

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇOKLU KINECT KULLANIMIYLA ELDE EDİLEN İSKELET HAREKET
VERİLERİNİN BİRLEŞTİRİLMESİ VE KARŞILAŞTIRILMASI**

MEHMED OĞUZ ŞEN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TEMMUZ 2016

Tezin Bařlıđı : Çoklu Kinect Kullanımıyla Elde Edilen İskelet Hareket Verilerinin Birleřtirilmesi ve Karřılařtırılması

Tezi Hazırlayan : Mehmed Ođuz ŐEN

Sınav Tarihi : 11.07.2016

Yukarıda adı geen tez jürimizce deđerlendirilerek Bilgisayar Mühendisliđi Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiřtir.

Sınav Jüri Üyeleri

Tez Danıřmanı : **Yrd. Do. Dr. Metin ERTÜRKLER**

İnönü Üniversitesi

Yrd. Do. Dr. Ahmet INAR

Fırat Üniversitesi

Yrd. Do. Dr. A. Fatih KOCAMAZ

İnönü Üniversitesi

Prof. Dr. Alaattin ESEN

Enstitü Müdürü

ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduđum “Çoklu Kinect Kullanımıyla Elde Edilen İskelet Hareket Verilerinin Birleřtirilmesi ve Karřılařtırılması” bařlıklı bu çalıřmanın bilimsel ahlak ve geleneklere aykırı düřecek bir yardıma bařvurmaksızın tarafımdan yazıldıđını ve yararlandıđım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden olduđunu belirtir, bunu onurumla dođrularım.

Mehmed Ođuz řEN

ÖZET

Yüksek Lisans Tezi

ÇOKLU KINECT KULLANIMIYLA ELDE EDİLEN İSKELET HAREKET VERİLERİNİN BİRLEŞTİRİLMESİ VE KARŞILAŞTIRILMASI

Mehmed Oğuz Şen

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

70 + x sayfa

2016

Danışman: Yrd. Doç. Dr. Metin Ertürkler

Bu yüksek lisans tezinde Microsoft'un ürettiği Kinect derinlik algılayıcı kameraların hareket yakalamada nasıl kullanılacağı uygulamalı olarak gösterilmektedir. Tez kapsamında Kinect kullanılarak tüm vücut hareketlerinin yakalanması ve işlenmesi üzerine odaklanılmıştır.

Literatüre iki önemli katkı sağlayan bu tezde sınıflandırma problemi olarak ele alınan hareket tanılama probleminin çözümüne farklı bir bakış açısı ile yaklaşılmıştır. Bu yaklaşımda hareket yakalama verileri ile hareket yörüngeleri oluşturulmuş ve bu yörüngeler Fréchet uzaklığı yöntemiyle birbirleriyle karşılaştırılarak hareket benzerliği ölçülmüştür. Elde edilen deneysel sonuçlar, referans harekete benzeyen hareketin diğer hareketlerden daha düşük Fréchet uzaklık değerlerine sahip olduğunu ve dolayısıyla daha benzer nitelikte olduğunu göstermiştir. Bu sayede uzuv hareketlerinin yörüngelerinin benzerliği bir sınıflandırıcı kullanılmaksızın doğrudan birbirleriyle kıyaslanabilmektedir.

Literatüre sağlanan bir diğer katkı ise nesnelerin birbirini örtmesi (occlusion) ve öz kaynaklı örtme (self occlusion) problemlerinin çözümü için çoklu kameralardan elde edilen iskelet eklem verilerinin birleştirilmesinde parçacık filtresinin kullanımının önerilmesidir. Özellikle, otonom robot hareketinde karşılaşılan SLAM probleminin çözümünde kullanılan parçacık filtresi algoritmasının çoklu Kinect iskelet eklemi verisinin birleştirilmesinde nasıl kullanılabileceği üzerinde çalışılmıştır. Yapılan uygulamada birbirine dik bir şekilde konumlandırılmış 2 adet Kinect kameradan elde edilen iskelet verileri birleştirilmekte ve elde edilen deneysel sonuçlar önerilen yaklaşımın örtme problemlerine çözüm sağladığını göstermektedir.

ANAHTAR KELİMELER: Kinect, işaretçisiz hareket yakalama, hareket benzerliği, Fréchet uzaklığı, eklem birleştirme, parçacık filtresi, örtme problemi

ABSTRACT

M.S. Thesis

FUSION AND COMPARISON OF SKELETON MOTION DATA ACQUIRED BY USING MULTIPLE KINECTS

Mehmed Oğuz Şen

İnönü University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

70 + x pages

2016

Advisor: Asst. Prof. Dr. Metin Ertürkler

In this master thesis, how Kinect depth sensor cameras are used at motion capture is demonstrated in practice. It is focused on full body motion capture and processing by using Kinect in the scope of thesis.

A solution for motion recognition problem, which is addressed as a classification problem in the literature, is approached with a different perspective in this thesis providing two important contributions to literature. In this approach motion trajectories are formed by motion capture data and similarity of these trajectories are measured by comparing them with Fréchet distance method. Experimental results show that the motion that is similar to reference motion has less Fréchet distance values than other motions and so that is such as to more similar. Thus, similarity of trajectories of limb movements can directly be checked against each other without using a classifier.

Another contribution to literature is the suggestion of using particle filters in fusion of skeleton joint data obtained by multiple Kinect cameras for the solution of occlusion and self-occlusion problems.

Particularly, it is studied on how particle filter algorithm, which is used for solution of SLAM problem encountered in autonomous robot movement, can be used at multiple Kinect skeleton data fusion. In the application performed, skeleton data acquired from two orthogonally positioned Kinects are fused and experimental results obtained indicate that proposed approach provides a solution for occlusion problems.

