

## Nokta Bulutu Verileri ile Nesne Modellerini Sınıflandırmak için Metrik Bir Yaklaşım

Erdal Özbay<sup>1</sup>, Ahmet Çınar<sup>2</sup>

<sup>1,2</sup> Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 23119, Elazığ/TÜRKİYE.

e-posta: erdalozbay@firat.edu.tr, acinar@firat.edu.tr

### Özet

3B nokta bulutu verilerini kullanarak nesne sınıflandırma araştırma alanında gelişmekte olan bir konudur. Nesne algılama ve tanıma bilgisayar grafiği çalışmalarında başlıca sorunlardan bazılarıdır. Bu çalışmada metrik bir yaklaşım ile gerçek nesnelere ait 3B nokta bulutu verilerinin etkin şekilde tespiti ve sınıflandırılması için yeni bir yöntem önermekteyiz. Yöntemin verimliliği gerçek 3B nokta bulutu verilerine göre doğrulanmaktadır. Sınıflandırma aşamasının sonuçlarını iyileştirmek için nokta bulutu verileri ve metrik bilgilerin kombinasyonu bir yazılım çerçevesinde uygulanmıştır. Birbirinden belirgin ölçüde ayrılan, farklı nesnelere ait 3B nokta bulutu verilerinin sınıflandırılması için sağlam segmentasyon ve metrik özellik çıkarımları değerlendirilmiştir. Farklı nesnelere ait verileri ile metrik sınıflandırma işlemi yürütülerek sınıflandırılmış veriler karşılaştırılarak başarı değerlendirilmiştir. Değerlendirme sonucunda, basit düzlemsel şekle sahip farklı nesnelere otomatik olarak sınıflandırılmasına ilişkin metrik sınıflandırma algoritmasının verimli bir sürece sahip olduğu gösterilmiştir.

### Anahtar kelimeler

Nokta bulutu;  
Sınıflandırma; Kinect;  
Nesne tanıma; 3B  
bölümleme;  
Modelleme.

## A Metrical Approach to Classification of the Object Modelling with Point Cloud Data

### Abstract

Object classification using 3D point cloud data is an emerging issue in the field of research. Object detection and recognition are some of the main problems in the computer graphics works. In this study, we propose a novel method with a metrical approach for effectively detecting and classification of 3D point cloud data from real objects. The efficiency of the method is validated according to the real 3D point cloud data. The point cloud data and the combination of metric information has been implemented in a software framework to improve the results of the classification phase. Robust segmentation and metrical feature extractions were evaluated with different objects which markedly separated from each other for classification of 3D point cloud data. Success is evaluated by comparing the raw data of different objects with classified metric data which executed classification process. In the evaluation, the metric classification algorithm has been shown to be efficient process for the automatic classification of different objects which have a simple planimetric shape.

### Keywords

Point cloud;  
Classification; Kinect;  
Object recognition; 3D  
segmentation;  
Modelling.

© Afyon Kocatepe Üniversitesi

### 1. Giriş

Gerçek dünyadaki nesnelere ait 3B modellerini oluşturmak bilgisayar alanında önemli bir yer tutmaktadır. Son yıllarda bilgisayarlı animasyon, bilgisayar oyunları, sanal ve artırılmış gerçeklik, bilgisayar insan etkileşimi gibi birçok bilgisayarlı grafik uygulamaları alanında gerçek nesnelere kullanmaya başlamıştır (Duarte et al. 2014). Ancak, 3B modellemenin, hassas, zaman alıcı, masraflı ve zahmetli bir süreç içermesinden dolayı, yaygın

olarak kullanılması ileri düzeyde bir artış göstermemektedir.

Gerçek nesnelere ait 3B modellerinin oluşturulmasında, bu nesnelere ait özellik çıkarımlarında, nesne tanıma ve nesne bulma gibi birçok uygulamada, Reconstruction (Yeniden Yapılandırma) yöntemleri içerisinde yaygın olarak kullanılmaya başlanan Point Cloud (Nokta Bulutu), iki boyutlu resim veya görüntü üzerinde yaptığımız işlemleri üç boyutlu sayısal ortamda yapmamızı sağlamaktadır (Guo et al. 2015). Nokta bulutları,

fiziksel nesnelerin, uzaysal düzlemdeki kapladıkları alanlara ait kütsel koordinat bilgilerinin bilgisayar ortamındaki sayısal karşılıklarıdır. Bir bütün şeklinde oluşturulan bu nokta bulutu yapısı birbiri ile ilişkilendirilmiş olan noktaların birleştirilmesi şeklindeki Mesh (Örgü) yapısını oluşturmaktadır (Bischoff et al. 2002).

Nokta bulutu kümeleriyle elde edilen nesnelere noktaları birleştiren üçgenler, çizgiler ve yüzeyler gibi geometrik varlıkların bağdaştırılması ile anlamlı hale getirilmektedir. Mevcut literatür çalışmalarında süreç, görüntülerdeki üç boyutlu nesnelere algılamaya geldiğinde ise çeşitli zorluklarla karşılaşmaktadır. Örneğin görüntüdeki hareket eden nesneyi tespit etmek ve müteakiben onu tanımlamak daha güçlü kütüphaneleri gerektirmektedir. Bu ihtiyaçtan yola çıkılarak geliştirilen NBK, eşleme ve nesne tanımlama yapmak için yüksek seviyeli özellikler içeren algoritmaları içermektedir (Rusu and Cousins 2011).

