmesinden kaynaklıdır. Daraltma işleminin bir iterasyon uygulanmasından sonra geometrik şekli oluşturan bir dizi nokta bulutunun yerel medianlar

ile tek boyutlu eğri iskelet yapısına dönüştüğü gösterilmektedir [111]. İnceltme sürecinin ardından organize edilmemiş bir dizi nokta bulutu  $P = \{q_j\}_{j \in J} \subset \mathbb{R}^3$  girdi olarak sisteme tanıtılmaktadır ve en uygun projeksiyon noktaları için  $X = \{x_i\}_{i \in I}$  medial tanımlamalar yapılmaktadır;

$$\underset{X}{\operatorname{argmin}} \sum_{i \in I} \sum_{j \in J} \|x_i - q_j\| \theta(\|x_i - q_j\|) + R(X) \quad (5.3)$$

Denklem 5.3'ün birinci kısmı  $P$  noktaların medyanıdır, ikinci kısımda  $X$ 'in yerel nokta dağılımı olan  $R(X)$  düzenlemeleridir. Buradaki  $I$ , yansıtılan noktalar kümesi olan  $X$ 'i,  $J$ , giriş noktalar kümesini olan  $P$ 'yi indexlemektedir. Denklemdeki  $\theta$  bir ağırlık fonksiyonu olup, iskelet oluşumu için yerel komşuluk değerini tanımlamaya yardımcı olan  $h$  destek yarıçapına sahip noktalarda hızlı kasılmalar yapan,  $\theta(r) = e^{-r^2/(h/2)^2}$  şeklinde sağlam iskelet yapısı oluşturan basit bir fonksiyondur.

Yöntemin avantajlarından bahsetmek gerekirse;

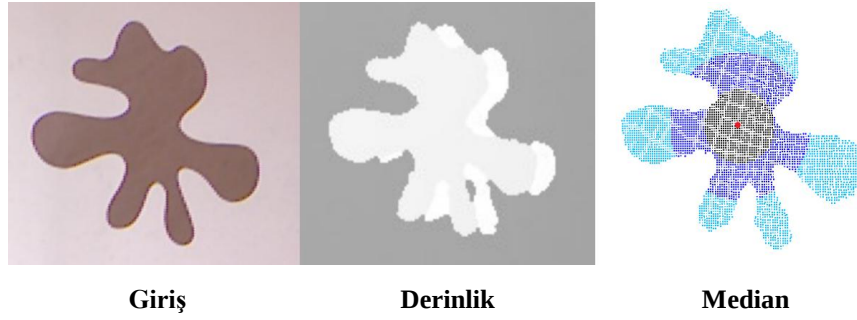
- Kinect gibi basit bir algılayıcı 3B kamera ile elde edilmiş kaba nokta bulutlarına doğrudan uygulanabilir olması ve model girdisini temsil eden nokta bulutu kümesi için kalite ve su geçirmezlik şartı aramayan nitelikte olmasıdır.
- Genel 3B şekiller üzerinde eğri iskeleti hesaplamak için kullanılan mevcut yöntemlerin başvurduğu şekil geometrisi (ör; silindirik) veya topolojisine dayalı varsayımlarda bulunmadan, aynı zamanda veri tamamlama, normal tahmini, belirsizlik ve gürültü giderme, mekânsal ayrıklaştırma, mesh yapılandırma gibi ön işleme ve filtrelemeye ihtiyaç duymamaktadır.

Medial iskelet çıkarma algoritması yürütmek için öncelikle bir  $x$  noktasının bütün veri seti ile olan minimum toplam Öklid uzaklığına sahip olması hesaplanmaktadır [111];

$$x = \underset{j \in J}{\operatorname{argmin}} \|x - q_j\| \quad (5.4)$$

Formül 5.4'teki temel tanımlama medial ağırlık merkezinin lokalize versiyonudur. Devamında lokalizasyon,  $h$  sonlu destek yarıçaplı,  $\theta$  ağırlık fonksiyonu ile eşleştirilmektedir. Burada nokta kümelerinin merkezlerinin hesaplanmasındaki avantaj medianların gürültüye duyarsız bir şekilde hesaplanabilmesidir.

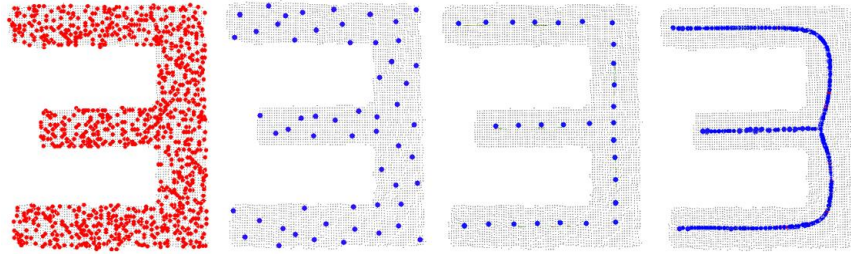




**Şekil 5.2.** Medianların verilerdeki gürültüye ve aşırılıklara duyarlılığı

Şekil 5.2’de verilerdeki medyan (kırmızı nokta) merkezine göre gürültü ve aşırılıklara olan duyarlılık gösterilmektedir. Soldan sağa sırasıyla bir RGB yamuk şekli, şekle ait derinlik görüntüsü ve nokta bulutu kümesinde orta-merkeze göre mavimsi farklı tonlardaki ağırlıklandırılmış gölgeleri yansıtılmıştır.

Medianın ham nokta bulutları üzerine tek başına uygulanması kümelerde seyrek bir yerel merkez dağılımına neden olabileceği durumu şekil 5.3(b)’de gösterilmektedir. Nesne modellerinin uygun geometrik yapısını korumak için denklem 5.3’deki  $R(X)$  kullanılarak, noktalar yerel merkez konumlarına bir kuvvet ile ötelenebilmektedir. Böylelikle nokta bulutları üzerinde koşullu düzenlilik sağlanmış olduğu şekil 5.3’te gösterilmektedir.