KEYWORDS: Kinect, markerless motion capture, motion similarity, Fréchet distance, joint fusion, particle filter, occlusion problem

TEŐEKKÜR

Tez konusunun Őekillenmesinde ve alıŐmalarım boyunca gsterdiĐi anlayıŐ ve ilgi, verdiĐi her trl destek ve katkılarından tr danıŐman hocam Sayın Yrd. Do. Dr. Metin ERTRKLER' e; paracık filtresi algoritmasını, tez alıŐmamda nasıl kullanacaĐım konusundaki deĐerli yardım ve ynlendirmelerinden tr Sayın Do. Dr. M. Fatih TALU' ya; alıŐmalarım esnasında laboratuvar, tehizat vb. altyapı imkânlarını ve deneyimlerini benden esirgemeyen baŐta blm baŐkanı Sayın Do. Dr. Celaleddin YEROĐLU olmak zere, blmmzn deĐerli Đretim yeleri ve araŐtırma grevlilerine; deney dzeneĐinin oluŐturulması srecinde ihtiya duyduĐum paraların 3B yazıcı ile retilmesini saĐlayan Makine MhendisliĐi Blm Đretim yesi Sayın Yrd. Do. Dr. Erkan BAHE' ye; btn Đrenim hayatım boyunca olduĐu gibi yksek lisans alıŐmalarımda da gsterdikleri sabır, destek ve ilgilerinden tr deĐerli aileme teŐekkrlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ.....	ix
SİMGELER VE KISALTMALAR.....	x
1. GİRİŞ	1
1.1 Hareket Yakalama ve Optik Tabanlı Sistemler.....	2
1.2 İşaretçi Tabanlı Hareket Yakalama Sistemleri.....	3
1.3 İşaretçisiz Hareket Yakalama Sistemleri	5
1.3.1 Uçuş Zamanı	5
1.3.2 Kızılötesi Işın Projeksiyonu	6
1.4 Tezin Amacı ve Kapsamı	7
2. KINECT KAMERA SİSTEMİ	9
2.1. Kinect'in Teknik Özellikleri	9
2.2. Kinect Yazılım Bileşenleri.....	12
2.3. Kinect'in İnsan Pozu ve İskeletinin Tespiti	13
2.4. Kinect ile Hareket Yakalama	19
2.5 Kinect Araştırma Konuları ve Akademik Çalışmalar	21
3. HAREKET YÖRÜNGELERİNİN BENZERLİK ÖLÇÜMÜ	22
3.1 Dinamik Zaman Bükmesi	23
3.2 Hausdorff Uzaklığı.....	27
3.3 Fréchet ve Kesikli Fréchet Uzaklığı.....	29
3.4 Probleme Uygun Yöntemin Belirlenmesi	32
3.5 3B Eğrileri Karşılaştırmak	33

4. ÇOK SAYIDA KINECT İLE VERİ ELDESİ	34
4.1 Çoklu Kinect Ortamının Hazırlanması.....	36
4.2 Kinect İskelet Verilerinin Birleştirilmesi.....	42
4.2.1 SLAM ve İskelet Eklemi Takibi Problemine Uyarlanması	42
4.2.2 Parçacık Filtresi Algoritması	43
5. UYGULAMALAR	46
6. SONUÇLAR VE DEĞERLENDİRME.....	64
KAYNAKÇA.....	66
TEZDEN ÜRETİLEN YAYIN VE SUNUMLAR.....	70

ŞEKİLLER LİSTESİ

Şekil 1.1	Vicon hareket yakalama sistemi ile oluşturulmuş hareket yakalama stüdyosu [6].....	4
Şekil 1.2	Vicon pasif işaretçili hareket yakalama sistemi bileşenleri [7]	4
Şekil 1.3	İnsan ve otonom nesnelere için Vicon hareket yakalama sistemi	5
Şekil 2.1	Kinect For Windows	9
Şekil 2.2	Kinect bileşenlerinin kamera üzerinde gösterimi [15].....	10
Şekil 2.3	Kinect derinlik ölçüm aralığı [17].....	11
Şekil 2.4	Kinect' in derinlik ölçümü [16].....	11
Şekil 2.5	Bir Kinect uygulamasının yazılım ve donanım etkileşimi [19].....	12
Şekil 2.6	Kinect SDK Mimarisi [19].....	13
Şekil 2.7	Portmantoya asılı ceketin neden olduğu ikinci kullanıcı iskeleti	15
Şekil 2.8	Model tabanlı çıkarımla Kinect eklem noktalarını belirleme süreci [18]16	
Şekil 2.9	3B görünüm [18].....	16
Şekil 2.10	İskelet takibi için Kinect' e yönelmek gerekir.....	18
Şekil 2.11	Kinect 2 kişinin iskeletini olmak üzere 6 kişiyi aynı anda takip edebilmektedir.....	19
Şekil 2.12	Kinect iskeleti eklem noktalarının isimlendirilmesi	20
Şekil 2.13	Kinect araştırma konularına genel bakış.....	21
Şekil 3.1	Örnek eklem koordinatlarının 80 görüntü karesi boyunca kesikli hareketi.....	22
Şekil 3.2	80 noktadan oluşan çokgensel eğri yörüngesi	23
Şekil 3.3	X ve Y dizilerinin Manhattan uzaklığı yerel maliyet ölçütü olarak kullanıldığında oluşan maliyet matrisinde koyu renkler düşük maliyeti göstermektedir [29]	24
Şekil 3.4	9 elemanlı X ve 7 elemanlı Y dizileri için geçerli ve geçersiz bükme yolu çizimleri [29].....	25
Şekil 3.5	(a) Şekil 3.3' teki maliyet matrisi ve (b) birikimli maliyet matrisi []	26
Şekil 3.6	DTW ile daha gerçekçi benzerlik ölçümü	26
Şekil 3.7	İçinde arama yapılacak görüntü örneği [34]	28
Şekil 3.8	Aranacak görüntü şablonu [34].....	28
Şekil 3.9	Görüntü şablonunun ana görüntüde bulunması [34].....	29
Şekil 3.10	İki doğru parçası üzerindeki “tasma” için boş alanın gösterimi [35].....	30

