


## Kinect Uygulamaları için Veri Transfer Platformu Tasarımı

\*<sup>1</sup> Erdal Erdal, <sup>2</sup> Atilla Ergüzen

<sup>1</sup> Mühendislik Fakültesi, Bilgisayar Mühendisliği Kırıkkale Üniversitesi, Türkiye, [erdalerdal@kku.edu.tr](mailto:erdalerdal@kku.edu.tr), 

<sup>2</sup> Mühendislik Fakültesi, Bilgisayar Mühendisliği Kırıkkale Üniversitesi, Türkiye, [atilla@kku.edu.tr](mailto:atilla@kku.edu.tr), 

Araştırma Makalesi

Geliş Tarihi: 06.08.2018

Kabul Tarihi: 15.03.2019

### Öz

Son yıllarda yazılım, donanım ve algoritma konularında büyük gelişmeler meydana gelmiştir. Teknolojide yaşanan bu gelişmeler sensör teknolojilerini de etkilemiştir. Başlangıçta bir oyun cihazı olarak piyasaya sürülen Kinect sensör gerek araştırmacılar gerek geliştiriciler tarafından büyük ilgiyle karşılanmıştır. Kinect sensör literatürde farklı alanlarda farklı amaçlar için kullanılmıştır. Kinect sensörden alınan tüm veriler Microsoft tarafından geliştirilen Yazılım Geliştirme Kiti (YGK) ile geliştiricilere iletilmektedir. Kinect sensörü sahne karmaşıklığına göre değişmek üzere normal durumlarda her saniyede 240 bin ile 270 bin nokta verisi üretmektedir. Bu çalışmanın amacı Kinect uygulamaları için veri transfer platformu tasarlanmasıdır. Geliştirilen platform istemci sunucu mimarisi üzerinde çalışmaktadır. Çevrimiçi ve çevrimdışı haberleşme durumlara uygun farklı senaryolar barındıran platform, aynı zamanda bir dizi filtreleme ve şifreleme algoritmalarını da sunmaktadır. Platformda 2D/3D görüntü ve nokta bulutu işleme için büyük ölçekli, açık kaynaklı bir proje olan Nokta Bulut Kütüphanesi (NBK) kullanılmıştır. İsteğe bağlı olarak VoxelGrid (VG) Filtre, Outlier Filtre, Histogram Tabanlı Koşullu Filtre, Octree-tabanlı Sıkıştırma ve PGP Şifreleme yöntemlerini de barındırmaktadır. Ayrıca Kinect uygulamalarına özel bir veri yapısı da geliştirilmiştir. Çevrimiçi haberleşme için WebRTC ara katman yazılımı kullanılmıştır. Tüm bu aşamalar sonucunda gereksiz veri noktaları temizlenmiş, sıkıştırılmış, güvenli hale getirilmiş ve geliştirilen veri yapısına uygun veri paketleri elde edilmiştir. Filtrelemeler sonucunda % 19.96 sıkıştırma oranı elde edilmiştir. İsteğe bağlı tasarım sayesinde uygulama veya istemci bazlı filtreleme sağlanmıştır. Filtrelemeler sonrasında uygulanan dosya sıkıştırma yaklaşımı ile % 10.38 oranında dosya sıkıştırma sonucu da elde edilmiştir. Sunulan platform araştırmacılar ve geliştiriciler tarafından kullanılan Kinect uygulamalarında performans sağlayacaktır.

**Anahtar Kelimeler:** Kinect, Veri Transferi, Servis Tabanlı Platform, Veri Yapısı.

## Data Transfer Platform Design for Kinect Platform for Kinect Applications

\*<sup>1</sup> Erdal Erdal, <sup>2</sup> Atilla Ergüzen

<sup>1</sup> Faculty of Engineering, Computer Engineering, University of Kırıkkale, Turkey, [erdalerdal@kku.edu.tr](mailto:erdalerdal@kku.edu.tr)

<sup>2</sup> Faculty of Engineering, Computer Engineering, University of Kırıkkale, Turkey, [atilla@kku.edu.tr](mailto:atilla@kku.edu.tr)

### Abstract

In recent years, software, hardware, and algorithms have come to fruition. These developments in technology have also affected sensor technologies. The Kinect sensor, initially marketed as a gaming device, has been met with great interest by both researchers and developers. The Kinect sensor has been used in different areas in the literature for different purposes. All data from the Kinect sensor is transmitted to developers through the Software Development Kit (SDK) developed by Microsoft. The Kinect sensor produces between 240 and 270 thousand points per second in normal conditions, depending on scene complexity. The purpose of this work is to design a data transfer platform for Kinect applications. The developed platform works on client server architecture. The platform, which has different scenarios for online and offline communication situations, also offers several filtering and encryption algorithms. The platform uses the Point Cloud Library (PCL), a large-scale open source project for 2D / 3D image and point cloud processing. VoxelGrid (VG) Filter, Outlier Filter, Histogram Based Conditional Filter, Octree-based Compression and PGP Encryption methods are also available on request. In addition, a special data structure has been developed for Kinect applications. WebRTC middleware software is used for online communication. Thanks to all these steps, unnecessary data points have been cleaned, compressed, secured and data packages suitable for the developed data structure have been obtained. Because of the filtering, a compression ratio of 19.96% has been obtained. Application-based or client-based filtering is provided through custom design. A file compression result of 10.38% has been obtained with the file compression

approach applied after filtering. The presented platform will provide performance for Kinect applications used by researchers and developers.