**Şekil 5.3.** Farklı kuvvetlere ( $\mu$ ) sahip koşullu öteleme yapılan nokta bulutları

Eğri iskeleti oluşturulurken meydana gelecek dallanmalar için her bir  $x_i$  noktasındaki  $3 \times 3$ ’lük kovaryans matrisinin öz değerleri ve kaba vektörleri denklem 5.5’teki gibi hesaplanmaktadır;

$$C_i = \sum_{i' \in I \setminus \{i\}} \theta(\|x_i - x_{i'}\|) (x_i - x_{i'})^T (x_i - x_{i'}) \quad (5.5)$$

Aşağıdaki denklem 5.6’da bulunan öz değerler  $\lambda_i^0 \leq \lambda_i^1 \leq \lambda_i^2$  ve her biri için tanımlı öz vektörler  $\{v_i^0, v_i^1, v_i^2\}$  bir nokta bulutu kümesinin ana bileşeni olan ortogonal çerçeveyi

oluşturmaktadır. Buradaki lokal komşuluğu bulunan  $x_i$ 'nin yön derecesi  $\sigma_i$ , 1'e doğru yaklaştıkça  $\lambda_i^0$  ve  $\lambda_i^1$  değerleri  $\lambda_i^2$  ile kıyaslanarak  $x_i$ 'nin çevresindeki daha fazla nokta ile birleştirilip bir dal boyunca hizalanması sağlanmaktadır [111].

$$\sigma_i = \sigma_{(x_i)} = \frac{\lambda_i^2}{\lambda_i^0 + \lambda_i^1 + \lambda_i^2} \quad (5.6)$$

Öteleme için düzenlilik fonksiyonu denklem 5.7'deki gibi tanımlanmıştır, buna göre  $\{\gamma_i\}$ ,  $X$ 'ler arasındaki sabitleri  $\{i \in I\}$  dengelemek için kullanılmaktadır;

$$R(X) = \sum_{i \in I} \gamma_i \sum_{i' \in I \setminus \{i\}} \frac{\theta(\|x_i - x_{i'}\|)}{\sigma_i \|x_i - x_{i'}\|} \quad (5.7)$$

Denklem 5.3'teki enerji gradyeni sıfır değerine sahip olduğunda her bir nokta için, sabit katsayılar eşliğinde aşağıdaki işlemler yerine getirilir [111];

$$\sum_{j \in J} (x_i - q_j) \alpha_{ij} - \gamma_i \sum_{i' \in I \setminus \{i\}} \frac{x_i - x_{i'}}{\sigma_i} \beta_{ii'} = 0, \quad i \in I,$$

Burada  $\alpha_{ij}$  ve  $\beta_{ii'}$  yerine yazılarak yeniden düzenlendiğinde;

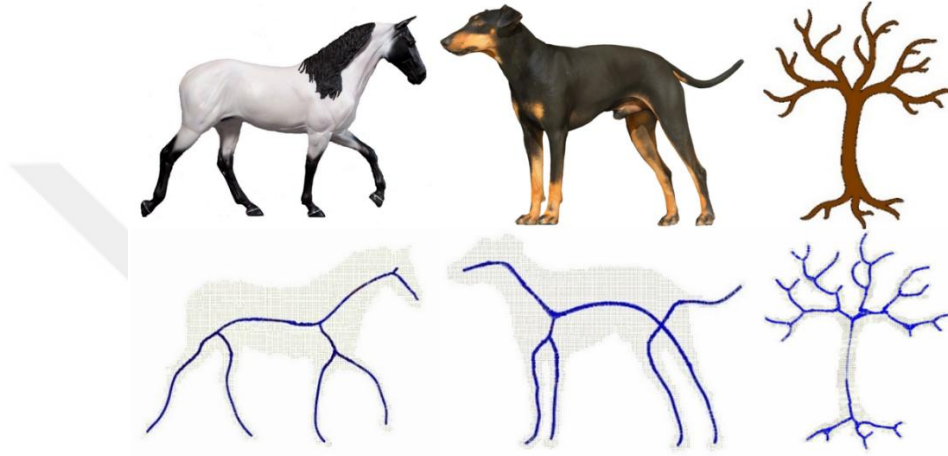
$$\alpha_{ij} = \frac{\theta(\|x_i - q_j\|)}{\|x_i - q_j\|}, j \in J, \quad \beta_{ii'} = \frac{\theta(\|x_i - x_{i'}\|)}{\|x_i - x_{i'}\|^2}, i' \in I \setminus \{i\}, \quad \mu = \frac{\gamma_i \sum_{i' \in I \setminus \{i\}} \beta_{ii'}}{\sigma_i \sum_{j \in J} \alpha_{ij}}, \forall i \in I,$$

$$(1 - \mu \sigma_i) x_i + \mu \sigma_i \frac{\sum_{i' \in I \setminus \{i\}} x_{i'} \beta_{ii'}}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}} = \frac{\sum_{j \in J} q_j \alpha_{ij}}{\sum_{j \in J} \alpha_{ij}}, \text{ elde edilir.} \quad (5.8)$$

Denklem 5.8 genel denklemi  $AX = BP$  şeklinde,  $X$ , gibi sabit bilinmeyenlere çözüm üreten bir fonksiyon denklemdir. Burada  $A$ 'nın hâkim olduğu kesin aralık olan  $0 \leq \mu \sigma_i < 1$  değerini tayin etmek için  $X = A^{-1}BP$ , çözümü sisteme dâhil edilmektedir. Mevcut yineleme göz önüne alınarak  $X^k = \{x_i^k\}, k = 0, 1, \dots$ , yinelemeleri sonraki noktalara  $\forall i \in I$  uygulanmaktadır. Böylece;

$$x_i^{k+1} = \frac{\sum_{j \in J} q_j \alpha_{ij}^k}{\sum_{j \in J} \alpha_{ij}^k} + \mu \sigma_i^k \frac{\sum_{i' \in I \setminus \{i\}} (x_i^k - x_{i'}^k) \beta_{ii'}^k}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}^k} \quad (5.9)$$

Noktaların birbirine göre hizalanması için uygulanan öteleme kuvveti hesaplanırken  $\sigma_i^k = \sigma(x_i^k)$  değeri düzenli olarak uyarlanmaktadır. Bu noktada sadece yığılan noktalara uygulanan yaptırım için parametre olarak  $\mu$  değeri [0-1) aralığında seçilmiştir. Bu çalışmadaki sonuçlar için ortalama varsayılan değer için  $\mu = 0,5$  değerine ayarlanmıştır. Kinect algılayıcı ile elde edilmiş kaba nokta bulutları üzerinden bu yöntemle elde edilen iskelet çıkarımları şekil 5.4'te üç farklı nesne için gösterilmektedir.