Kamera veya kameralar yardımıyla elde edilmiş gerçek nesnelerin nokta bulutları ile nesne tanıma ve bulma işlemleri iki boyutlu bir görüntü yapısı ile içeriğini yeniden düzenleyip, yeniden yapılandırmak son derece zor olmaktadır. Bu nedenle gerçek dünya nesnelerinin, 360 derecelik hareketli olarak tarama üretme kapasitesine sahip bir standart 3B kamera ile oluşturulacak olan sanal alanlar içerisinde, nesnenin 3B modeline ait nokta bulutları üretilerek bunlar üzerinden çıkarımlar elde edilmektedir (Droeschel et al. 2011).

Nesnelerin 3B modellerinin oluşturulup bunlar üzerinden çıkarımlar üretilmesinde nokta bulutu yapısının tercih edilme sebeplerinden en önemlisi 3B model verilerinin elde etme yöntemlerinde yaşanan sıkıntılardır. Genel olarak görüntüler üzerinde modelleme yapılırken çoğu zaman 2B görüntülerin yeterli olmadığı durumlarda 3B görüntüler devreye girmektedir (Wang et al. 2015). Kullanılacak görüntülerin belirli bir cepheden veya bakış açısından yakalanması gerektiği de bir başka problemdir. Görüntülerin işleme alınmadan önce gri seviye imgeler haline getirilmeleri gibi birkaç ön işleme tabi tutulması yine bir sıkıntılı süreçtir. 3B

görüntülerin üretilmesi ile nesne modelleme ve özellik çıkarımlarında sonuca olumsuz etki edecek etkenler en aza indirilse de düzlemsel uzayda nesneye ait verilerde kaymaların oluşması hizalama dönüştürme ve ilişkilendirme aşamalarında meydana gelebilecek sorunlara engel olunamamaktadır (Alam et al. 2010). 3B modellerin oluşturulması için tek görüntüyle yeniden oluşturma, yapılandırılmış ışık teknolojileri, fotometrik stereo ve çoklu görünüm gibi farklı yöntemler kullanılmaktadır (Vouzounaras et al. 2010, Zheng et al. 2011, Huang and Chen 2011). Tek görüntü ile oluşturma kontrolsüz bir ortamda yakalanan 2B görüntülerden faydalanılarak, 360 derecelik tam bir nesne görünümünde olmayan kesintili 3B verilerin oluşturulmasını kapsamaktadır. Yapısal ışık, fotometrik stereo ve çoklu görünüm stereo yöntemleri nesne modellemelerinde ortamdaki gölgelenmelere, parlaklıklara ve düzensiz aydınlatmalara karşı oldukça hassastırlar (Young et al. 2009). Ayrıca oluşturulan görüntüler yüzey normallerinden oluşmaktadır. Nesnelerin yüzey normallerinden mesh yapısının üretilmesi gerektiğinden bir dönüşüm işlemi sağlanmalıdır (Kim et al. 2009). Bu işlem ise söz konusu işlem zamanını artırmakta ve ayrıca ek hatalar oluşturmaktadır. Bu sebeple daha keskin ve gerçeğe yakın 3B modeller üretilmesi, bunlara ait tanıma ve bulma işlemlerinde kullanıma faydalı nokta bulutu oluşturma yöntemi kullanılması amaca uygun düşmektedir. Böylece bir sahne içerisindeki nesne modellerinin üretilmesi, tanınması ve bulunması işleminin otomatik olarak gerçekleştirilmesi insan bilgisayar etkileşimi alanında önemli bir alt konu olmakla birlikte geliştirilmeye açık olan bu alana bir öneri niteliğindedir (Lin et al. 2004).

## **2. Materyal ve Metot**

Bu çalışmada, gerçek nesnelerin, tanınması, belirlenmesi sınıflandırılması ve 3B modellerinin oluşturulması sürecinde kullanılan yöntem ve algoritmaların incelenip sağlam çalışan bir algoritma geliştirilerek bu algoritmaya bağlı bir uygulama geliştirilmesi önerilmektedir. Öncelikle nesnelere ait 3B sayısal verilerin çıkarılması

amacıyla nesnelerin derinlik görüntülerinin yakalanması için Kinect tabanlı kamera sistemli yöntemden faydalanılmıştır (Essmaeel et al. 2012). Kinect tabanlı kullanılacak olan nokta bulutu yöntemi, nesnelerin 3B sayısal verilerini oluşturmada, diğer yöntemler arasındaki dış ortama olan hassasiyetin en aza indirilmiş, bu şartlar altında daha doğru sonuçlar ürettiği, zamansal, maliyet ve uygulanabilirliği açısından kolaylıklar sağladığı için tercih edilmesi öngörülmüştür.

Bu çalışmasının amacı tanıma ve sınıflandırma işlemlerini gerçekleştirmek amacıyla, kamera kullanarak nesnelere ait nokta bulutlarını elde etmek ve elde edilen nokta bulutlarının metrikleri üzerinden geliştirilecek algoritma ve yöntemler ile çıkarımlar sağlamaktır (Ozbay and Cinar 2013).

Nokta bulutu üretme aşaması için Kinect kameranın sunduğu imkânlardan kısaca bahsedecek olursak, Kinect algılayıcı kameralar, derinlik bilgisi, renk verileri ve iskelet takip verilerini içermektedir. Kinect SDK kullanıcı ara yüzü ile bu verilere erişebilen ve verileri işleyebilen uygulamalar geliştirilebilmektedir. Alınan veriler sayesinde hareket algılama, ses algılama, yüz tanıma ve iskelet takip gibi pek çok uygulamalar geliştirilmesine imkân sunan Kinect algılayıcı içerisinde çalışan dört adet donanım bileşeni mevcuttur:

- Renkli VGA Video Kamera
- Derinlik Algılayıcı
- Birden çok mikrofon
- Motorlu eğme özelliği

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 bir fotoğraf makinesi gibi çalışarak uygulamalarda renkli görüntü kullanabilmemize olanak sağlamaktadır (Wang et al. 2015).