**Keywords:** Kinect, Data Transfer, Service Based Platform, Data Structure.

## 1. GİRİŞ

Son yıllarda bilgisayar bilimlerinde, donanımlarda, algoritmalarda ve yazılımlarda büyük gelişmeler yaşanmıştır. Bu gelişmeler neticesinde hayatımızda kullandığımız cihaz ve sensör sayısındaki artış ile dijital ve elektronik içerik boyutları artış göstermiştir. Bu sayede geliştirilen teknolojiler bilgisayarlarla etkileşimlerinde yeni yöntemler geliştirdikçe, makineleri kullanma hedeflerimiz de önemli ölçüde değişmiştir. Teknolojide yaşanan bu gelişmeler ve kullanım değişimleri sensör teknolojilerini de etkilemiş ve Microsoft Kinect kameraları [1] piyasaya sürülmüştür. Temelde bir oyun cihazı olarak Microsoft tarafından geliştirilen Kinect sensörü piyasadaki ilk görünümünden bu yana, çok düşük maliyetiyle birlikte bir ölçüm aracı olarak kullanıldığında ve yüksek potansiyeli ile araştırmacıların ilgisini çekmeyi başarmıştır. Kinect sensör kullanılarak çeşitli alanlarda uygulamalar incelenmiş ve geliştirilmiştir.

Örneğin tıbbi alanda Kinect sensörünün kullanılması, fiziksel rehabilitasyon testlerini gerçekleştirmek ve kişinin yürüme parametrelerinin incelenmesi için kullanılmıştır [2-4].

İşyerindeki güvenlik konularıyla ilgili olarak, Kinect sensörünü işyeri ergonomisi değerlendirmelerinin gerçekleştirilmesi için taşınabilir bir 3-D hareket yakalama sistemi olarak kullanma olasılığı incelenmiştir ve görüntü işleme teknikleri ve derinlik bilgileri kullanarak inşaat işçilerini takip etmek için kullanılmıştır [5, 6].

Mimari alanda, sanal ortamda uygulama için bir nesnenin üç boyutlu bir modelini hızla oluşturmak için Microsoft Kinect sensörünün kullanımı değerlendirilmiştir [7].

3D tarama alanında, Kinect sensörünün görevleri yerine getirmek için nispeten ucuz ve erişilebilir bir alternatif olduğu kanıtlanmıştır. Bunun başlıca nedeni, kolayca işlenebildiği ve birçok uygulama için kabul edilebilir bir doğrulukla gerçek zamanlı olarak 3D gösterimleri oluşturabilmesidir [8].

Bu alandaki uygulamalar, arkeolojik alanların 3 boyutlu temsillerini yeniden inşa etmek için bir prototipten [9], artırılmış gerçeklik simülasyonları ve fizik tabanlı etkileşimlere izin veren bir dizi Kinect sensöründen oluşturulan iç mekan sahnelerine kadar değişmektedir [10].

Robotik alanında da çok fazla sayıda Kinect çalışması yapılmıştır. İnsan operatörünün hareketlerini temas etmeksizin iki robot manipülatörüne iletmesine izin veren Kinect bazlı bir yöntem sunulmuştur [11]. Bir İnsan-Makine Ara yüzü olarak bir Kinect sensörünü kullanarak bir

endüstriyel robotun kontrol edilmesi ve izlenmesi, çevrimiçi programlamanın bir konsepti ve uygulanmasına dair bir çalışma sunulmuştur [12].

Araç kontrol sistemlerinin geliştirilmesi ve yönetilmesi alanlarında da Kinect sensörüne tabanlı çalışmalar ve araştırmalar yapılmıştır. Bu alanda yapılan bir çalışmada, Microsoft Kinect sensörü, otonom bir taşıt aracına eklenen bir bileşen haline getirilmiş ve iç mekânlarda gezinmeyi optimize etmek ve bir robotun bunu başarılı bir şekilde yapmasına yardımcı olmak için atalet ölçüm birimi gibi diğer sensörlere entegre edilmiştir [13]. Bir diğer çalışmada, hızını ve tutumunu doğrudan kontrol etmek için bir mobil robot üzerinde bir Kinect sensörü ile insan hareketinin bir izleme kontrol sistemi geliştirilmiştir [14]. Mobil bir robotun renkli işaretleri tanıması amacıyla sağlam, adaptif ve öğrenme yaklaşımı için Kinect-tabanlı rehberlik ve konumlandırmasını geliştiren bir çalışma da yapılmıştır [15].

Bir diğer araştırmada Kinect sensörü, dinamik ortamlarda robot navigasyonu için mekânsal gösterimlerde Öklid mesafe haritaları ve genelleştirilmiş Voronoi diyagramlarının güncelleme stratejilerinin test edildiği çalışmalarda kullanılmıştır [16].

Ayrıca Kinect sensörü için önerilen farklı kullanımların sunulduğu bir çalışma da yapılmıştır [17].

Kinect sensörünün metrik özelliklerini tanımlamayı amaçlayan ve derinlik ölçümlerindeki doğruluğa özellikle değinen çalışmalar da yapılmıştır [18-22].

Yapılan literatür araştırmasında da görüldüğü üzere Kinect, bilgisayarlarla çalışma becerilerini zenginleştirmek isteyenler için sunduğu çok çeşitli seçeneklerle iyi bir alternatiftir. Şekil 1'de görüldüğü üzere Kinect kamera üzerinde RGB kamera, çoklu dizi mikrofonlar ve yüz ve ses tanıma yetenekleri ile birlikte tam vücut 3D hareket yakalama yeteneğine sahip derinlik sensörü bulundurmaktadır [23].