Şekil 5.4. Medial iskelet algoritmasının tam bir eğri iskelet çıkarımı

### 5.3. Uygulama

Gerçek nesnelere ait Kinect algılayıcı ile elde edilen kaba nokta bulutları üzerine uyguladığımız iskelet çıkarma algoritması farklı taramalar ile test edilmiştir. Kullanılan nesneler kapalı, silindirik yapıya sahip olmayan örneklerden seçilmiştir. Bu algoritma çeşitli gürültülere karşı bir ön işleme gerek duymamakta ve mevcut olan kısmen eksik verilere rağmen sonuç ortaya çıkarmaktadır. Algoritma şekil 5.4'te gösterildiği üzere kalın gövdeli hayvanlar ve çok sayıda dallanma üreten ağaç modeli gibi farklı topolojik özelliğe sahip nesneler üzerinden tecrübe edilmiştir. 1B iskelet çıkarımlarında algoritmanın sunduğu en önemli avantaj, yapısı gereği değişken başlangıç yarıçap ( $h_0$ ) değerine sahip (ör; kalın gövdeli, ince bacaklı hayvan) nesneler için inceltme işlemi ile bu değer sabit bir değer olarak tüm veriye uygulanmasına olanak sağlamasıdır. Böylece tüm veriler aynı değer ile hesaplanabilmektedir. Laplace daraltma ile bir iterasyon adımı kaba nokta bulutlarına uygulanarak Medial eğri iskelet çıkarma algoritması yürütülmüştür. İnceltme adımı, Kinect verilerinin kullanımı sırasında yerleştirilmiş median komşuluk büyüklüğünün daha küçük yarıçap değerinde doğru sonuç üretmesini sağlamıştır. Büyük

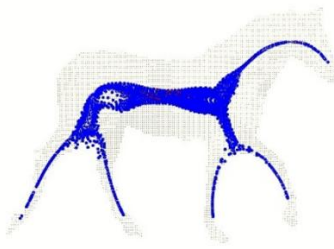
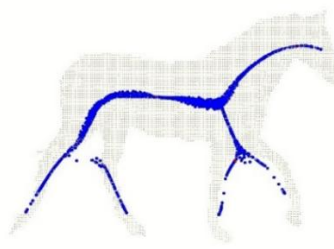
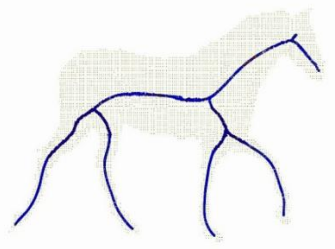
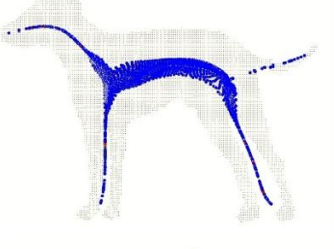
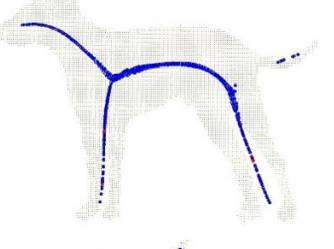
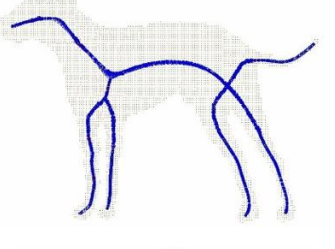
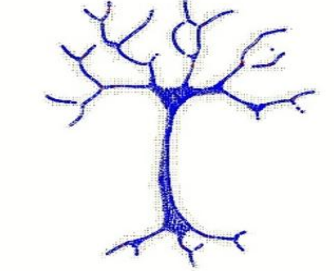
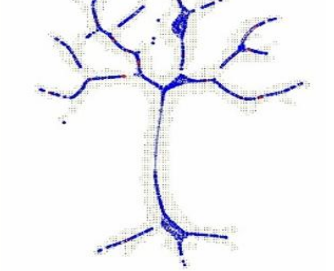
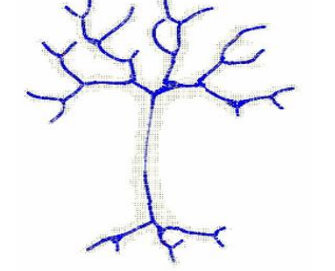
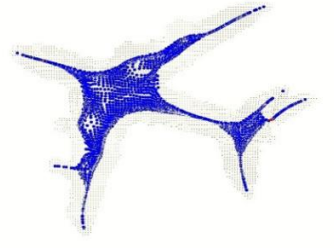
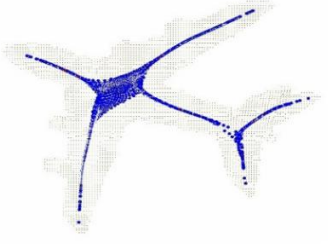
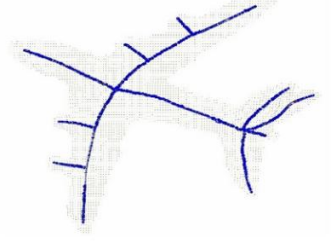
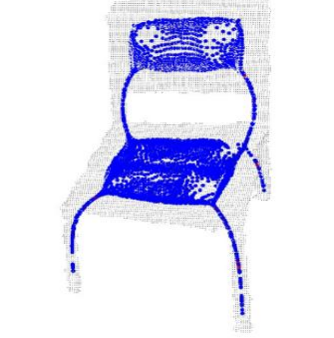
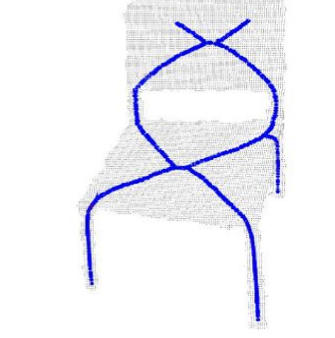
yarıçap değerlerinin kullanımının dezavantajı, nesnelerin dar kısımlarındaki verinin iskelet çıkarımlarında uzuvların yok olmasına neden olmasıdır.