Şekil 3.11 İki çokgensel eğri arasındaki boş alan diyagramı [35]	30
Şekil 3.12 Fréchet ve kesikli Fréchet uzaklıkları	31
Şekil 3.13 Aralarındaki Hausdorff mesafesi düşük, Fréchet mesafesi yüksek olan A ve B eğrileri.....	32
Şekil 3.14 (a) 3B bir spline eğrisinin (b) XY izdüşümü ve (c) XZ izdüşümü	33
Şekil 4.1 Kinect IR kamerası, diğer Kinect'lerin yaydığı ışın demetlerini algılar.	37
Şekil 4.2 Tek Kinect ile elde edilen derinlik görüntüsüne (üstte) ikinci Kinect' in neden olduğu ışın girişiminin etkisi (altta)	37
Şekil 4.3 Titreşim motorunun Kinect'e takılması (solda) Kinect'in serbest titreşim için akrilik çerçeveye oturtulması (sağda) [46].....	38
Şekil 4.4 OmniKinect kurulumunun (a) plan görünümü (b) 3B görünümü [47]....	39
Şekil 4.5 Kinect' in tükettiği bant genişliği	39
Şekil 4.6 Aynı USB denetleyiciye bağlı ikinci Kinect'in çalışmaması	40
Şekil 4.7 Farklı USB denetleyicilerine bağlı sorunsuz çalışan Kinect' ler.....	41
Şekil 4.8 Çoklu Kinect for Windows v2 sistemi örneği.....	41
Şekil 5.1 Fréchet uzaklığı ölçümü uygulamasının genel işleyişi	46
Şekil 5.2 Örnek hareketin başlangıç ara ve bitiş pozisyonları	47
Şekil 5.3 Referans hareket ve karşılaştırılacak hareketlerin verileri.....	48
Şekil 5.4 Çoklu Kinect deney düzeneği	49
Şekil 5.5 Ölçüm aşamasında parçacıklar ortama rastgele dağıtılarak ölçüm değerleri alınır	51
Şekil 5.6 Ağırlık güncelleme aşamasında "aday" olmayan parçacıklar elenir.....	52
Şekil 5.7 Yeniden örnekleme aşamasında yeni parçacıklar oluşturulur	53
Şekil 5.8 Hareket güncelleme aşamasında parçacıklar hareket modeline göre hareket ettirilir	53
Şekil 5.9 Bir sonraki iterasyonda ölçüm ve ağırlık güncelleme aşamaları	54
Şekil 5.10 Birinci Kinect'e dönük kullanıcının 12. ekleminin birinci Kinect verileri	55
Şekil 5.11 Birinci Kinect'e dönük kullanıcının 12. ekleminin ikinci Kinect verileri	55
Şekil 5.12 Birinci Kinect'e dönük kullanıcının 12. ekleminin birleştirilmiş verileri	56
Şekil 5.13 İkinci Kinect'e dönük kullanıcının 12. ekleminin birinci Kinect verileri	57

Şekil 5.14	İkinci Kinect'e dönük kullanıcının 12. ekleminin ikinci Kinect verileri	57
Şekil 5.15	İkinci Kinect'e dönük kullanıcının 12. ekleminin birleştirilmiş verileri	58
Şekil 5.16	Kullanıcı 12. ekleminin ortam referans sistemine göre 1. Kinect koordinatları	59
Şekil 5.17	Kullanıcı 12. ekleminin birinci Kinect'e göre 1. Kinect koordinatları ..	59
Şekil 5.18	Kullanıcı 12. ekleminin ortam referans sistemine göre 2. Kinect koordinatları	60
Şekil 5.19	Kullanıcı 12. ekleminin ikinci Kinect'e göre 2. Kinect koordinatları	60
Şekil 5.20	Karşıya dönük kullanıcı 12. ekleminin birleştirilmiş koordinatları	61
Şekil 5.21	Engel birinci Kinect önüdeyken kullanıcı 12. ekleminin birinci Kinect verileri	62
Şekil 5.22	Engel birinci Kinect önüdeyken kullanıcı 12. ekleminin ikinci Kinect verileri	62
Şekil 5.23	Karşıya dönük kullanıcı 12. ekleminin birleştirilmiş koordinatları	63

TABLÖLAR LİSTESİ

Çizelge 2.1 Modelden bağımsız ve model tabanlı çıkarımların karşılaştırılması 17

Çizelge 5.1 Hareketlerin XY izdüşümlerinin aralarındaki Fréchet uzaklığı 48

SİMGELELER VE KISALTMALAR

DTW	Dynamic Time Warping
ELM	Extreme Learning Machine
FPS	Frame Per Second
ICCV	International Conference on Computer Vision
IEEE	Institute of Electrical and Electronics Engineers
RGB	Red Green Blue
RGB-D	Red Green Blue – Depth
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
SVM	Support Vector Machine
USB	Universal Serial Bus

1. GİRİŞ

İnsan hareketlerinin analitik biçimde değerlendirilmesi, bilgisayarlı görme, robotik ve insan-bilgisayar etkileşimi alanlarındaki çeşitli uygulamalarda kullanılmaktadır. Bu alanlardaki çalışmalarda kullanılan veriler, hareketin analiz edileceği ortama yerleştirilen hareket sensörleriyle veya kamera kayıtlarına görüntü işleme algoritmalarının uygulanması sonucunda elde edilmektedir.

Hareket yakalama; optik, mekanik veya manyetik sensörler kullanılarak gözlemlenen insan veya nesnelerin hareket verilerini elde etme, hareketi tanımlayan özellikleri çıkararak (genellikle eklem noktaları ve koordinat değerleri) 3B grafik modeline aktarma ve istenilen canlandırmayı yaptırabilme sürecidir.

Hareket yakalama oyun geliştirme, sinema vb. grafik uygulamalarında 3B animasyonların gerçekçiliğini artırma ve animasyonu kolaylaştırma amacıyla yaygın olarak kullanılmaktadır. Ayrıca spor (biyomekanik), askeri ve tıbbi uygulamalar olmak üzere bilgisayarlı görme ve robotik alanlarındaki araştırmalarda elde edilen bulguların doğrulanması için de hareket yakalama sistemlerinin yüksek hassasiyetli ölçümlerine ihtiyaç duyulmaktadır.

Hareket yakalama, kullanım amacına göre yüz ifadesi yakalama (facial motion capture) veya tüm vücut hareketlerini yakalama (full body motion capture) şeklinde de sınıflandırılabilir. Bu çalışmada Kinect kullanılarak tüm vücudun hareketlerinin yakalanması ve işlenmesi üzerine odaklanılmıştır. Kinect ile yüz takibi bu çalışmanın kapsamı dışındadır.