Şekil 1. Kinect Kamera

Halen piyasada kullanıcıların ve geliştiricilerin kullanımına sunulmuş durumda olan iki farklı Kinect kamera versiyonu bulunmaktadır. İki kamera arasındaki farklar Tablo 1’de gösterilmektedir [24]. Hem literatürde yapılan araştırmalarda hem de sektörde geliştirilen uygulamalarda Microsoft tarafından geliştirilen ilk Kinect kamera yoğun olarak tercih edilmektedir. Bu tercihin sebepleri; bu kamera ile kullanılan YGK’ya ait daha fazla doküman olmasıyla beraber aralarındaki büyük fiyat farkıdır.

**Tablo 1.** Kinect Versiyon Özellikleri

Özellik	Windows için Kinect 1	Windows için Kinect 2
Renk Kamera	640 x 480	1920 x 1080
Saniyede Yakalanan Kare	30 FPS	30 FPS
Derinlik Kamera	320 x 240	512 x 424
Maksimum Derinlik Mesafesi	~4.5 M	~4.5 M
Minimum Derinlik Mesafesi	Yakın modunda 40 cm	50 cm
Yatay Görüş Alanı	57 derece	70 derece
Dikey Görüş Alanı	43 derece	60 derece
Eğim Motoru	Var	Yok
Tanımlı İskelet Eklemleri	20 Eklem	26 Eklem
İzlenen Tam İskelet	2	6
Desteklene USB Standardı	2.0	3.0
Desteklenen İşletim Sistemi	En az Windows 7, 8	En az Windows 8

Kinect kameradan alınan tüm veriler yine Microsoft tarafından geliştirilen YGK ile USB portu üzerinden geliştiricilere iletilmektedir. Şekil 2’de belirlenen iskelet ve sağ el eklem bilgilerine ait Kinect verisini alan kodlar görülmektedir.

```
//İskeletler
foreach (Skeleton skeleton in skeletons)
{
    // Sağ El
    Joint rightHand = skeleton.Joints[JointType.HandRight];

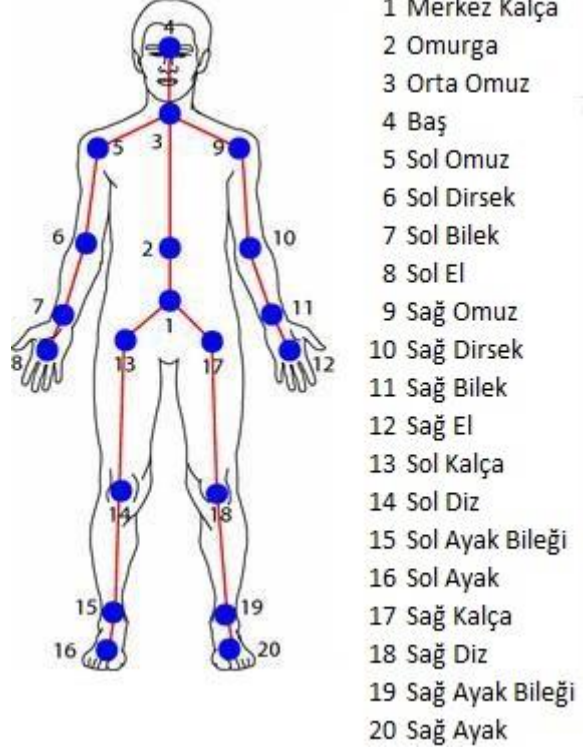
    // sağ elin bireysel noktalarını al
    double rightX = rightHand.Position.X;
    double rightY = rightHand.Position.Y;
    double rightZ = rightHand.Position.Z;
}
```

**Şekil 2.**Kinect YGK C# Kodu

Kinect 1 sensöründe, tanımlı iskeletlere ait 20 farklı eklem bilgisi sunulmaktadır. Sensörün sunduğu eklem noktaları Şekil 3’de görülmektedir.

Tüm bu avantajları, özellikleri ve düşük maliyeti ile Kinect sensörlerinin uzun süre araştırmacılar tarafından tercih edileceği öngörülmektedir. Windows için Kinect 1 sensörü düşünüldüğünde, iki iskelete ait 20 eklem, her ekleme ait X,

Y ve Z koordinat bilgileri saniyede 30 kare halinde alınmaktadır. Kinect sensörü sahne karmaşıklığına göre değişmek üzere normal durumlarda her saniyede 240 bin ile 270 bin nokta verisi üretmektedir. Bu nedenle dinamik nokta bulutları, büyük miktarda veri olan saniyede yaklaşık 7,5 milyon nokta içerir.



**Şekil 3.** Tanımlı İskelet Eklemleri

Bu çalışmanın amacı Kinect uygulamaları için veri transfer platformu tasarlanmasıdır. Geliştirilen platform istemci sunucu mimarisi üzerinde çalışmaktadır. Çevrimiçi ve çevrimdışı durumlara farklı senaryolar barındıran platform aynı zamanda bir dizi filtreleme ve şifreleme algoritmalarını da sunmaktadır. Platformda 2D/3D görüntü ve nokta bulutu işleme için büyük ölçekli, açık kaynaklı bir proje olan Nokta Bulut Kütüphanesinin (NBK) kullanımı tercih edilmiştir [25]. Bu kütüphanenin barındırdığı filtreler ve algoritmalar Kinect sensörden elde edilen noktalara uygulanmıştır. Hassas verilere sahip uygulamalar için opsiyonel PGP şifreleme yöntemi kullanılmıştır. Sensörden alınan ve filtrelerden geçirilen noktalar, geliştirilen veri yapısına uygun olarak paketlenmiştir. Çevrimiçi uygulamalarda anlık veri transferi aşamasında WebRTC kullanılmıştır. WebRTC, eşler arası ses, video ve veri paylaşımını gerçek zamanlı olarak sağlayan ve çeşitli platformlar ve farklı ağ yapılandırmaları arasında çoklu ortam iletişimindeki bazı sorunları hafifleten bir ara katman yazılımıdır [26-28]. Çevrimdışı uygulamalarda paketler klasik yöntemlerle sunucuya iletilmiştir. Yapılan çalışma sayesinde Kinect sensöründen elde edilen nokta bulutu isteğe bağlı temizlenmiş ve şifrelenmiş, literatüre Kinect uygulamalarına özel yeni bir veri yapısı kazandırılmış ve bu paketin transferi aşamasında WebRTC kullanılmıştır.