Şekil 5.5’de farklı veri setleri üzerinde koşturulan birbirinden bağımsız algoritma sonuçları gösterilmektedir. İlk sütunda (a) farklı veri setleri için uygulanan Laplace inceltme algoritmasının tüm iterasyonlar sonrası sonucu gösterilmiştir. Medial algoritmasına nazaran daha zayıf bir sonuç üreten bu algoritma 1B eğri iskeleti üretimine daha uzak bir sonuç ortaya koymaktadır. İkinci sütunda (b) bulunan Medial algoritması eğri iskeleti çıkarma konusunda daha etkili olmasına rağmen aynı Kinect veri seti içerisindeki farklı nokta yoğunluğuna sahip uzuvlar için kesin bir sonuç üretememiştir. Üçüncü sütunda ise bir iterasyon inceltme işlemi uygulandıktan sonra Medial iskelet çıkarma algoritması hibrit olarak uygulanmıştır.

**Tablo 5.1.** İskelet çıkarma algoritmalarının çalışma süreleri (sn)  $t_1$ ,  $t_2$  ve  $t_3$

| Model | Verteks | $t_1$ | $t_2$ | $t_3$ | $t_4$ |
|-------|---------|-------|-------|-------|-------|
| At    | 5979    | 1,95  | 74,68 | 92,45 | 35,33 |
| Köpek | 5941    | 1,92  | 69,04 | 87,38 | 29,31 |
| Ağaç  | 4359    | 0.75  | 22,76 | 46,41 | 13,42 |
| Uçak  | 4452    | 0.77  | 23,35 | 47,49 | 13,78 |

Tüm deneysel çalışmalar için 8GB RAM 2.40 Ghz Intel Core i7-3610QM CPU, GT650M ekran kartına sahip bir donanım kullanılmıştır. Tablo 5.1’de farklı sayıdaki verteks’e sahip her bir veri seti için uygulanmış algoritmaların çalışma süreleri saniye türünden gösterilmektedir.  $t_1$ ,  $t_2$  ve  $t_3$  sütunları sırasıyla; bir iterasyon inceltme, Medial ve full iterasyon inceltme süresini,  $t_4$  ise önerilen, bir iterasyon inceltme sonrası uygulanan Medial algoritmasının süreleri toplamını göstermektedir (sn cinsinden). Önerilen bir iterasyon Laplace daraltma ve Medial algoritması, her birinin yalnız başına uygulanmasına nazaran daha kısa sürede daha sağlam ve doğal bir sonuç ortaya çıkarmaktadır.

| Veri Seti   | (a) Tam iterasyon Laplace daraltma                                                  | (b) Medial                                                                           | (c) Laplace daraltma + Medial                                                         |
|-------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| At Model    |    |    |    |
| Köpek Model |    |    |    |
| Ağaç Model  |   |   |   |
| Uçak Model  |  |  |  |
| Sand. Model |  |  |  |

Şekil 5.5. Farklı veri setleri için uygulanan yöntemlere ait sonuçlar

#### 5.4. Sonuç

Nokta bulutlarına uygulanan Laplace daraltma ve Medial iskelet çıkarma algoritmaları kullanılarak zamansal olarak daha hızlı ve güçlü sonuçlara ulaştığımız deneysel sonuçlarla ortaya konulmuştur. 3B nesnelerin iskelet eğrisi, 3B şekillerin soyut olarak, geometrik ve topolojik bir gösterimidir ve geometrik parçaların uzaysal ilişkisini, grafik yapısına göre haritalandırmaktadır. Bu iskelet yapısı, var olan nokta bulutu içerisinde elde edilmiş bir ortalama nokta bulutu verilerinin bütünleşmiş halidir. Sunulan algoritma, nokta bulutlarının ortalama metrik değerleri ile çalışmaktadır. Algoritma nesnelerden üretilen nokta bulutlarında meydana gelebilen bazı kayıp nokta bulutlarını telafi etmektedir. Bununla birlikte kısmen deforme olmuş nokta bulutu ile de bir iskelet eğrisi çıkarılabilmektedir. Böylece iskelet eğrisi yardımıyla grafik alanında kullanılan nesneleri ayırt etmek, tanımlamak ve işlemek için uygun hale gelmektedir. Sonuçta elde edilen iskelet yapısı sayesinde gürültülü giriş veya karmaşık geometriye sahip nesneler ile başa çıkabilen nesne tanıma tekniğine bir yöntem sunulmaktadır.

## 6. SONUÇ

Tez çalışmasında, doğrudan gerçek fiziksel nesnelerin Kinect algılayıcı ile elde edilmiş 3B ham nokta bulutları üzerinden eğri iskelet yapısı çıkarılması için uygulanması basit ancak sağlam sonuçlar üreten bir yaklaşım önerilmektedir. İskelet çıkarım algoritmaları genel olarak 360 derece tam bir tarama üzerinde koşturulduğunda doğal sonuçlar üretmektedir. 360 derece olmayan, tek bakış açısı ile elde edilen veriler üzerinden yapılacak iskelet çıkarımları için ham nokta bulutu verileri üzerine bir Laplace daraltma algoritması uygulanmıştır. Böylece Medial iskelet çıkarım algoritmasının uygulanabilmesi sağlanmıştır. Bu sorunun üstesinden gelmek için kullanılan daraltma işleminin yanı sıra tam bir tarama elde etmek için Kinect ile elde edilen ham nokta bulutları bir dizi geometrik dönüştürme, hizalama ve iyileştirme sürecinden geçirilmiştir. Böylece doğrudan iskelet çıkarım algoritmasının uygulanabileceği bir tarama sistemi oluşturulmuştur. Geliştirilen sistem uygun maliyetli ve uzman bilgisine gerek duymayan kapalı mekânlarda kullanılabilecek pratik bir yapıya sahiptir. Kinect tarama sistemi ile bir sahne içerisindeki nesnenin tam bir nokta bulutunu üretmek nesneyi bulunduğu sahne ayrıntılarından ayırmaya yardımcı olmaktadır. Elde edilen veriler üzerinden de iskelet bilgisinin çıkarılması, ayırt etme ve tanıma süreçleri için net bir ortam sağlamaktadır.