Derinlik algılayıcı, aydınlatma koşullarından bağımsız olarak üç boyutlu alanı görmek için bir kızılötesi yansıtıcısı ve tek renkli CMOS

(Complimentary metal-oxide semiconductor) ile birlikte çalışır. RGB ve Kinect derinlik akışlarının birleştirilmesi sonucu her bir piksel için renk değerine ilave olarak, derinlik bilgileri içeren bir 640x480 çözünürlükte RGB-D görüntü (nokta bulutu) üretir. İdeal koşullarda, derinlik akışı 3mm çözünürlüğe sahiptir. RGB ve derinlik kameralarının ikisi de 30 Hz frekansla çalışır (Smisek et al. 2013). Kinect algılayıcı RGB ve IR kamera ve IR vericiden oluşmaktadır. Kinect algılayıcı kameranın belirli bir ham derinlik mesafesi vardır. Ölçüm sırasında metrik derinliği ham derinliğe dönüştürmemize olanak sağlamaktadır. IR kamera açısı ile her pikselin derinliği pozisyonunu, metrik derinlik ölçüleri sayesinde ham derinlik mesafesini hesaplamaktadır. Her bir nokta bulutu kümesi içerisinde çok sayıda 3-boyutlu sayısal veri barındıran veri topluluğudur. Bir nokta bulutu topluluğu 3B uzayda farklı açılardan ve ışık koşullarından görüntüleyebileceğiniz, hatta kendi yüzeylerine renkler ve dokular uygulayabileceğiniz noktalar kümesidir. Nokta bulutu yeniden yapılandırma için tam bir 3B model oluşturmayı sağlayan bir kütüphanedir (Rusu and Cousins 2011). NBK,  $n$ -boyutlu nokta bulutu ve 3B geometri işlemi için yapılmış bir BSD, (Brekel Software Development), lisanslı yazılım kitaplığıdır (Rusu and Cousins 2011). NBK, ROS (Robot Operating System) ile tamamen uyumlu olacak şekilde inşa edilebilir, robotik, bilgisayarlı görme ve 3B geometri işlemleri gibi geniş bir kullanım alanına sahiptir.

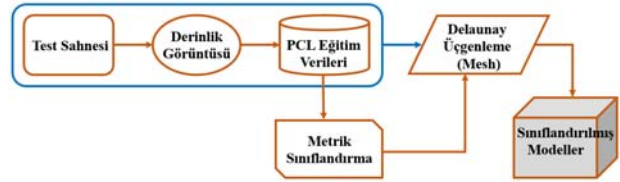
Nokta bulutu çok boyutlu veriyi içinde tutan bir veri yapısına sahiptir. Nokta bulutları genelde üç-boyutlu verileri göstermek için kullanılır ve bunlar belirli bir çözünürlükte nesne yüzeyine ait her bir noktanın x, y ve z koordinatlarıdır. Özellikle z koordinatı sadece derinliğe bağlı olarak, kameranın nesne yüzeyine olan uzaklığını kaydetmektedir. Bunun dışında kalan diğer koordinat uzunlukları Kinect algılayıcı kameranın karakteristik özelliği olarak kameradan 0,8 metre 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.

Ham veri üzerindeki ayırt etme ve bölgesel ayırma işlemleri için sınıflandırma algoritmaları kullanarak aşırı bölümlenmeden kaçınacak şekilde, anlamlı bir nokta bulutu kümeleme işlemi sağlanmaktadır. Kümeleme ve sınıflandırma yöntemleri içerisinde probleme hızlı ve kolayca çözüm sunacak bir yöntem tercih edilmektedir. Bu doğrultuda nokta bulutu alanları bağlı bulunduğu pürüzsüz sınırlar (yüzeyler) içerisindeki noktaların yerel yüzey normaleri ve K-en yakın ve sabit mesafe komşuluğu gibi bilinen sınıflandırma algoritmaları ile noktaların birbirine olan bağlantıları hesaplanmaktadır (Rabbani et al. 2006). Tanımlanacak olan nesne dışında farklı görüntüler kalmayacak şekilde, nesne haricinde yakalanan sahne içerisinde kalan arka planı yok etme ve gerekirse bazı filtreleme, hizalama ve dönüşüm işlemleri sonrasında metrik çıkarımlar ve sınıflandırma işlemleri gerçekleştirilmektedir (Tölgyessy and Hubinský 2011).