Makalenin kalan bölümleri aşağıdaki şekilde düzenlenmiştir. 2. bölümünde yöntem ve uygulama sunulmaktadır. Makalenin 3. bölümünde platformun kullanımı sonucu elde edilen bulgular paylaşılmaktadır. Son olarak bölüm 5'te çalışmanın sonuçlarından ve gelecek çalışmalardan bahsedilmektedir.

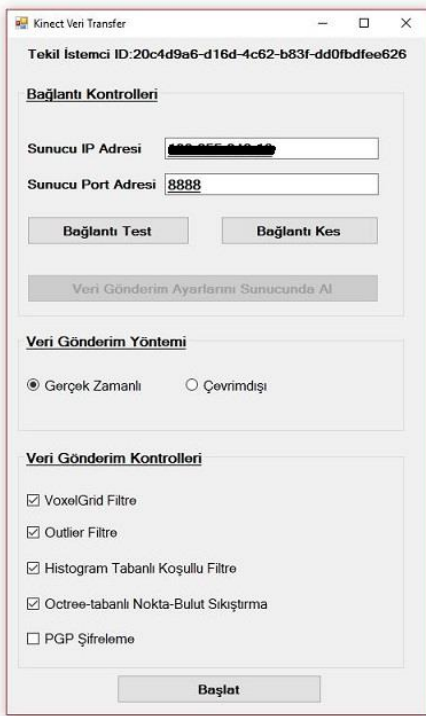
## 2. YÖNTEM VE UYGULAMA

Çalışmanın bu bölümünde platformda uygulanan yöntem ve uygulama sunulmuştur. Bu bölümün daha detaylı ve anlaşılabilir olması için dört alt bölüm halinde sunulmaktadır. Bu bölümler aşağıda listelenmiştir.

- 1) Geliştirilen platformun sistem mimarisi,
- 2) İsteğe bağlı sistem özellikleri,
- 3) Kinect uygulamalarına özel geliştirilen veri yapısı,
- 4) WebRTC ara katman yazılımından bahsedilmektedir.

### 2.1. Sistem Mimarisi

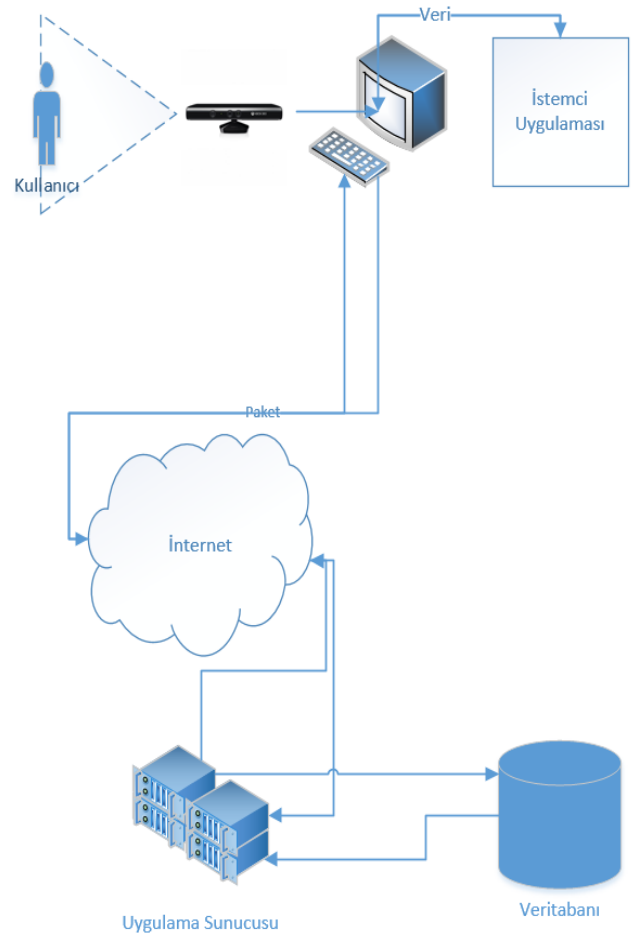
Kinect sensörler istemci bilgisayara USB portundan bağlı bulunmaktadır. Sensörden alınan veriler bu port üzerinden istemci bilgisayarına aktarılmaktadır. Aktarılan veri Şekil 4'te görülen arka planda çalışan platforma özel geliştirilen kontrol menüsü ile uygulama sunucusuna aktarılmaktadır.



Şekil 4. Platform Kontrol Menüsü

Bu menü üzerinden istemci sunucu bağlantısı kurulmaktadır. Kullanıcı isteğe bağlı veri gönderim özelliklerini seçebileceği gibi sunucu IP adresi ve port bilgisini girdikten sonra bu ayarların varsayılan değerlerini sunucudan da alabilmektedir.

Geliştirilen platforma genel bakış Şekil 5'te görüldüğü gibidir.