## KAYNAKLAR

- [1] **Yin, X., Han, J., & Philip, S.Y**, 2007. Object distinction: Distinguishing objects with identical names. In Data Engineering, IEEE 23rd International Conference on, 1242-1246.
- [2] **Bonnechere, B., Jansen, B., Salvia, P., Bouzahouene, H., Omelina, L., Cornelis, J., Van Sint Jan, S.**, 2012. What are the current limits of the Kinect sensor. In Ninth International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT), 287-294.
- [3] **Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T. L., 2001.** Topology matching for fully automatic similarity estimation of 3D shapes, Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, USA, August 12-17, 203-212.
- [4] **Au, O. K. C., Tai, C. L., Chu, H. K., Cohen-Or, D., Lee, T. Y., 2008.** Skeleton extraction by mesh contraction, ACM Transactions on Graphics (TOG), **27**, 44.
- [5] **Sharf, A., Lewiner, T., Shamir, A., Kobbelt, L., 2007.** On- the- fly curve-skeleton computation for 3d shapes, Computer Graphics Forum, **26**, 323-328.
- [6] **Bai, X., Wang, X., Latecki, L. J., Liu, W., Tu, Z., 2009.** Active skeleton for non-rigid object detection, Proceedings of the 12th IEEE International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 29-October 2, 575-582.
- [7] **Zhu, L., Chen, Y., Yuille, A., Freeman, W., 2010.** Latent hierarchical structural learning for object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, USA, June 13-18, 1062-1069.
- [8] **Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D., 2010.** Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. Proceedings of the International Symposium on Experimental Robotics, ISER 2010, New Delhi and Agra, India, December 18-21, 477-491.
- [9] **Yastikli, N., 2007.** Documentation of cultural heritage using digital photogrammetry and laser scanning. Journal of Cultural Heritage, **8**, 423-427.
- [10] **Bernardini, F. and Rushmeier, H., 2002.** The 3D model acquisition pipeline, Computer Graphics Forum, **21**, 149-172.
- [11] **Curless, B., 1999.** From range scans to 3d models, ACM SIGGRAPH Computer Graphics, **33**, 38-41.
- [12] URL-1, <http://www.smedunia.in/patjala/microindustrialsolut/product/coordinate-measuring-machine> , Koordinat ölçme makinesi. 2 Mayıs 2018.



- [13] **Park, J. J., Kwon, K., Cho, N.,** 2006. Development of a coordinate measuring machine (CMM) touch probe using a multi-axis force sensor, *Measurement Science and Technology*, **17**, 2380-2386.
- [14] **Lindner, M., Schiller, I., Kolb, A., Koch, R.,** 2010. Time-of-flight sensor calibration for accurate range sensing, *Computer Vision and Image Understanding*, **114**, 1318-1328.
- [15] URL-2, <https://www.nist.gov/image/cuvette3-800png>, Lazer sensörünün çalışma prensibi. 2 Mayıs 2018.
- [16] **Xie J., Huang S., Duan Z., Shi Y., Wen S.,** 2005. Correction of the image distortion for laser galvanometric scanning system, *Optics & Laser Technology*, **37**, 305-311.
- [17] **Çakılcı, K., Yakut, M., Kandilli, İ.,** 2005. Dijital görüntü işleme ile Lazer tarama tekniği kullanarak ürün kalitesinin belirlenmesi, EMO – III. Otomasyon Sempozyumu, Denizli, 11-12 Kasım.
- [18] URL-3, <http://www.instructables.com/id/Structured-Light-3D-Scanning>, Yapısal Işık yöntemi. 2 Mayıs 2018.
- [19] **Geng, Z. J.,** 1996. Rainbow three-dimensional camera: new concept of high-speed three-dimensional vision systems, *Optical Engineering*, **35**, 35-376.
- [20] **Huang, P. S., Hu, Q., Jin, F., Chiang, F. P.,** 1999. Color-encoded digital fringe projection technique for high-speed three-dimensional surface contouring, *Optical Engineering*, **38**, 1065-1072.
- [21] **Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., Fuchs, H.,** 1998. The office of the future: A unified approach to image-based modeling and spatially immersive displays, *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, Orlando, Florida, USA, July 19-24, 179-188.
- [22] **Hall-Holt, O., Rusinkiewicz, S.,** 2001. Stripe boundary codes for real-time structured-light range scanning of moving objects, *Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001*, Vancouver, Canada, July 7-14, 359-366.
- [23] **Morano, R. A., Ozturk, C., Conn, R., Dubin, S., Zietz, S., Nissano, J.,** 1998. Structured light using pseudorandom codes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 322-327.
- [24] **Liu, K., Wang, Y., Lau, D. L., Hao, Q., Hassebrook, L. G.,** 2010. Dual-frequency pattern scheme for high-speed 3-D shape measurement, *Optics Express*, **18**, 5229–5244.