Hareket yakalama ile canlandırma oluşturma sürecinde ilk olarak ihtiyaç duyulan hareket verisi için hareket yakalama sisteminin ve ortamının ihtiyaca uygun şekilde planlanması gerekir. Uygun sistem oluşturulduktan sonra hareket yakalama için kayıt alınır ve bütünlük bozucu veriler ayıklanarak temiz veriler elde edilir. Bu verilerden hareketler oluşturulur ve canlandırılacak 3B modele aktarılır.

Hareket yakalama ile insansı robotlar başta olmak üzere çeşitli robot hareketlerinin analizi için kesin referans değerleri elde edilebilmektedir. Field, Stirling, Naghdy ve Pan robotikteki hareket yakalamayla ilgili araştırma konuları ve uygulama alanlarını ele almakta ve deney sonuçlarını değerlendirmektedir [1].

Hareket yakalama için optik, eylemsiz (inertial), mekanik ve manyetik tabanlı sistemler kullanılmakta olup; optik tabanlı sistemler yaygın olarak kullanılmaktadır. Bütün bu sistemlerin performansı; hareketi yakalanacak kişi veya nesnenin sayısı,

hareketin yakalanacağı ortamın ebatları, yakalanacak hareket verisinin hassasiyeti vb. faktörlere bağlıdır.

Literatürde hareket tanılama bir sınıflandırma problemi olarak ele alınmakta, SVM, ELM vb. sınıflandırıcıların eğitiminde işaretçi tabanlı hareket yakalama sistemlerinden elde edilen veri setleri kullanılmaktadır. Kinect ve benzeri RGB-D sistemlerle elde edilen hareketin tanımlanması, hareket verisinin eğitilmiş sınıflandırıcılarla elde edilen hangi hareket grubuna ait olduğunun belirlenmesiyle yapılmaktadır [2].

1.1 Hareket Yakalama ve Optik Tabanlı Sistemler

Optik tabanlı sistemler, hareketi yakalanacak kişi veya nesnelerin kameralar aracılığıyla takip edilerek, kameralardan elde edilen verilerle kişi veya nesnenin 3B konumunun nirengi (triangulation) yapılarak bulunmasını sağlar. Bu sistemlerde kullanılan kameralar standart RGB kameralardan farklı nitelikte olup, derinlik algılama veya belirli renkleri algılama özelliklerine sahip olabilmektedir.

Optik tabanlı sistemlerde farklı kameralardan elde edilen hareket verilerinin işlenmesi ve animasyonun 3B modele giydirilmesi için yazılımlar da kullanılmaktadır. Optik tabanlı sistemler işaretçi tabanlı veya işaretçi kullanmaksızın hareket yakalama işlemini gerçekleştirebilmektedir.

Optik tabanlı hareket yakalama sistemlerinde hareketi yakalanacak nesnelerin kameralar tarafından fark edilemediği durumlar oluşabilmektedir. Kayıt alınan ortamdaki diğer nesnelerin hareketi yakalanacak nesne ile kamera arasına girmesi, kameranın nesneyi takip edememesine ve hareket verisinin elde edilememesine yol açmaktadır. Bu durum nesnelerin birbirini örtmesi (occlusion) olarak ifade edilmektedir. Ayrıca insan veya insansı robotların hareketi esnasında oluşan bazı duruşlarda uzuvların (kol, bacak vb.) vücudun bir kısmını örtmesiyle bazı bölgeler kamera tarafından görülememektedir. Bu durum ise öz kaynaklı örtme (self occlusion) olarak ifade edilir.

İşaretçi tabanlı optik hareket yakalama sistemlerinde bu problemlerin çözümü için farklı açılardan hareket takibini sağlayan çok sayıda kamera kullanılarak yüksek frekansta görüntü kareleri elde edilmekte ve hareket yakalama verilerindeki düzensizlikler giderilebilmektedir. Örtme probleminin işaretçisiz optik hareket

yakalama sistemlerindeki çözümleri için farklı yöntemler önerilmektedir [3], [4].

1.2 İşaretçi Tabanlı Hareket Yakalama Sistemleri

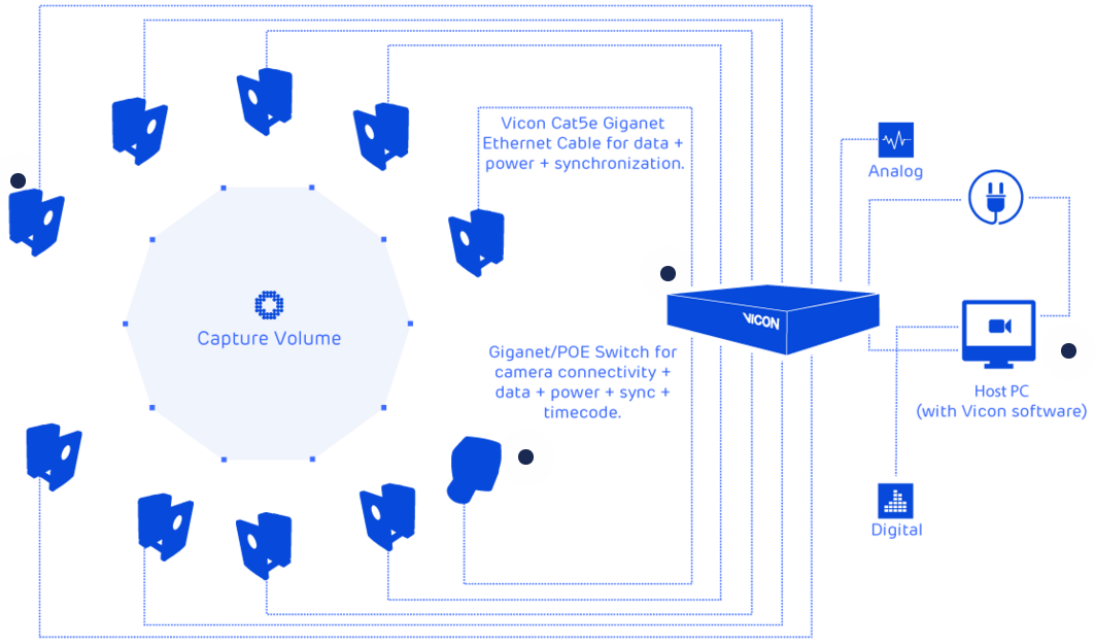
Optik tabanlı sistemlerde işaretçi kullanımı ile kullanıcıların hareketlerini temsil edecek özellikleri tanımlayan eklem noktaları belirlenmektedir. Bu eklem noktaları ile sistemin algıladığı kullanıcı (performans aktörleri) silüetlerinin iskelet yapısı oluşturulmakta ve hareketin bütünlüğü sağlanmaktadır. Elde edilmek istenen canlandırmanın kullanım alanına ve ihtiyaç duyulan ortama göre sistemde kullanılacak işaretçi sayısı değişebileceğinden, işaretçi hareket yakalama sistemleri oldukça esnek bir şekilde ayarlanıp kurulabilir. İşaretçi tabanlı bir hareket yakalama sistemi Şekil 1.1 ile gösterilmiştir.