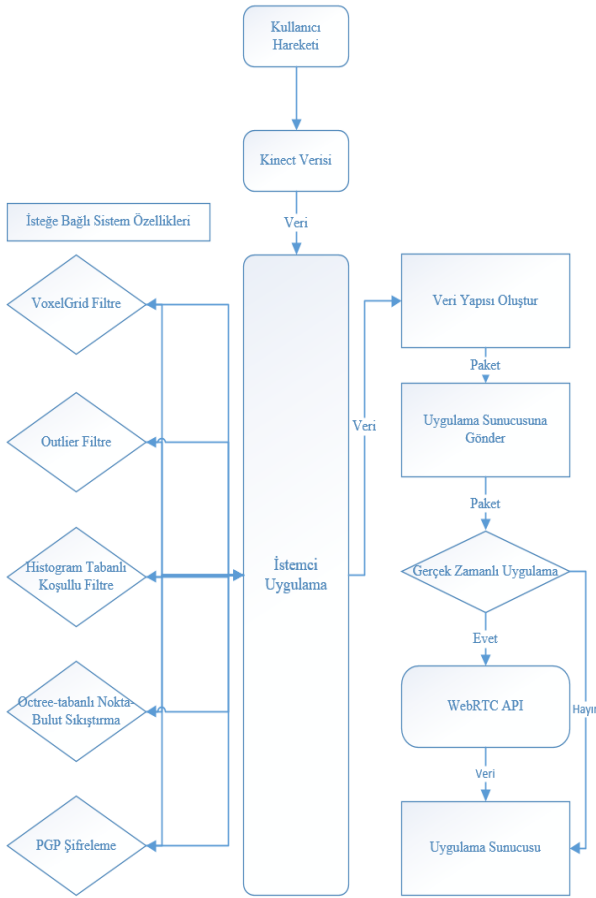


Şekil 5. Platform Genel Bakış

Kinect kameradan alınan veri bilgisayarda bulunan istemci uygulama ile yorumlanır. Bu aşamada isteğe bağlı filtreler ve şifreleme algoritmaları uygulanır. Platforma özel geliştirilen veri yapısı ile veri paket haline getirilir. Bu aşamada anlık bir uygulama için Kinect kullanılıyorsa yani gerçek zamanlı gönderim seçeneği seçilmişse WebRTC ile transfer gerçekleşir. Eğer Çevrimdışı seçeneği seçilmişse klasik veri gönderim yapılarak veri aktarımı tamamlanır. Sisteme ait akış şeması Şekil 6'da görülmektedir.

### 2.2. İsteğe Bağlı Sistem Özellikleri

Platformda daha kaliteli veri elde etmeye, nokta sayısını dolayısıyla veri boyutunu küçültmeye ve güvenliği sağlamak amacıyla veriyi şifrelemeye yönelik filtreler ve algoritmalar kullanılmaktadır. Bu filtre ve algoritmalar aşağıda açıklanmıştır. Bu aşamada NBK kullanılmıştır. NBK, açık kaynak kodlu bir proje olup, BSD lisansı koşulları altında yayınlanmış olsa da kullanımı ticari ve araştırma amaçlı kullanım için ücretsizdir.



Şekil 6. Sistem Akış Şeması

NBK yazılım çerçevesi içerisinde filtreleme, özellik tahmini ve alt örnekleme gibi son teknoloji algoritmalar barındırmaktadır [25]. Olası bir veri kaybının engellenmesi ya da bu tip durumların gerektiği uygulamalarda problem yaşanmaması için bu filtrenin kullanımı tamamen istemci ya da sunucu tarafından belirlenebilecek şekilde tasarlanmıştır.

### 2.2.1. VoxelGrid (VG) Filtre

VG alt örnekleme tekniği, bir 3D voxel ızgarasına dayanmaktadır. Bu teknik, girdi alanını bölmek ve nokta sayısını azaltmak için bilgisayar grafikleri alanında geleneksel olarak kullanılmaktadır [29, 30]. VG algoritması, 3 boyutlu uzayda bir Voxel ızgarası tanımlar ve her Voxel için bir vektör, o Voxel üzerinde bulunan tüm noktaların temsilcisi olarak seçilir. Kinect sensör tarafından istemciye gönderilen verilerin sayısının azaltılması aşamasında isteğe bağlı VG filtresi kullanılmış ve veri noktaları küçültülmüştür.

### 2.2.2. Outlier Filtre

Platformun istemci tarafında kullanılan bir diğer filtre ise outlier filtresidir. Bu filtre sayesinde Kinect sensörden elde edilen verilerin, geri kalanıyla istatistiksel olarak tutarsız olan noktalar, bir aykırı kaldırma filtresi kullanılarak

kaldırılır [31, 32]. Bu aşamada veri boyutunun küçültülmesi yanında temel hedef verilerde tutarlılığın sağlanmasıdır.

### 2.2.3. Histogram Tabanlı Koşullu Filtre

Renk histogramı, bir görüntünün renk dağılımının bir temsilidir. Dijital görüntüler için, bir renk histogramı, renk aralıklarının sabit listesinin her birinde aynı renk değerine sahip piksel sayısını temsil etmektedir [33, 34]. Ancak bazı uygulamalar, kişilerin yalnızca eklem bilgilerine ve bu eklemelerin koordinatlarına gereksinim duymaktadır. Bu gibi durumlarda Kinect verisi içerisinde bulunan gereksiz renk verisinin taşınması sistem performansını olumsuz yönde etkilemektedir. Bu gibi durumlarda performansın korunması ve gereksiz bilgilerin platformdan temizlenmesi için Histogram tabanlı koşullu filtre kullanılmıştır.

### 2.2.4. Octree-tabanlı Sıkıştırma

Üretilen verinin en az bant genişliği gereksinimi ve en az boyutlu dosya olarak gönderilmesi ve alınması platform performansı için hayati önem taşımaktadır. Bu nedenle bir sıkıştırma yöntemine ihtiyaç duyulmaktadır. Bu gereksinim için Octree tabanlı sıkıştırma kullanılmıştır. Bu sayede veri içerisinde bulunan mekânsal ve zamansal fazlalık azaltılmıştır [35, 36].