- [25] **Wang, Y., and Zhang, S.,** 2011. Superfast multifrequency phase-shifting technique with optimal pulse width modulation, *Optics Express*, **19**, 9684–9689. doi:10.1364/OE.19.005149.
- [26] **Lathrop, R. A., Hackworth, D. M., Webster III, R. J.,** 2010. Minimally invasive holographic surface scanning for soft-tissue image registration, *IEEE Transactions on Biomedical Engineering*, **57**, 1497-1506.
- [27] **Király, Z., Springer, G. S., Van Dam. J.,** 2006. Stereoscopic vision systems, *Optical Engineering*, 45.
- [28] **Beeler, T., Bickel, B., Beardsley, P., Sumner, B., Gross, M.,** 2010. High-quality single-shot capture of facial geometry, *ACM Transactions on Graphics*, 29.
- [29] URL-4, <http://www.di4d.com/systems/di3d-system/>, DI4D-4D Capture Systems. 2 Mayis 2018.
- [30] URL-5, <http://www.3dmd.com/3dmd-systems/4d/>, 3DMD-4D Capture Systems. 2 Mayis 2018.
- [31] **Benedikt, L., Cosker, D., Rosin, P. L., Marshall, D.,** 2010. Assessing the uniqueness and permanence of facial actions for use in biometric applications, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, **40**, 449-460.
- [32] **Cosker, D., Krumhuber, E., Hilton, A.,** 2011. A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling, *Proceedings of IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13*, 2296-2303.
- [33] **Woodham, R. J.,** 1980. Photometric method for determining surface orientation from multiple images, *Optical Engineering*, 19, 139-144.
- [34] **Frankot, R. T., and Chellappa, R.,** 1988. A method for enforcing integrability in shape from shading algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, 439-451.
- [35] **Agrawal, A., Raskar, R., Chellappa, R.,** 2006. What is the range of surface reconstructions from a gradient field?, *Proceedings of Ninth European Conference on Computer Vision ECCV, Graz, Austria, May 7-13*, 578-591.
- [36] **Agrawal, A., Chellappa, R., Raskar, R.,** 2005. An algebraic approach to surface reconstruction from gradient fields, *Proceedings of Tenth IEEE International Conference on Computer Vision, ICCV 2005, October 17-21*, 74-181.
- [37] **Hansen, M. F., Atkinson, G. A., Smith, L. N., Smith, M. L.,** 2010. 3D face reconstructions from photometric stereo using near infrared and visible light, *Computer Vision and Image Understanding*, **114**, 942-951.

- [38] **Brostow, G. J., Hernández, C., Vogiatzis, G., Stenger, B., Cipolla, R.,** 2011. Video normals from colored lights, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 2104-2114.
- [39] **Lensch, H. P., Heidrich, W., Seidel, H. P.,** 2001. A silhouette-based algorithm for texture registration and stitching. *Graphical Models*, **63**, 245-262.
- [40] **Kemelmacher-Shlizerman, I., and Basri, R.,** 2011. 3D face reconstruction from a single image using a single reference face shape, *IEEE transactions on pattern analysis and machine intelligence*, **33**, 394-405.
- [41] **Wang, S. F., and Lai, S. H.,** 2011. Reconstructing 3d face model with associated expression deformation from a single face image via constructing a low-dimensional expression deformation manifold, *IEEE transactions on pattern analysis and machine intelligence*, **3**, 2115-2121.
- [42] **Blanz, V., and Vetter, T.,** 1999. A morphable model for the synthesis of 3D faces, *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, USA, August 08 – 13, 187-194.
- [43] **Brand, W.,** 2001. Morphable 3D models from video, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001*, Kauai, USA, December 8-14, 456-463.
- [44] **Blanz, V., and Vetter, T.,** 2003. Face recognition based on fitting a 3D morphable model, *IEEE Transactions on pattern analysis and machine intelligence*, **25**, 1063-1074.
- [45] URL-6, <http://mathnathan.com/2012/03/registering-two-point-clouds>, Registering Two Point Clouds. 2 Mayıs 2018.
- [46] URL-7, <http://www.meshlab.net/>, Mesh Lab. 2 Mayıs 2018.
- [47] **Rusu, R. B., and Cousins, S.,** 2011. 3d is here: Point cloud library (Pcl), *Proceedings of IEEE International Conference on Robotics and automation, ICRA 2011*, Shanghai, China, May 9-13, 1-4.
- [48] URL-8, <https://msdn.microsoft.com/en-us/library/jj131033.aspx?f=255&MSPPError=-2147217396>, Kinect for Windows Sensor Components and Specifications. 2 Mayıs 2018.
- [49] **Han, J., Shao, L., Xu, D., & Shotton, J.,** 2013. Enhanced Computer Vision with Microsoft Kinect Sensor: A Review, *IEEE Transactions on Cybernetics*, **43**, 1318–1334.
- [50] **Freedman, B., Shpunt, A., Machline, M., Arieli, Y.,** 2010. Depth mapping using projected patterns. U.S., No: 2010/0118123 dated 05.13.2010.
- [51] **Solony, M.,** 2011. Scene reconstruction from kinect motion. *Proceeding of the 17th conference and competition student EEICT 2211*, Brno University of Technology, Brno.

- [52] URL-9, <http://nicolas.burrus.name/index.php/Research/KinectCalibration>, Kinect Calibration Parameters. 2 Mayıs 2018.
- [53] **Freedman, B., Shpunt, A., Machline, M., Arieli, Y.**, 2013. Depth mapping using projected patterns, Washington, US, No: 8,493,496 dated 23 July 2013.
- [54] **Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., ..., Fitzgibbon, A.**, 2011. Kinect Fusion: real-time 3D reconstruction and interaction using a moving depth camera. Proceedings of the 24th annual ACM symposium on User interface software and technology, Santa Barbara, USA, October 16-19, 559-568.
- [55] **Cui, Y., and Stricker, D.**, 2011. 3d shape scanning with a kinect. Proceeding of the Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH '11, Vancouver, Canada, August 7-11, 57.
- [56] URL-10, <http://www.openni.ru/>, OpenNI. 2 Mayıs 2018.
- [57] URL-11, <https://www.microsoft.com/en-us/download/details.aspx?id=40278>, Microsoft Kinect SDK. 2 Mayıs 2018.
- [58] URL-12, <https://github.com/OpenKinect/>, OpenKinect. Mayıs 2018.
- [59] **Zhu, J., Humphreys, G., Koller, D., Steuart, S., Wang, R.**, 2007. Fast omnidirectional 3D scene acquisition with an array of stereo cameras, Proceedings of the Sixth IEEE International Conference on 3-D Digital Imaging and Modeling, 3DIM'07, Montreal, Canada, August 21-23, 217-224.
- [60] **Özbay, E., Çınar, A., Güler, Z.**, 2016. Classification of point cloud images. Proceedings of 4th International Conference on Engineering and Natural Sciences, ICENS 2016, Sarajevo, Bosnia and Herzegovina, May 24-28, 645-652.
- [61] **Zhang, Z.**, 1992. Iterative point matching for registration of free-form curves, PhD Thesis. Inria.
- [62] **Fischler, M. A., and Bolles, R. C.**, 1987. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM, 24, 381–395.
- [63] **Bozkurt, I., and Özden, K. E.**, 2014. Depth-scale method in 3D registration of RGB-D sensor outputs, Proceedings of IEEE International Conference on Computer Vision Theory and Applications, VISAP 2014, Lisbon, Portugal, January 5-8, 470-475.
- [64] **Lowe, D. G.**, 2004. Distinctive image features from scale-invariant keypoints, International journal of computer vision, **60**, 91-110.
- [65] **Galarza, A. I. R., and Seade, J.**, 2007. Introduction to classical geometries. Springer Science & Business Media.