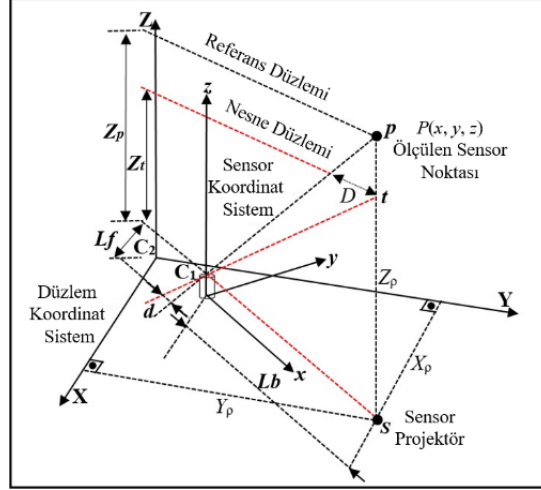
### 2.1. Derinlik Verilerinin Ölçülmesi ve Matematiksel Modeli

Bu çalışmada derinlik bilgileri yardımıyla gerçek nesnelere elde edilen nokta bulutu kümesi oluşturmada, içerisinde yer alan basit geometrik yüzeyleri çıkarmak için metrik sınıflandırma algoritması kullanılmıştır. Mean-Shift algoritmasını uygulamak için gereken normalizasyon işleminde Visual Studio .Net ortamında C# programı yürütülmüştür. Nokta bulutu verilerinin üretilmesinde açık kodlu bir kütüphane olan nokta bulutu kütüphanesinden (Pcl: Point cloud library) yararlanılmıştır. Nokta bulutu kütüphanesi açık kaynak kodludur. PCL ile filtreleme, nesne tahminleme, yüzey yeniden yapılandırma, model oluşturma, segmentation, registration vb. algoritmaları gerçekleştirilebilmektedir.

Şekil 1'deki akış diyagramında gösterildiği gibi, bu bölümde kısaca derinlik ölçüm prensibi için önerilen sahne adımlarına ve matematiksel model için 3B gerçek nesne verileri ölçümleme parametrelerine değinilmektedir (Rusu and Cousins 2011).



Şekil 1. Önerilen 3B model sınıflandırma akış diyagramı.



Şekil 2. Tarayıcı sensör ölçümlerine ait koordinat sistemi ve sensör ile zemin arasındaki derinlik eşitsizliğinin gösterimi.

Şekil 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  $Z_p$  mesafe uzaklıktaki nesnenin bulunduğunu farz edelim. Eğer bu nesne sensör kameradan belirli bir mesafe kadar uzaklaşır ya da yaklaşır, görüntü düzlemi üzerindeki mekisi  $x$  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 değiştirir. 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 gösterir,  $L_f$ , kızılotesi kameranın odak uzaklığı,  $L_b$ , taban uzunluğu,  $\delta x$ ,  $\delta y$ , lens bozulma katsayıları ve  $x_p$ ,  $y_p$ , temel ofset noktaları (Khoshelham, 2011).

$$\frac{D}{L_b} = \frac{Z_p - Z_t}{Z_p} \text{ ve } \frac{d}{L_f} = \frac{D}{Z_t} \quad (1)$$

Denklem 1'den  $D$  verisi,  $Z_t$  gibi farklı parametreler

ile ifade edilecek olursa,

$$Z_t = \frac{Z_p \cdot L_f \cdot L_b}{L_f \cdot L_b + Z_p \cdot d} \quad (2)$$

Denklem 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 planimetrik nesnelerin görüntü koordinatları, ölçek değerleri üzerinden hesaplanabilir:

$$X_t = -\frac{Z_t}{L_f}(x_t - x_p + \delta x), Y_t = -\frac{Z_t}{L_f}(y_t - y_p + \delta y) \quad (3)$$

Denklem 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 düzeltmeler. Burada biz temel görüntü koordinat sistemi ve derinlik koordinat sisteminin birbirine paralel olduğunu varsayarak gerçekleştirdik.

Bu hesaplamalar dikkate alınarak basit geometrik şekillere sahip olan birbirinden farklı 3 adet şekil barındıran bir sahne taranarak bir veri seti elde edilmiştir. Bu veri setinde bulunan farklı geometrik yapıya sahip üç adet nesne sahne içerisine yerleştirilmiş ve nokta bulutları üretilmiştir. Şekil 3'de nesneye, onun derinlik bilgisine ve ham nokta bulutu verilerine ait görüntüler gösterilmiştir.



**Şekil 3. Sol:** Taranan sahne içerisindeki 3 farklı planimetrik nesne, **Orta:** Nesnelerin yansıtılan kızılötesi derinlik görüntüsü, **Sağ:** Nesnelere ait elde edilen ham veriler (Pcl).

### 3. Metrik Sınıflandırma

Nokta bulutu içinden basit geometrik yüzeylerin çıkarılmasında, birden fazla planimetrik nesne içeren her bir sahne için metrik sınıflandırma algoritması derlenmiştir. Farklı geometrik şekle

sahip birbirinden belirgin biçimde ayrılan nesnelere için bir sınıflandırma işlemi sağlanmıştır. Bunun için yukarıda ölçümleme işlemleri dikkate alınarak elde edilen nesnelerin, nokta bulutu veri seti üzerinde çalışılmıştır.

Metrik sınıflandırma algoritması kullanılarak nokta bulutu içinde yer alan noktaların her birinin hangi nesneye ait olduğunun hesaplanması gerekmektedir. Bunun için de bu ardışıl noktaların birbirlerine komşu olan noktaların bilinmesi gerekmektedir. Komşu noktaların belirlenmesi için  $k$ -NN ( $k$ -Nearest Neighbor) algoritması kullanılmaktadır. En yakın komşuluk ilkesine göre işlem yapılacak noktaya en yakın noktaların belirlenmesi için, verilen nokta sayısına göre komşu noktaların birbirlerine olan Öklid mesafeleri değerlendirilmektedir.