### 2.2.5. PGP Şifreleme

Günümüzde internet üzerinden iletilen verilerin güvenliğinin sağlanması farklı ve zorlu bir çalışma alanıdır. Kinect sensör kritik alanlarda kullanılmaya başlanılmış ve güvenlik gereksinimi doğmuştur. Bu nedenle geliştirilen platform üzerinde iletilen verinin güvenliğinin sağlanması için isteğe bağlı özellikler arasına güvenlik seçeneği de eklenmiştir. Açık kaynak kodlu ve kolay uygulanabilir olması sebebiyle PGP şifreleme yöntemi tercih edilmiştir. PGP yöntemi bünyesinde DSA, MD5, SHA-1 ve RSA gibi şifreleme algoritmalarını barındırmaktadır. PGP'de veriler karşılıklı kontrol edilmeden erişilemez durumdadırlar bu sayede tam güvenlik sağlanmaktadır [37, 38].

## 2.3. Veri Yapısı

Kinect sensörden elde edilen veriler büyük boyutlara ulaşmaktadır ve bu verilerin yönetilmesi ve saklanması ciddi problemlere neden olabilmektedir. Platformun stabil çalışmasını ve platformda yer alan veri bütünlüğünün sağlanması için özel bir veri yapısına ihtiyaç duyulmuştur. Kinect uygulamalarına özel geliştirilen veri yapısı Şekil 7'de görülmektedir.

İstemci tarafından seçilebilir durumda bulunan parametreler, veri yapısının başlık bilgisini oluşturmaktadır. Hemen ardından sensör tarafından tespit edilen her iskeletin her eklemine, her eklemde X, Y ve Z bilgilerinin saklandığı veri kısmından oluşmaktadır. Paketlerin sıralı şekilde saklanması ve yönetilebilmesi için her paket zaman damgası ile damgalanmaktadır.

Veri	Ekleme 1			Ekleme 2			Ekleme N			
	X	Y	Z	X	Y	Z	X	Y	Z	
İskelet 1							...			
PGP Şifreleme	1-bit									
Octree-Tabanlı Nokta-Bulut Sıkıştırma	1-bit									
Histogram Tabanlı Koşullu Filtre	1-bit									
Outlier Filtre	1-bit									
VoxelGrid Filtre	1-bit									
WebRTC	1-bit									
Zaman	64-bit									
Veri Büyüklüğü										

Şekil 7. Veri Yapısı

#### 2.4. WebRTC

WebRTC, Kinect uygulamalarında gerçek zamanlı etkileşim için ideal bir ağ platformudur. Belirlenen eşler arasında video, ses ve veri için gerçek zamanlı veri paylaşımı sağlamaktadır. Platform bağımlılığı gibi sorunları da ortadan kaldırmaktadır [27, 28, 39]. Akış verisini edinmek ve haberleşmek için, WebRTC Eş Bağlantısı, Medya Akışı ve Veri Kanalı olmak üzere üç ara katman yazılımı sağlamaktadır [39]. Geliştirilen platform WebRTC ara

katman yazılımının veri kanalı katmanını kullanmaktadır. Bu sayede geliştirilen veri yapısı uygulama sunucuna anlık olarak transfer edilmektedir.

### 3. BULGULAR

Geliştirilen platformun değerlendirilmesi aşamasında gerçek Kinect verisi kullanılmıştır. Öncelikle uygulanan filtreler ve sonuçlarında elde edilen sıkıştırma oranları değerlendirilmiştir.

Kinect sensör tarafından gönderilen verilerin sayısının azaltılması için VG filtre kullanılmış ve veri noktaları küçültülmüştür. Bu filtre sonucu elde edilen sonuçlar Tablo 2'de görülmektedir.

**Tablo 2.** VoxelGrid (VG) Filtre Sonucu

Ham Veri Noktası	VoxelGrid (VG) Filtre Sonucu	Sıkıştırma Oranı
97834	30195	3.24

Platformun istemci tarafında uygulanan bir diğer filtre ise outlier filtresidir. Sensör üzerinden elde edilen verilerde bulunan istatistiksel tutarsız noktalar bu filtre sayesinde temizlenmektedir. Bu filtre sonucu elde edilen sonuçlar Tablo 3'te görülmektedir.

**Tablo 3.** Outlier Filtre Sonucu

Ham Veri Noktası	Outlier Filtre Sonucu	Sıkıştırma Oranı
97834	87351	1.12

Geliştirilen platformun performansının korunması ve gereksiz bilgilerin platformdan temizlenmesi için Histogram tabanlı koşullu filtre kullanılmıştır. Bu filtre sonucu elde edilen sonuçlar Tablo 4'te görülmektedir.

**Tablo 4.** Histogram Tabanlı Koşullu Filtre Sonucu

Ham Veri Noktası	Histogram Tabanlı Koşullu Filtre Sonucu	Sıkıştırma Oranı
97834	70894	1.38

Sensör tarafından üretilen verinin sıkıştırılması aşamasında Octree tabanlı sıkıştırma kullanılmıştır. Bu sıkıştırma sonucunda elde edilen sonuçlar Tablo 5'de görülmektedir.