- [66] **Allen, B., Curless, B., Popović, Z.,** 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (TOG)*, **22**, 587–594.
- [67] **Guarnieri, A., Pirotti, F., Pontin, M., Vettore, A.,** 2006. 3D Surveying for structural analysis applications. *Proceedings of 3rd IAG Symposium on Geodesy for Geotechnical and Structural Engineering/12th FIG Symposium on Deformation Measurement*, Baden, Austria, May 22-24, 22-24.
- [68] **Yu, X., Hyypä, J., Karjalainen, M., Nurminen, K., Karila, K., Vastaranta, M., Kukko, A.,** 2015. Comparison of laser and stereo optical, SAR and InSAR point clouds from air-and space-borne sources in the retrieval of forest inventory attributes, *Remote Sensing*, **7**, 15933-15954.
- [69] **De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H. P., Thrun, S.,** 2008. Performance capture from sparse multi-view video. *Proceedings of ACM SIGGRAPH, SIGGRAPH'08*, Los Angeles, California, August 11–15.
- [70] **Tabb, A.,** 2013. Shape from silhouette probability maps: reconstruction of thin objects in the presence of silhouette extraction and calibration error. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, Portland, USA, June 23-28, 161-168.
- [71] **Laurentini, A.,** 1994. The visual hull concept for silhouette-based image understanding, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 150–162.
- [72] **Kim, Y. M., Theobalt, C., Diebel, J., Kosecka, J., Matusik, B., Thrun, S.,** 2009. Multi-view image and tof sensor fusion for dense 3d reconstruction, *Proceedings of 12th IEEE International Conference on Computer Vision Workshops, ICCV Workshops*, Kyoto, Japan, September 27- October 4, 1542–1549.
- [73] **Zollhöfer, M., Martinek, M., Greiner, G., Stamminger, M., Süßmuth, J.,** 2011. Automatic reconstruction of personalized avatars from 3D face scans, *Computer Animation and Virtual Worlds*, **22**, 195-202.
- [74] **Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., & Davis, J.,** 2005. SCAPE: shape completion and animation of people, *ACM Transactions on Graphics (TOG)*, **24**, 408-416.
- [75] **Chen, Y., Liu, Z., Zhang, Z.,** 2013. Tensor-based human body modeling, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, June 23-28, 105-112.
- [76] **Cui, Y., and Stricker, D.,** 2011. 3D body scanning with one Kinect, *Proceedings of the 2nd International Conference on 3D Body Scanning Technologies*, Lugano, Switzerland October 25-26, 121-129.

- [77] **Senin, N., Colosimo, B. M., & Pacella, M.,** 2013. Point set augmentation through fitting for enhanced ICP registration of point clouds in multisensor coordinate metrology, *Robotics and Computer-Integrated Manufacturing*, **29**, 39-52.
- [78] **Cui, Y., Schuon, S., Thrun, S., Stricker, D., Theobalt, C.,** 2013. Algorithms for 3d shape scanning with a depth camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 1039-1050.
- [79] **Huang, Q. X., Flöry, S., Gelfand, N., Hofer, M., Pottmann, H.,** 2006. Reassembling fractured objects by geometric matching, *ACM Transactions on Graphics (TOG)*, **25**, 569-578.
- [80] **Roth, G., and Wibowoo, E.,** 1997. An Efficient volumetric method for building closed triangular meshes from 3D image and point data, *Proceedings of the Graphics Interface'97*, Kelowna, British Columbia, Canada, May 21-23, 173-180.
- [81] **Rusinkiewicz, S., Hall-Holt, O., Levoy, M.,** 2002. Real-time 3D model acquisition, *ACM Transactions on Graphics (TOG)*, **21**, 438-446.
- [82] **Khoselham, K.,** 2011. Accuracy analysis of kinect depth data, *Proceedings of the ISPRS Workshop Laser Scanning*, Calgary, Canada, August 29-31, 133-138.
- [83] **Bouguet, J. Y., and Perona, P.,** 1998. 3D photography on your desk, *Proceedings of the Sixth IEEE Conference on Computer Vision*, Bombay, India, January 7, 43-50.
- [84] **Scharstein, D., and Szeliski, R.,** 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *International Journal of Computer Vision*, **47**, 7-42.
- [85] **Wong, K. Y., and Cipolla, R.,** 2001. Structure and motion from silhouettes, *Proceedings of IEEE International Conference Computer Vision, ICCV 2001*, Vancouver, Canada, July 7-14, 217-222.
- [86] **Wong, K. Y. K., Mendonça, P. R., Cipolla, R.,** 2001. Head model acquisition from silhouettes, *Proceedings of the 4th International Workshop on Visual Form, IWVF-4*, Capri, Italy, May 28-30, 787-796.
- [87] **Izadi, S., Newcombe, R. A., Kim, D., Hilliges, O., Molyneaux, D., Hodges, S., Fitzgibbon, A.,** 2011. Kinectfusion: real-time dynamic 3d surface reconstruction and interaction, *Proceedings of the International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2011*, Vancouver, Canada, August 7-11, 23.
- [88] **Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., ..., Fitzgibbon, A.,** 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera, *Proceedings of the 24th Annual ACM Symposium on User interface Software and Technology*, Santa Barbara, USA, October 16-19, 559-568.