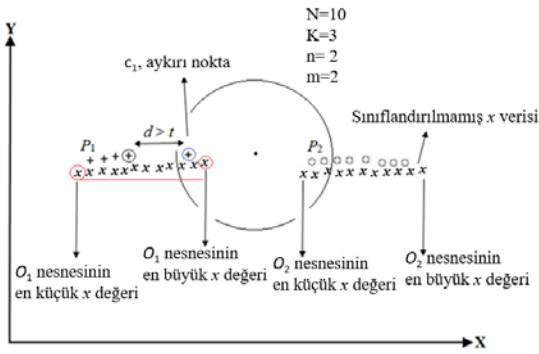
Nesnelerin nokta bulutlarına ait parametreler her bir nokta için  $P(x, y, z)$  olarak belirlenmiştir.  $Z$  boyutu tüm nesnelere sabit ve eşit mesafede ve onların nokta bulutlarına ait parametrelerinde ortak olduğundan dolayı devre dışı bırakılmıştır. Nokta parametreleri belirlendikten sonra, düzleme belirlenen bir eşik değer mesafesi ( $t$ ) içinde kalan noktalar düzleme atanır. Bütün bir nokta bulutu içerisinde her bir nesneye ait ilk referans verisi eşik değerinin üzerindeki ilk aykırı noktanın koordinatları kabul edilir. Noktaların verilen eşik değer içinde kalıp kalmadığını belirlemek için noktalar  $P_i(x_i, y_i)$  ile  $P_{i+1}(x_{i+1}, y_{i+1})$  arasındaki Öklid mesafesi hesaplanmaktadır. Bu sınıflandırmada belirlemesi gereken en önemli değerlerden birisi komşu noktaların birbirlerinden en fazla ne kadar uzakta olması gerektiğini gösteren eşik değeri mesafesinin ( $t$ ) belirlenmesidir. Bu makalede  $t$  eşik değeri her bir veri seti için deneysel olarak belirlenmiştir. Eşik değer mesafesi ( $t$ ), yaklaşık 1,5 metre mesafeden nesneye kalibre edilmiş bir kamera için 6 mm olarak belirlenmiştir.

$$Z_t^{-1} = \left(\frac{m}{L_f \cdot L_b}\right) d^{-1} + \left(Z_p^{-1} + \frac{n}{L_f \cdot L_b}\right) \quad (4)$$

Yukarıdaki denklem 4 veri seti içerisindeki ardışıl her bir noktanın bir sonraki nokta ile yakınlığını ölçen Öklid mesafe hesaplama formülasyonunu

göstermektedir. Metrik sınıflandırma algoritması ile birlikte en yakın komşuluk sınıflandırma kullanıldığında, onun çok basit ve sezgisel bir fikir taşıması ve uygulanmasının kolay olması etkindir. Metrik sınıflandırma algoritmasının sözde kodunu açıklayacak olursak:

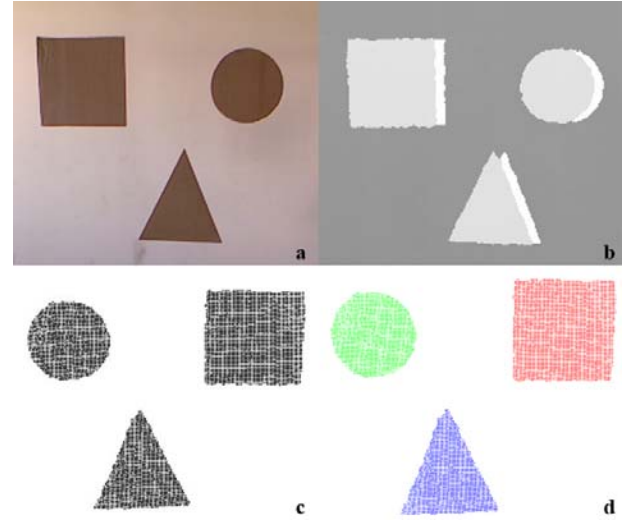
- Başla
- $P_1(x_1, y_1)$  veri setinin ilk noktasını ilk nesne  $O_1$  için referans noktası olarak ata ( $P_1$ ).
- Giriş:  $D = \{(P_b, P_{b+1}), \dots, (P_{N-1}, P_N)\}$
- $P = (P_1, \dots, P_N)$  sınıflandırılacak yeni örnekler
- Hesapla  $d(P_b, P_{b+1})$  işaretli her bir örnek için ( $P_b, P_{b+1}$ )
- Sırala  $d(P_b, P_{b+1})$  küçükten büyüğe doğru, ( $i = 1, \dots, N$ )
- if  $d(P_b, P_{b+1}) \geq t$  şart sağlanması durumunda karar ver.
  - Mevcut nesneye dahil edilecek bir nokta ya da,
  - Yeni bir nesne için oluşturulacak ilk referans noktası,
- (smallest  $x O_1 < c_1 < biggest x O_1$ ) && ( $d(c_1, x) (P_b, P_{b+1}) \leq t$ ) Mevcut nesneye dahil edilir  $O_1$ .
- else
- Yeni bir  $O_2$  nesnesinin ilk referans noktası kabul edilir.
- Bitir



**Şekil 4.** Noktalar arası metriksel yakınlık hesaplama, m sınıf sayısını, n belirleyici değişkenlerin sayısını ve N etiketli vaka sayısını ifade etmektedir.