İşaretçi tabanlı hareket yakalama sistemleri pasif ve aktif işaretçi tabanlı sistemler olmak üzere ikiye ayrılır. Pasif işaretçi tabanlı sistemlerde kullanıcının üzerine geri yansıtıcı (retroreflective) maddeyle kaplı işaretçiler yerleştirilir ve kamera yakınlarından sahneye gönderilen ışık işaretçilerden yansır. Bu ışığın rengine duyarlı yüksek hızlı (500 FPS veya daha fazlası) kameralar, işaretçilerin yüzeydeki diğer nesnelere izole edilmesini sağlayan görüntü verilerini kaydeder. Elde edilen görüntülerin kalitesi kameranın (megapiksel cinsinden) çözünürlük değerlerine bağlıdır. İşaretçilerin 3B konumlarının nirengi yapılabilmesi ve işaretçilerin görünürlüğünün sağlanması için çok sayıda kameraya ihtiyaç duyulmaktadır [5]. Bu kameralardan gelen veriler merkezi bir sunucuda birleştirilerek, kullanıcı üzerindeki her bir pasif işaretçinin 3B konum verisi Şekil 1.2'deki gibi elde edilebilir.



Şekil 1.1 Vicon hareket yakalama sistemi ile oluşturulmuş hareket yakalama stüdyosu [6]

Pasif işaretçiler ayırt edici sinyaller gönderen herhangi bir elektronik donanımı içermediğinden hareket yakalama sisteminin işaretçileri birbirlerinden ayırt edebilmesi gerekir. Bunun için işaretçilerin peş peşe gelen görüntü kareleri boyunca tutarlılığına bakılmakta ve bu tutarlılık görüntü karesi başına elde edilen konumların sürekliliğiyle belirtilmektedir. Yüksek görüntü karesi oranına (frame rate) sahip görüntü verileriyle işaretçilerin 3B konum verilerinin sürekliliği sağlanmaktadır [5].



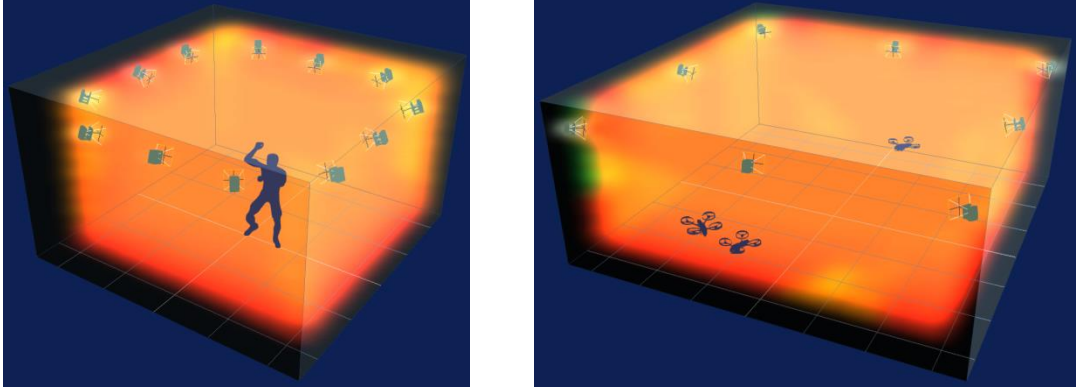
Şekil 1.2 Vicon pasif işaretçili hareket yakalama sistemi bileşenleri [7]

Aktif işaretçi tabanlı sistemlerde işaretçiler ışık yansıtmak yerine üzerlerindeki LED lambalar aracılığıyla (kızılötesi) ışık yayarlar. Bu sistemlerde işaretçilerin 3B konumlarını nirengi yaparken (triangulation) bir anda işaretçilerden sadece bir tanesi (çok hızlı bir şekilde) yakılarak veya bütün işaretçileri aynı anda farklı genlik ve frekanslarda yakarak işaretçilerin görelî konumları belirlenebilir. İşaretçi LED' inin iki farklı şekilde aydınlatılması ile bir işaretçiyi tanımlamak mümkündür. Pasif işaretçilerde olduğu gibi çok sayıda kamera, bu işaretçilerden gelen ışığa duyarlı ve yüksek hızlı (500 FPS veya daha fazlası) çalışacak şekilde ayarlanmaktadır. Aktif işaretçiler ışık yaydıklarından çalışabilmesi için pil vb. elektrik enerjisi kaynaklarına ihtiyaç duyarlar [1].

İşaretçilerin düzgün bir şekilde yerleştirilebilmesi için özel tasarlanan

kıyafetler kullanıldığı gibi [8], (özellikle pasif işaretçilerin) doğrudan vücuda yapıştırılması da söz konusudur. Tasarlanan kıyafetlerin renklerinin işaretçileri takip edecek kameraların algılayacağı ışık rengiyle karışmaması önemlidir.

Şekil 1.3'de görüldüğü gibi quadcopter vb. otonom nesnelerin hareketlerinin takibinde hareket yakalama ile konum doğrulama verileri elde edilebilmektedir.



Şekil 1.3 İnsan ve otonom nesneler için Vicon hareket yakalama sistemi

1.3 İşaretçisiz Hareket Yakalama Sistemleri

Görüntü işleme ve bilgisayarlı görme alanındaki gelişmeler, işaretçi tabanlı hareket yakalama sistemlerine alternatif olarak işaretçisiz hareket yakalama sistemleri yaklaşımını ortaya koymuştur. İşaretçi tabanlı sistemlerin maliyetleri ve uygulamadaki düzenek ve uygun ortam kurma gereklilikleri, işaretçisiz sistemleri ön plana çıkarmaktadır.