**Tablo 5.** Octree Tabanlı Sıkıştırma Sonucu

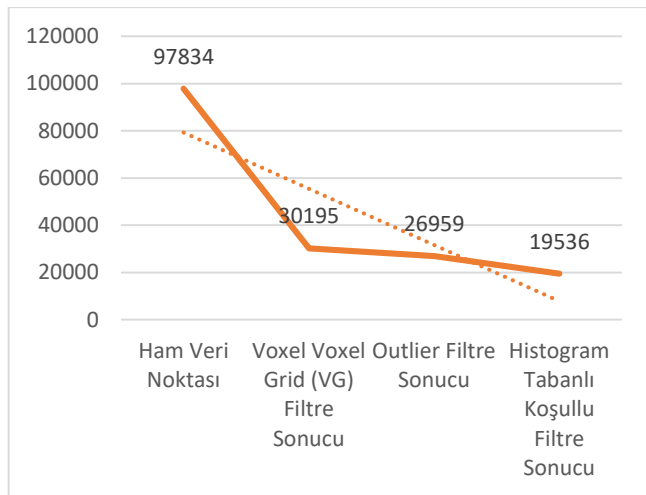
Ham Veri Dosya Boyutu (KB)	Octree Tabanlı Sıkıştırma Sonucu (KB)	Sıkıştırma Oranı
229.19	205.39	%10.381

Yapılan deneysel sonuçlar ayrı ayrı değerlendirmelerden oluşmaktadır. Ancak istemci tüm filtreleme seçeneklerini seçtiğinde filtrelemelerden elde edilen gerçek sıkıştırma verisi Tablo 6'da görülmektedir.

**Tablo 6.** Genel Sıkıştırma Sonuçları

Ham Veri Noktası	Voxel Grid (VG) Filtre Sonucu	Outlier Filtre Sonucu	Histogram Tabanlı Koşullu Filtre Sonucu	Sıkıştırma Oranı
97834	30195	26959	19536	5,007

Elde edilen filtreleme ve sıkıştırma sonuçları grafiği Şekil 8'de görüldüğü gibidir.



**Şekil 8.** Filtreleme ve sıkıştırma sonuçları grafiği

Değerlendirme aşamasında kullanılan veri boyutları elde edilen sonuçlara dair net bilgiler vermektedir. Ancak platformun gerçek başarısı için Kinect uygulamaları sonucu elde edilen büyük verilerin düşünülmesi bir gerekliliktir.

#### 4. SONUÇ VE GELECEK ÇALIŞMALAR

Kinect sensörün önemi ve kullanımı her geçen gün artmaktadır. Bu nedenle güvenli ve efektif bir veri transfer platformuna ihtiyaç duyulmuştur. Yapılan çalışma sonucunda aşağıdaki kazanımlar elde edilmiştir.

- İstemci – Sunucu mimarisine uygun Kinect veri paylaşım platformu geliştirilmiştir.
- İstemci tarafından geliştirilen mimari sayesinde farklı filtreler ve algoritmalar kullanılmıştır.
- İsteğe bağlı VoxelGrid (VG) Filtre ile nokta sayısı azaltılmıştır.
- İsteğe bağlı Outlier Filtre ile istatistiksel tutarsız noktalar temizlenmiştir.
- İsteğe bağlı Histogram Tabanlı Koşullu Filtre ile gereksiz bilgiler platformdan temizlenmiştir.

- İsteğe bağlı Octree Tabanlı Sıkıştırma elde edilen dosyalar sıkıştırılmıştır.
- Platformun stabil çalışmasını ve platformda yer alan veri bütünlüğünün sağlanması için özel bir veri yapısı geliştirilmiştir.
- İsteğe bağlı PGP şifreleme ile veri güvenliği sağlanmıştır.
- Çevrimiçi uygulamalar için WebRTC veri katmanları kullanılmış ve anlık veri paylaşımı sağlanmıştır.

Sunulan platform araştırmacılar ve geliştiriciler tarafından kullanılan Kinect uygulamalarında performans sağlayacaktır.

Geliştirilen platform için istemci bilgisayarlara ilgili dosya paketinin kurulumunun yapılması ve uygulama sunuculara tanımlama yapılması gerekmektedir. Bu nedenle geliştirilen sistemin gelecekte servis tabanlı olarak geliştirilmesi hedeflenmektedir. Bu sayede geliştiriciler ve araştırmacılar tarafından kendi yazılımları ya da uygulamaları içerisinde çağrılabilir hale getirilmesi hedeflenmektedir. Ayrıca şu anda veri yapısında bulunan iskelet, eklem ve koordinat bilgilerinin yanına yine tamamen isteğe bağlı eklemeler arası açı değerlerinin de eklenmesi ve bu sayede uygulama sunucusu tarafında ihtiyaç duyulacak hesaplama gereksiniminin azaltılması hedeflenmektedir.

#### KAYNAKÇA

- [1] Z. Zhang, "Microsoft Kinect Sensor and Its Effect," IEEE Multimedia, vol. 19, no. 2, pp. 4-10, 2012.
- [2] M. Gabel, R. Gilad-Bachrach, E. Renshaw, and A. Schuster, "Full body gait analysis with Kinect," Conf Proc IEEE Eng Med Biol Soc, vol. 2012, pp. 1964-7, 2012.
- [3] N. Kitsunezaki, E. Adachi, T. Masuda, and J. Mizusawa, "KINECT applications for the physical rehabilitation," presented at the 2013 IEEE International Symposium on Medical Measurements and Applications (MeMeA), Gatineau, QC, Canada, 2013.
- [4] B. Lange et al., "Interactive game-based rehabilitation using the Microsoft Kinect," presented at the 2012 IEEE Virtual Reality Workshops (VRW), Costa Mesa, CA, USA, 2012.
- [5] T. Dutta, "Evaluation of the Kinect sensor for 3-D kinematic measurement in the workplace," Appl Ergon, vol. 43, no. 4, pp. 645-9, Jul 2012.
- [6] I. P. T. Weerasinghe, J. Y. Ruwanpura, J. E. Boyd, and A. F. Habib, "Application of Microsoft Kinect Sensor for Tracking Construction Workers," pp. 858-867, 2012.
- [7] Z. Zhang, M. Zhang, Y. Chang, E.-S. Aziz, S. K. Esche, and C. Chassapis, "Real-Time 3D Model Reconstruction and Interaction Using Kinect for a Game-Based Virtual Laboratory," p. V005T05A053, 2013.
- [8] L. Cruz, D. Lucio, and L. Velho, "Kinect and RGBD Images: Challenges and Applications," pp. 36-49, 2012.
- [9] H. Richards-Rissetto, J. von Schwerin, and G. Girardi, "Kinect and 3D GIS in archaeology," pp. 331-337, 2012.
- [10] S. Izadi et al., "KinectFusion," p. 559, 2011.