Yukarıda açıklanan metrik sınıflandırma sürecindeki  $-d(P_i, P_{i+1}) \geq t$  sözde kod adımı ilgili noktanın hangi durumlarda mevcut nesneye dâhil edileceği hangi durumlarda ise yeni bir nesnenin ilk referans noktası olarak belirleneceği Şekil 4'te şekillendirilmiştir. Şekilde görüldüğü gibi  $c_1$  aykırı nokta, yani ardışık düzendeki komşusu ile aralarındaki Öklid uzaklığı 6mm'yi aşan nokta, yeni bir nesnenin ilk referans noktası olmaya aday bir noktadır.  $d > t$  yani  $d > 6\text{mm}$  şartı sağlandığından bundan sonra metrik sınıflandırma yaklaşımı değerlendirilecektir.  $P_1$  nesnesine ait olup olmadığı değerlendirilecek olan aykırı  $c_1$  noktası kendisine komşuluğu olan ( $d(c_1, x) \leq t$ ) aday noktalar ile kontrol edilmektedir. Bu aday noktalar arasından en az bir tanesinin kendisine komşu olması ve bu  $c_1$

noktasının  $x$  boyutunun en küçük ve en büyük aday koordinatları arasında bulunması yeterlidir. Bazı istisnai durumlarda  $X$  doğrultusu ile birlikte  $Y$  doğrultusu da benzer şekilde alınarak aynı mantıksal işleme tabi tutulmuştur. Tüm veri seti içerisindeki veriler sınıflandırılıp nesne sayısına bağlı olarak renklendirilmiştir. RGB renk kodları belirgin bir biçimde nesne nokta bulutlarının ayırt edilebilmesi açısından sınıflandırılan verilere değer olarak atanmıştır.



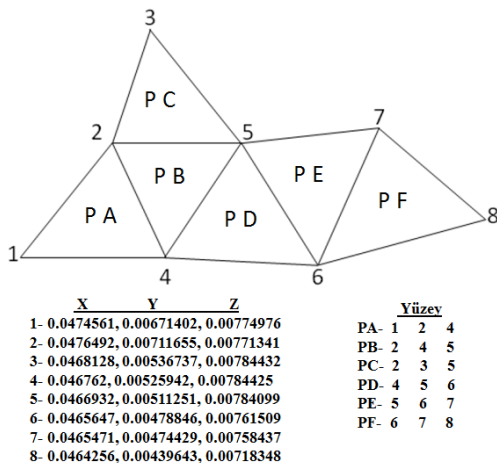
**Şekil 5.**  $t = 0.006\text{m}$  için sınıflandırılan planimetrik nesne düzlemleri. a) Üç farklı nesnenin RGB görüntüsü, b) Derinlik görüntüsü, c) Nesnelerin ham derinlik bilgileri - Pcl, d) Üç farklı nesneye sahip nokta bulutunun metrik sınıflandırılmış hali.

Sensör kameralar ile tarama işlemi gerçekleştirilirken gerek taranan objenin yüzey pürüzlülüğü gerekse, lazer ışınının yüzeyden saçınım özelliklerinden dolayı nokta bulutları obje yüzeyinde bir kalınlık oluşturmaktadır. Yüzey üzerindeki noktaların, çıkarılan model üzerine atanması için  $t$  eşik değeri mesafesinin doğru bir şekilde tespit edilmesi gerekmektedir. Bu metrik classification algoritması ile nokta bulutu içerisindeki nesne yüzeyi çıkarımında belirlenmesi gereken en önemli parametre  $t$  eşik değeri mesafesidir. Bu kapsamda  $t$  değerine farklı değerler verilerek metrik sınıflandırma algoritması veri setine uygulanmıştır.  $t$  eşik değeri mesafeleri  $t = 0.001\text{m}$ ,  $t = 0.01\text{m}$ ,  $t = 0,005\text{m}$  ve  $t = 0.007\text{m}$  seçilerek veri setine uygulanmıştır. Elde edilen denemeler sonucunda en iyi sonucun  $t = 0.006\text{m}$

değerinde alındığı gözlemlenmiştir. Böylece planimetrik nesnelere ait nokta bulutlarının test veri seti üzerine metrik classification algoritması nihai  $t$  değeri ile sınıflandırma işlemi uygulanmıştır. Şekil 5'te görülen üç farklı nesneye ait veri seti içinde toplam 8495 adet nokta sayısı mevcuttur. Bu üç nesneli veri seti toplamda  $0.76 \times 10^3$  ms gibi kısa bir sürede otomatik olarak çıkarılmıştır. Şekilde aynı veri seti içerisindeki farklı nesnelere ait düzlemler sınıflandırılarak farklı renklerle renklendirilmiştir. Her bir düzleme ait nokta sayısı ise renklere bağlı olarak gösterilmiştir. 8495 adet nokta bulutu veri setinin sınıflandırılması sonucunda 4024 adet veri kırmızı 2277 adet veri yeşil 2194 adet veri ise mavi renkli nokta bulutu ile renklendirilmiştir.

#### 4. Delaunay Üçgenleme

Çokgen kafes (Mesh), donanımla çizildiklerinde, 3B modelleri depolamak için oldukça yaygın kullanılan bir veri formatıdır (Erdal and Ahmet 2013). 3B modelleri oluşturan unsurlar genellikle köşeler, kenarlar, yüzeyler ve çokgenlerdir. Her köşenin kendi yeri vardır ve iki köşe bir kenar oluşturmaktadır. Üç kenar bir kapalı yüzey oluşturur ve bu nedenle de yüzler kapalı kanarlar dizisidir. Çokgenler bir kenarlar dizisinden meydana gelmektedir ve eğer sistem çok kenarlı yüzleri destekliyorsa çokgenler ve yüzler arasında bir fark yoktur. Genellikle oluşturma sistemleri çok kenarlı yüzleri desteklemezler ve bu da kullanıcıya çok kenarlı çokgenleri seçmekten başka seçenek bırakmamaktadır.