İşaretçi tabanlı sistemlerde hareketi tanımlayan eklem noktaları işaretçilerle belirtilirken, işaretçisiz sistemlerde bu eklem noktalarının görüntü verilerinden elde edilmesi gerekmektedir. Standart RGB kameralar 3B ortamın perspektif izdüşümünü alarak 2B görüntü verilerini oluşturduğundan, 3B koordinat verilerini tespit edebilmek için kameraların görüntülediği ortamdaki nesnelerin derinlik bilgisine ihtiyaç duyulmaktadır. Bu derinlik bilgisi uçuş zamanı (time-of-flight) [9] veya kızılötesi ışın projeksiyonu [3] ile elde edilebilmektedir.

1.3.1 Uçuş Zamanı

Uçuş zamanı yöntemi, görüntüleme yapılacak ortama ışık atımları (light

pulse) gönderilmesi ve ortamdan yansıyan ışınların kamera merceğiyle toplanarak kendisine gelen ışığın gidiş-dönüş süresine bağlı olarak yüzeyin mesafesini bulmaya dayanmaktadır. Bir ışık atımı 1 GHz frekanstaki bir saat çevriminde yaklaşık 30 cm mesafe almaktadır. Işık atımı ve ölçümler daha kısa bir zaman aralığında yapıldığında ışığın kat ettiği mesafe daha hassas bir şekilde ölçülebilmektedir. Bu yöntemde her pikselin birbirinden bağımsız olarak derinliği ölçülebilmektedir.

Kinect for Windows v2 derinlik sensör sistemi uçuş zamanı yöntemiyle çalışmaktadır. Söz konusu sistemde arka plan ışığını engelleme özelliği de yer aldığından, arka plan ışığının baskın olduğu ortamlarda da derinlik ölçümü yapılabilmektedir.

Grzegorzek, Theobalt, Koch ve Kolb derinlik görüntüleme ve derinlik verisi işleme alanlarındaki çalışmaları ve insan odaklı uygulamaları derledikleri kitabında uçuş zamanı (time-of-flight) kameraları kullanılarak derinlik görüntüleri elde etme konusunu ayrıntılı şekilde ele almaktadır [10].

1.3.2 Kızılötesi Işın Projeksiyonu

Kinect for Windows ve Kinect for Xbox 360 kameralarında kullanılan bu yöntemde ortama gönderilen kızılötesi ışın demetlerinin, ortamda bulunan cisimlerden yansıyan ışın demetleriyle nirengi yapılarak ortamdaki bir cismin derinliği ölçülmektedir. Bu yöntemin detayları ikinci bölümde Kinect kamera sisteminin teknik özellikleri kısmında açıklanmaktadır.

İşaretçisiz hareket yakalama sistemleri, görüntü bölütlendirme ve sınıflandırma işlemleri için eğitim verisi olarak hareketi yakalanacak nesnelere çeşitli pozisyonlardaki görüntülerine ihtiyaç duyarlar. Bu görüntülerden elde edilen silüetler kullanılarak, önceden oluşturulan vücut yüzeyi ve iskelete sahip 3B insan modellerinin uzuvlarına oturtulması için 2B-3B poz tahminleri yapılır. Bu tahminin hassasiyeti ve gerçekçiliği, dolayısıyla oluşturulacak canlandırmanın kalitesi çok sayıda kamera kullanılarak artırılabilir.[11]

Biyomekanik uygulamalar için yüksek hızlı kameralarla (200 Hz'e kadar) elde edilen görüntü verilerini, daha önce oluşturulmuş morfolojik ve kinematik modellerle eşleştirerek, eklem noktalarını ve açılarını hesaplayabilen bir işaretçisiz hareket yakalama sistemi önerilmiştir [12]. Önerilen bu sistem, eklem merkezlerini uygun hassasiyetle tanımlayarak insan vücudu kinematığını hesaplayabilmektedir

[12], [13].

Bu tez çalışmasında işaretçisiz hareket yakalama sistemi olarak Kinect for Windows kullanılmıştır. Kinect for Windows sağladığı iskelet verisi ile hareket yakalamada kullanılabilir. Ismail ve Basah, Kinect ile insan hareket analizinin rehabilitasyon ve biyomekanik uygulamalarını konu aldıkları çalışmalarında Kinect'in söz konusu uygulamalardaki performansını değerlendirmişlerdir [14]. Tez kapsamında önerilen yaklaşımın rehabilitasyon, biyomekanik vb. alanlarda kullanılabilmesi öngörülmektedir.

1.4 Tezin Amacı ve Kapsamı

Bu tezde Microsoft'un ürettiği Kinect derinlik algılayıcı kameraların hareket yakalamada nasıl kullanılacağına uygulamalı olarak gösterilmesi; elde edilen hareket yakalama verileri ile oluşturulan yörüngelerin farklı algoritmalar ele alınarak karşılaştırılması suretiyle hareket benzerliğinin ölçülmesi; nesnelerin birbirini örtmesi (occlusion) ve öz kaynaklı örtme (self occlusion) problemlerinin çözümü için, çoklu kameralardan elde edilen iskelet eklem verilerinin parçacık filtresi kullanılarak birleştirilmesi amaçlanmaktadır.

Bu tezin literatüre iki önemli katkısı bulunmaktadır.

1) Literatürde sınıflandırma problemi olarak ele alınan hareket tanılama probleminin çözümüne farklı bir bakış açısı ile yaklaşmıştır. Bu yaklaşımda hareket yakalama verileri ile hareket yörüngeleri oluşturulmuş ve bu yörüngeler Fréchet uzaklığı yöntemiyle birbirleriyle karşılaştırılarak hareket benzerliği ölçülmüştür. Bu sayede uzuv hareketlerinin yörüngelerinin benzerliği bir sınıflandırıcı kullanılmaksızın doğrudan birbirleriyle kıyaslanabilmektedir.