- [11] G. Du and P. Zhang, "Markerless human-robot interface for dual robot manipulators using Kinect sensor," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 2, pp. 150-159, 2014.
- [12] S. Zolkiewski and D. Pioskowik, "Robot Control and Online Programming by Human Gestures Using a Kinect Motion Sensor," vol. 275, pp. 593-604, 2014.
- [13] R. A. El-laithy, J. Huang, and M. Yeh, "Study on the use of Microsoft Kinect for robotics applications," pp. 1280-1288, 2012.
- [14] M. Eiji, C. Meifen, M. Toshiyuki, and H. Hiroshi, "Human motion tracking of mobile robot with Kinect 3D sensor," presented at the 2012 Proceedings of SICE Annual Conference (SICE), Akita, Japan, 20-23 Aug. 2012.
- [15] Z.-R. Tsai, "Robust Kinect-based guidance and positioning of a multidirectional robot by Log-ab recognition," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1271-1282, 2014.
- [16] B. Lau, C. Sprunk, and W. Burgard, "Efficient grid-based spatial representations for robot navigation in dynamic environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1116-1130, 2013.
- [17] K. Berger, S. Meister, R. Nair, and D. Kondermann, "A State of the Art Report on Kinect Sensor Setups in Computer Vision," vol. 8200, pp. 257-272, 2013.
- [18] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. E. Saddik, "Evaluating and Improving the Depth Accuracy of Kinect for Windows v2," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4275-4285, 2015.
- [19] B. Galna, G. Barry, D. Jackson, D. Mhiripiri, P. Olivier, and L. Rochester, "Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease," *Gait Posture*, vol. 39, no. 4, pp. 1062-8, Apr 2014.
- [20] C. Raposo, J. P. Barreto, and U. Nunes, "Fast and Accurate Calibration of a Kinect Sensor," pp. 342-349, 2013.
- [21] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications," *Sensors (Basel)*, vol. 12, no. 2, pp. 1437-54, 2012.
- [22] H. Gonzalez-Jorge, B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez, and P. Arias, "Metrological evaluation of Microsoft Kinect and Asus Xtion sensors," *Measurement*, vol. 46, no. 6, pp. 1800-1806, 2013.
- [23] J. Kramer, N. Burrus, F. Ehtler, H. C. Daniel, and M. Parker, *Hacking the Kinect*. Apress, 2012.
- [24] L. Caruso, R. Russo, and S. Savino, "Microsoft Kinect V2 vision system in a manufacturing application," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 174-181, 2017.
- [25] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," pp. 1-4, 2011.
- [26] E. Bertin, S. Cubaud, S. Tuffin, N. Crespi, and V. Beltran, "WebRTC, the day after: What's next for conversational services?," pp. 46-52, 2013.
- [27] S. Loreto and S. P. Romano, "Real-Time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts," *IEEE Internet Computing*, vol. 16, no. 5, pp. 68-73, 2012.
- [28] J. Alan and B. Daniel, *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*. Digital Codex LLC, 2014.
- [29] C. Connolly, "Cumulative generation of octree models from range data," vol. 1, pp. 25-32, 1984.
- [30] L. Kobbelt and M. Botsch, "A survey of point-based techniques in computer graphics," *Computers & Graphics*, vol. 28, no. 6, pp. 801-814, 2004.
- [31] H. Tsaknakis and P. Papantoni-Kazakos, "Outlier resistant filtering and smoothing," *Information and Computation*, vol. 79, no. 2, pp. 163-192, 1988.
- [32] Y. Wang and H.-Y. Feng, "Outlier detection for scanned point clouds using majority voting," *Computer-Aided Design*, vol. 62, pp. 31-43, 2015.
- [33] T. Kaur and R. K. Sidhu, "Performance Evaluation of Fuzzy and Histogram Based Color Image Enhancement," *Procedia Computer Science*, vol. 58, pp. 470-477, 2015.
- [34] V. Rajinikanth and M. S. Couceiro, "RGB Histogram Based Color Image Segmentation Using Firefly Algorithm," *Procedia Computer Science*, vol. 46, pp. 1449-1457, 2015.
- [35] J. Navarrete, D. Viejo, and M. Cazorla, "Compression and registration of 3D point clouds using GMMs," *Pattern Recognition Letters*, vol. 110, pp. 8-15, 2018.
- [36] V. Morell, S. Orts, M. Cazorla, and J. Garcia-Rodriguez, "Geometric 3D point cloud compression," *Pattern Recognition Letters*, vol. 50, pp. 55-62, 2014.
- [37] H. Meyer, "Privacy better than 'Pretty Good'," *Computers & Security*, vol. 16, no. 7, p. 620, 1997.
- [38] B. Zajac, "Pretty good privacy," *Computer Fraud & Security Bulletin*, vol. 1994, no. 9, pp. 14-17, 1994.
- [39] M. Becke, E. P. Rathgeb, S. Werner, I. Rungeler, M. Tuxen, and R. Stewart, "Data channel considerations for RTCWeb," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 34-41, 2013.