- [89] **Kolev, K., Klodt, M., Brox, T., Cremers, D.,** 2009. Continuous global optimization in multiview 3d reconstruction, *International Journal of Computer Vision*, **84**, 80-96.
- [90] **Newcombe, R. A., and Davison, A. J.,** 2010. Live dense reconstruction with a single moving camera, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, USA, June 13-18*, 1498-1505.
- [91] **Newcombe, R. A., Lovegrove, S. J., Davison, A. J.,** 2011. DTAM: Dense tracking and mapping in real-time, *Proceedings of IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13*, 2320-2327.
- [92] **Kutulakos, K. N., and Seitz, S. M.,** 2000. A theory of shape by space carving, *International Journal of Computer Vision*, **38**, 199-218.
- [93] **Maimone, A., and Fuchs, H.,** 2012. Reducing interference between multiple structured light depth sensors using motion, *Proceedings of IEEE Virtual Reality Short Papers and Posters, VRW 2012, Costa, Mesa, USA, March 4-8*, 51-54.
- [94] **German, K. M. A.,** 2003. 3D Reconstruction Algorithm Combining Shape-From-Silhouette with Stereo [J], *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 1265-1278.
- [95] **Chen, Y., and Medioni, G.,** 1992. Object modelling by registration of multiple range images, *Image and Vision Computing*, **10**, 145-155.
- [96] **Tong, J., Zhou, J., Liu, L., Pan, Z., Yan, H.,** 2012. Scanning 3d full human bodies using Kinects, *IEEE Transactions on Visualization and Computer Graphics*, **18**, 643-650.
- [97] **Kolb, A., Barth, E., Koch, R., Larsen, R.,** 2009. Time-of-flight sensors in computer graphics, *Proceedings of Eurographics 2009, Munich, Germany, March 30-April 3*, 119-134.
- [98] **Cui, Y., Schuon, S., Chan, D., Thrun, S., & Theobalt, C.,** 2010. 3D shape scanning with a time-of-flight camera, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, USA, June 13-18*, 1173-1180.
- [99] **Chen, L.C., Hoang, D.C., Lin, H.I., Nguyen, T.H.,** 2016. Innovative methodology for multi-view point cloud registration in robotic 3D object scanning and reconstruction, *Applied Sciences*, **6**, 132.
- [100] **Ashburner, J., and Karl J. F.,** 2007. Rigid body registration, in *Statistical parametric mapping: The analysis of functional brain images*, 49-62.
- [101] **Battle, E., Matabosch, C., Salvi, J.,** 2007. Overview of 3d registration techniques including loop minimization for the complete acquisition of large manufactured parts and complex environments, *Proceedings of IEEE/SPIE 8th International Conference on Quality Control by Artificial Vision, QCAV'07, Le Creusot, France, May 23-25*.

- [102] **Makadia, A., Patterson, A., Daniilidis, K.,** 2006. Fully automatic registration of 3D point clouds, *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE, 2006.
- [103] **Frey, Pascal J., and Houtman Borouchaki.,** 1998. Geometric evaluation of finite element surface meshes, *Finite Elements in Analysis and Design*, **31**, 33-53.
- [104] **Cornea, N. D., Silver, D., Min, P.,** 2007. Curve-skeleton properties, applications, and algorithms, *IEEE Transactions on Visualization and Computer Graphics*, **13**, 530-548.
- [105] **Fang, S., and Chen, H.,** 2000. Hardware accelerated voxelization, *Computers & Graphics*, **24**, 433-442.
- [106] **Svensson, S., Nyström, I., Di Baja, G. S.,** 2002. Curve skeletonization of surface-like objects in 3D images guided by voxel classification, *Pattern Recognition Letters*, **23**, 1419-1426.
- [107] **Verroust, A., and Lazarus, F.,** 2000. Extracting skeletal curves from 3D scattered data. *The Visual Computer*, **16**, 15-25.
- [108] **Tagliasacchi, A., Zhang, H., Cohen-Or, D.,** 2009. Curve skeleton extraction from incomplete point cloud, *ACM Transactions on Graphics (TOG)*, **28**, 71.
- [109] **Nguyen, A., and Le, B.,** 2013. 3D point cloud segmentation: A survey, *Proceedings of 6th IEEE International Conference on Robotics, Automation and Mechatronics, RAM 2013, Manila, Philippines, November 12-15*, 225-230.
- [110] **Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., Su, Z.,** 2010. Point cloud skeletons via laplacian based contraction, *Proceedings of IEEE International Conference on Shape Modeling, SMI 2010, Aix-en-Provence, France, June 21-23*, 187-197.
- [111] **Huang, H., Wu, S., Cohen-Or, D., Gong, M., Zhang, H., Li, G., Chen, B.,** 2013. L1-medial skeleton of point cloud, *ACM Transactions on Graphics*, **32**, 65-1.
- [112] **Ozbay, E., and Cinar, A.,** 2013. 3D reconstruction technique with kinect and point cloud computing, *Global Journal on Technology*, **3**, 1748-1754.



## ÖZGEÇMİŞ

**Adı Soyadı:** Erdal ÖZBAY

**Doğum Yeri ve Tarihi:** Elazığ, 13.01.1986.

**Adres:** Fırat Üniversitesi, Bilgisayar Mühendisliği, 23119, Elazığ.

**E-Posta:** erdalozbay@firat.edu.tr

**Lisans:** Girne Amerikan Üniversitesi, Bilgisayar Mühendisliği Bölümü (2005 - 2010)

**Yüksek Lisans:** Fırat Üniversitesi, Fen Bilimleri Enst., Bilgisayar Mühendisliği, Yazılım ABD. (2010 - 2013)

**Doktora:** Fırat Üniversitesi, Fen Bilimleri Enst., Bilgisayar Mühendisliği, Yazılım ABD. (2013 - )

**İş Tecrübeleri:** Fırat Üniversitesi, Bilgisayar Mühendisliği, Araştırma Görevlisi (2012 - )  
Karabük Üniversitesi Bilgisayar Mühendisliği, Araştırma Görevlisi (2011 - 2012).