2) Nesnelerin birbirini örtmesi (occlusion) ve öz kaynaklı örtme (self occlusion) problemlerinin çözümü için çoklu kameralardan elde edilen iskelet eklem verilerinin birleştirilmesinde parçacık filtresinin kullanılması önerilerek örtme probleminin çözümüne yeni bir katkı sağlanmaktadır.

Yapılan uygulamalarda birbirine dik bir şekilde konumlandırılmış 2 adet Kinect kameradan elde edilen iskelet verileri birleştirilmiş ve birleştirilmiş eklem noktalarının koordinatlarının doğru tespiti için parçacık filtresi (particle filter) algoritması kullanılmıştır. Eklem noktalarının tespitinden sonra kullanıcıların iskelet

verileri oluşturulmaktadır.

Bir dięer uygulamada kullanıcı hareketleri, kayıt altına alınan derinlik görüntü kareleri boyunca her karedeki iskelet verisinin birleşimi ile elde edilmektedir. Bu hareket verileri, her uzuv için belirlenen referans hareket verileriyle Fréchet uzaklığı yöntemi kullanılarak karşılaştırılmakta ve hareket benzerliklerinin ölçüsü tespit edilmektedir.

Bu çalışmada Kinect kullanılarak tüm vücudun hareketlerinin yakalanması ve işlenmesi üzerine odaklanılmıştır. Kinect ile yüz takibi bu çalışmanın kapsamı dışındadır.

2. KINECT KAMERA SİSTEMİ

Kinect, Microsoft tarafından geliştirilen derinlik ve hareket algılama özelliklerine sahip özel bir kamera sistemidir. İçerisinde kızılötesi projektör, RGB kamera, derinlik algılayıcı kamera ve mikrofonlar bulunmaktadır. Xbox 360 oyun konsolları için üretilen sürümü “Kinect for Xbox 360” şeklinde, ticari amaçlı kullanım için üretilen sürümü Kinect for Windows (Şekil 2.1) olarak isimlendirilmiş ve piyasaya sürülmüştür.



Şekil 2.1 Kinect For Windows

2014 yılında Kinect' in ikinci modeli, sadece Xbox One oyun konsolu ile kullanılabilen “Kinect for Xbox One” ve ticari amaçlı kullanım için geliştirilen “Kinect for Windows v2” olarak piyasaya sürülmüştür. Daha sonra “Kinect for Windows v2” nin üretimi durdurulmuş, Xbox One için üretilen Kinect' i (USB üzerinden) bilgisayara bağlayan adaptör üretilerek, ticari amaçlı kullanımına izin verilmiştir.

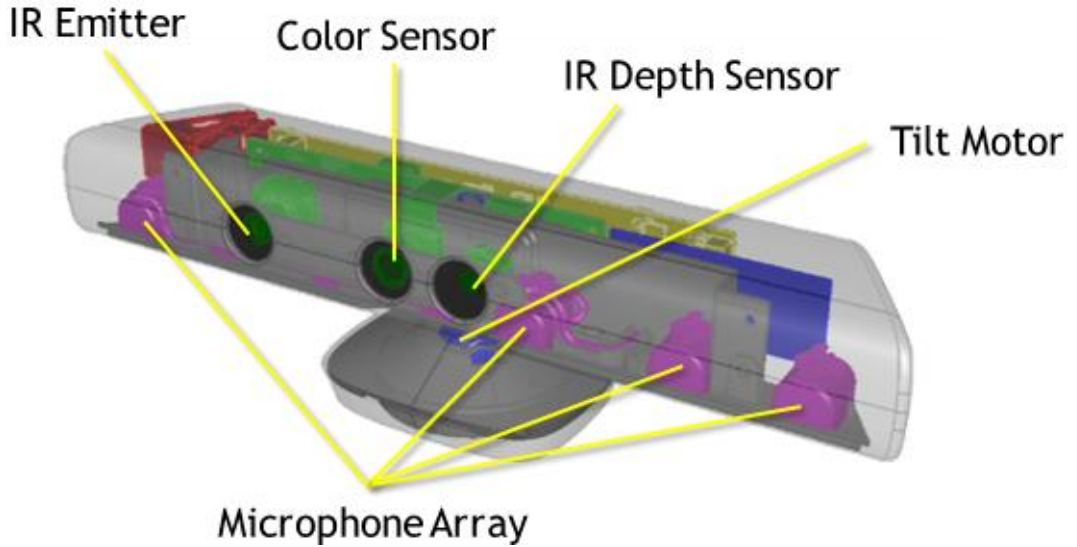
2.1. Kinect'in Teknik Özellikleri

Kinect'in (Kinect for Windows ve Kinect for Xbox 360) optik bileşenleri Şekil 2.2'de gösterildiği gibi kızılötesi projektör, IR kamera ve renkli (RGB) kameradan oluşmaktadır.

- 1) **Renkli (RGB) Kamera:** Görüntünün temel üç renk (kırmızı-yeşil-mavi) bileşenini vermektedir. 30 Hz' te çalışmakta ve (kanal başına 8 bit) 640 × 480 piksel çözünürlükte görüntü oluşturmaktadır. Saniyede 10 görüntü karesinde çalıştırıldığında 1280×960 piksel çözünürlüğünde görüntü elde

edilebilmektedir.