Şekil 6. İlişkili verilerin üçgenlenmesi.

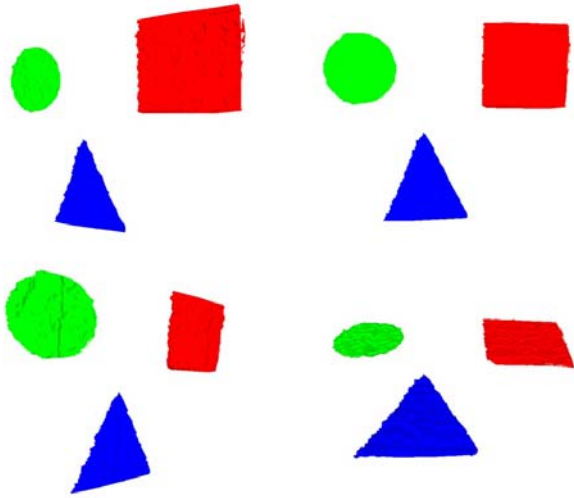
Çokgen bir kafeste, her çokgen üç köşeden oluşmaktadır ve her köşe çoklu çokgenlerin bir parçası olabilmektedir. Bir nokta birden fazla kenara ait köşeyi temsil edebilmektedir. Şekil 6'da üç köşeden oluşan çokgen kafesin her yüzünü görebilirsiniz. Örneğin, PA yüzü 1, 2 ve 4 köşesinden oluşmuştur. Aynı zamanda köşelerin çoğunun çoklu yüzlerin bir parçası olduğunu da görebiliriz. Örneğin; 5 köşesi PD, PB ve PC yüzlerinin bir parçasıdır (Bischoff et al. 2002). Şekilde görüldüğü gibi bir yüzeye ait köşe noktaları  $x$ ,  $y$ ,  $z$  koordinat noktaları ile birleştirilerek çokgenler şeklinde birbirleriyle ilişkilendirilmektedir. Nokta bulutu verilerinin birleştirilmesi aşamasında delaunay üçgenleme yöntemi kullanılmaktadır.

Delaunay üçgenleme yönteminden bahsedecek olursak, öncelikle temel prensip bir üçgenin çevrel çemberi içerisinde başka bir dayanak noktası bulunamamasıdır. Yani üçgeni oluşturacak üç köşeden daha fazla noktanın aynı çember üzerinde yer alması bu kuralın istisnasıdır. Algoritmada bu özel durum dikkate alınmalıdır. Bazı belirsizlik durumlarının ortadan kalkması için algoritmaya bir seçim ölçütü konulmaktadır. Çevrel çember ölçütü genellikle artan üçgenleme yöntemlerinde kullanılan ölçüttür. Oluşturulacak olan her bir üçgenin 3. noktası belirlenirken bu üç noktadan geçen çevrel çemberin içerisinde bir başka dayanak noktası olup olmadığı kontrol edilmektedir. Çember içerisinde bir başka noktanın olduğu belirlenirse, belirlenen bu nokta yeni oluşacak üçgenin 3. köşesi olmaya aday olmaktadır. Daha sonra ise bu aday üçgenin çevrel çemberi içerisinde bir başka nokta olup olmadığı kontrol edilmektedir. Bu işlem çevrel çember içerisinde nokta kalmayana kadar devam etmektedir ve üçgen oluşturulmaktadır (Chew, 1989).

Böylece her bir nokta bulutu verisine ait üçgenleme bilgileri veri seti içerisine eklenerek modelleme işlemi tamamlanmıştır. Nokta bulutu içerisindeki her bir veri hangi koordinat bilgisi ile eşleştirileceği dosyaya kaydedilmiştir.

Aynı veri seti içerisinde bulunan farklı geometrik

şekilli 3 ayrı nesne nokta bulutları çıkarıldıktan sonra sınıflandırılarak birbirinden ayrıştırılmıştır. Daha sonra sınıflandırılmış veriler üçgenleme yöntemi ile modellenmesi görselleştirilmiştir (Cinar and Ozbay 2014). Bu sonuçlar 64 bit Windows 10 işletim sistemi üzerinde, 8Gb ram ve Core i7 2.30Ghz işlemciye sahip olan kişisel bir bilgisayarda elde edilmiştir. Sonuçlar değerlendirildiğinde sınıflandırılan verilerin sayısı ham veri seti için de aynı olduğu göz önüne alındığında önerilen yöntemin model sınıflandırma için önemli bir avantaj sağladığı gösterilmiştir.



Şekil 7. Metrik sınıflandırma işlemi uygulanmış veri setine ait modellerin farklı açılardan gösterimi.

## 5. Tartışma ve Sonuç

Bu makalede, geometrik düzleme sahip (planimetrik) nesnelere Kinect sensör kamera ile elde edilen derinlik görüntülerine ait 3D nokta bulutları üzerinden bir sınıflandırma işlemi gerçekleştirilmiştir. Sınıflandırma işlemi gerçekleştirilirken kullanılan metrik sınıflandırma algoritması ile ham veri setleri değerlendirilmiştir. Genellikle sınıflandırma algoritmaları, parametrik olmayan yöntemlerdir. Bazı durumlarda sınıflandırılacak olan nesne ve içinde bulunduğu ortamın (arka plan) rengi, dolayısıyla renk olasılık dağılımları, çok benzer olabilmektedir. Bu gibi durumlarda sınıflandırma işlemleri için renk dağılımı bilgisini kullanmak iyi sonuç vermeyecektir. Bu problemin üstesinden gelmek için, sunulan yöntemde, nesnelere ait elde edilen nokta bulutları üzerinden sayısal işlemler yürütülmüştür. Bunun

için öncelikle bir sahnenin Kinect sensör kamera ile taranarak 3B sayısal verilere sahip nokta bulutları üretilmiştir. Ardından bu ham nokta bulutu verilerine metrik sınıflandırma adımı gerçekleştirilmiştir. Delaunay üçgenleme yöntemi ile sınıflandırılan nokta bulutu veri seti ilişkilendirilerek sınıflandırılmış modeller elde edilmiştir. Güvenilir bir sınıflandırma çalışması yapabilmek amacı ile doğrudan uygulanabilecek sayısal veriler üzerinden işlemler yapılmış ve metrik sınıflandırma işlemi, nokta bulutu üretme (Pcl) ve delaunay üçgenleme yöntemi ile birleştirilmiştir. Sınıflandırılacak olan nesnelere hızlı ve düzenli olarak yürütülmesi için bir metrik sınıflandırma algoritması kullanılmış ve verilere ait başlangıçtaki sayısal değerleri korunmuştur. Geliştirilen yöntemin performansı gerçek hayattan alınan görüntü dizileri üzerinde uygulamalar yapılarak uygulanmıştır.

## Kaynaklar

- Duarte, N., Octavian, P., and Jacob, S., 2014. KSGphysio-Kinect serious game for physiotherapy. *Electrical and Power Engineering (EPE), International Conference and Exposition on. IEEE*, 606-611.
- Guo, Y., et al. 2015. An Integrated Framework for 3-D Modeling, Object Detection, and Pose Estimation From Point-Clouds. *Instrumentation and Measurement, IEEE Transactions*, **64**, 683-693.
- Bischoff, B.S., et al. 2002. OpenMesh—a generic and efficient polygon mesh data structure. *In OpenSG Symposium*.
- Rusu, R.B., Cousins, S., 2011. 3d is here: Point cloud library (pcl). *Robotics and Automation (ICRA), IEEE International Conference*, 1-4.
- Droeschel, D., Stuckler, J., Behnke, S., 2011. Learning to interpret pointing gestures with a time-of-flight camera. *Proceedings of the 6th international conference on Human-robot interaction ACM*, 481-488.
- Wang, C., et al. 2015. Real-Time Depth Image Acquisition and Restoration for Image Based Rendering and Processing Systems. *Journal of Signal*



- Processing Systems*, **79**, 1-18.
- Alam, M., Goh, S. F., Dacharaju, S., 2010. Three-dimensional color pattern recognition using fringe-adjusted joint transform correlation with CIE Lab coordinates. *IEEE Transactions on Instrumentation and Measurement*, **8**, 2176-2184.
- Vouzounaras, G., et al. 2010. 3d reconstruction of indoor and outdoor building scenes from a single image. *Proceedings of the ACM workshop on Surreal media and virtual cloning*, 63-66.
- Zheng, H., Yuan, J., Gu, R., 2011. A novel method for 3D reconstruction on uncalibrated images. *Proceedings of the Third International Conference on Internet Multimedia Computing and Service ACM*, 138-141.
- Huang, Y.Y., Chen, M.Y., 2011. 3D object model recovery from 2D images utilizing corner detection. *System Science and Engineering (ICSSE), International Conference on. IEEE*, 76-81.
- Young, M.K., Theobalt, C., Diebel J., Kosecka J., Miscusik B., and Thrun S., 2009. Multi-view image and tof sensor fusion for dense 3d reconstruction. *In IEEE 12th International Conference on Computer Vision Workshops*, 1542–1549.
- Kim, Y.M., et al. 2009. Multi-view image and tof sensor fusion for dense 3d reconstruction. *ICCV Workshops*, 1542-1549.
- Lin, H.W., Tai, C.L., Wang, G.J., 2004. A mesh reconstruction algorithm driven by an intrinsic property of a point cloud. *Computer-Aided Design*, **36**, 1-9.
- Essmaeel, K., et al. 2012. Temporal denoising of kinect depth data. *Signal Image Technology and Internet Based Systems (SITIS), Eighth International Conference on. IEEE*, 47-52.
- Ozbay, E., Cinar, A., 2013. 3D Reconstruction Technique with Kinect and Point Cloud Computing. *Global Journal on Technology*, 1748-1754.
- Smisek, J., Jancosek, M., Pajdla, T., 2013. 3D with Kinect. *Consumer Depth Cameras for Computer Vision. Springer London*, 3-25.
- Rabbani, T., van den Heuvel, F., Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **36**, 248-253.
- Tölgýessy, M., Hubinský, P., 2011. The kinect sensor in robotics education. *Proceedings of 2nd International Conference on Robotics in Education*, 143-146.
- Khoshelham, K., 2011. Accuracy analysis of kinect depth data. *Remote Sensing and Spatial Information Sciences, ISPRS workshop laser scanning*, **38**, 5-12.
- Erdal, O., Ahmet, C., 2013. A novel approach to smoothing on 3d structured adaptive mesh of the kinect-based models. *2th International Conference on Advanced Information Technologies and Applications ICAITA*, 13-22.
- Chew, L.P., 1989. Constrained delaunay triangulations. *Algorithmica*, **4**, 97-108.
- Cinar, A., Ozbay, E., 2014. Üç Boyutlu Derinlik Bilgileri Yardımıyla Modelleme Yöntemlerinin İncelenmesi. *7. Mühendislik ve Teknoloji Sempozyumu, Çankaya Üniversitesi*, 101-107.