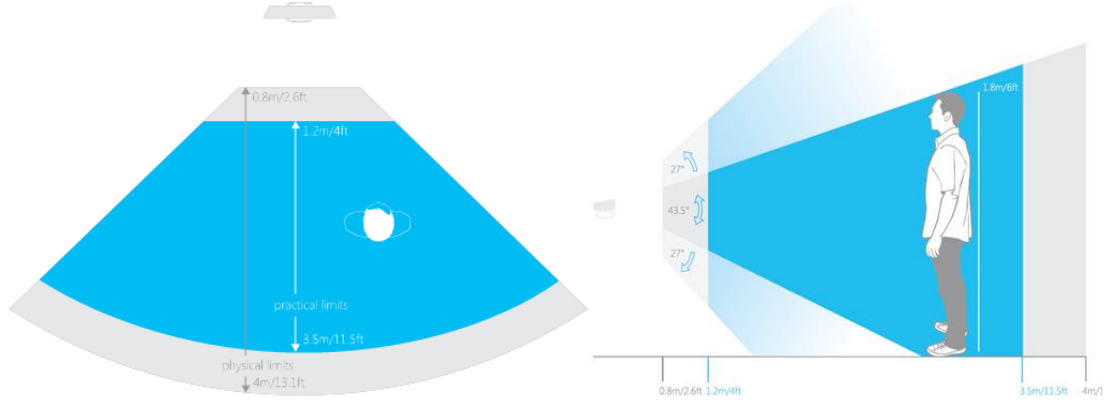
- 2) **3-B Derinlik Algılayıcı:** IR kamera ve kızılötesi projektör derinlik algılama sistemini oluşturmaktadır. Bu iki bileşenle sahnedeki bir nesnenin kameraya olan uzaklığını içeren derinlik haritası çıkarılabilmektedir. Derinlik algılayıcı; Şekil 2.3’de gösterildiği gibi kameradan yakın modda 0,4 m – 3 m, normal modda 0,8 m – 4 m uzaklık aralığında sağlıklı görüntü verisi sağlayabilmekte ve bu veriyi 30 FPS’ de 640×480 piksel çözünürlüğünde görüntüleyebilmektedir. 4m’den daha uzaktaki nesnelere de takip edilebilmekte, ancak kullanıcı iskeleti takibinde veri üzerindeki gürültü artmakta ve elde edilen iskelet verisi bozulmaktadır. Görüntüleme alanı yatayda 57,5°, dikeyde 43,5° görüş açıları ile tanımlanmaktadır.
- 3) **Motorlu Tilt:** Kinect’ in aşağı ve yukarı yönde 27° oynamasına imkân sağlayan bileşendir.



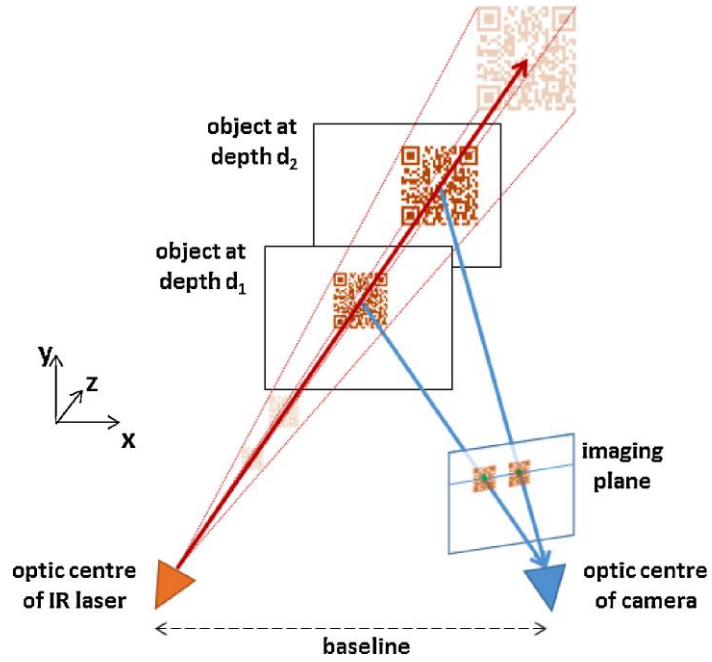
Şekil 2.2 Kinect bileşenlerinin kamera üzerinde gösterimi [15]

Kızılötesi projektörler kameranın görüş alanına (field of view) kızılötesi ışınlar (speckle dot pattern) yaymakta ve IR kamera ise yansıyan kızılötesi ışınları yakalamaktadır. (Şekil 2.4) Kızılötesi projektör ve kamera arasındaki geometrik ilişki offline kalibrasyon ile elde edilmektedir. Kızılötesi projektörün yaydığı ışık renkli kamerada görünmemesine karşın kızılötesi kamera ile tespit edilebilmektedir. Yayılan noktaların her yerel örüntüsü eşsiz nitelikte olduğundan görüntüdeki gözlenen yerel nokta örüntüleri ile kalibre edilmiş projektör nokta örüntülerini karşılaştırmak mümkündür. Bir noktanın derinliği nokta örüntüsünün göreceli sağ-sol

taşınmasından çıkarılabilir. Bu taşıma, nesnenin kamera-projektör düzlemine olan uzaklığına bağlı olarak değişmektedir [16].



Şekil 2.3 Kinect derinlik ölçüm aralığı [17]

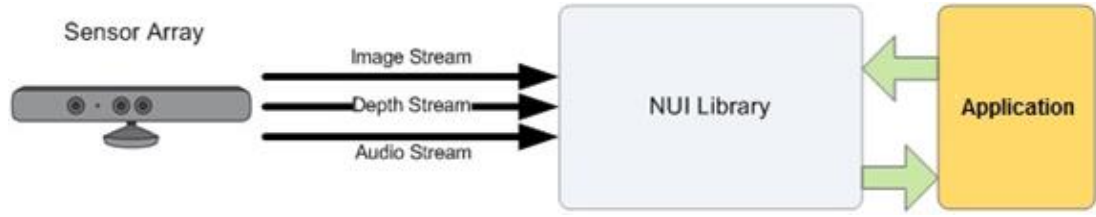


Şekil 2.4 Kinect' in derinlik ölçümü [16]

Kinect' in üzerinde mikrofondan oluşan ses algılama sistemi de yer almaktadır [18]. Bu tez kapsamında ses algılamaya ilişkin bir faaliyet yürütülmediğinden Kinect' in ses algılama detaylarına değinilmeyecektir.

2.2. Kinect Yazılım Bileşenleri

Kinect yazılımı, Kinect donanımını (ön tanımlı sınırlar kapsamında) kontrol eden ve sağladığı verilerin işlenmesini mümkün kılan yazılım geliştirme kütüphanesi olarak tanımlanmaktadır. Kinect uygulaması ise (Şekil 2.5) Kinect'in sağladığı verileri kullanan, genellikle grafik kullanıcı arabirimi içeren, başka yazılımlarla ilintili veya kendi başına çalışabilen kullanıcı tanımlı yazılım olarak ifade edilmektedir. Microsoft; C#, C++ ve Visual Basic dillerinde geliştirilmiş örnek uygulamalar sunmaktadır.

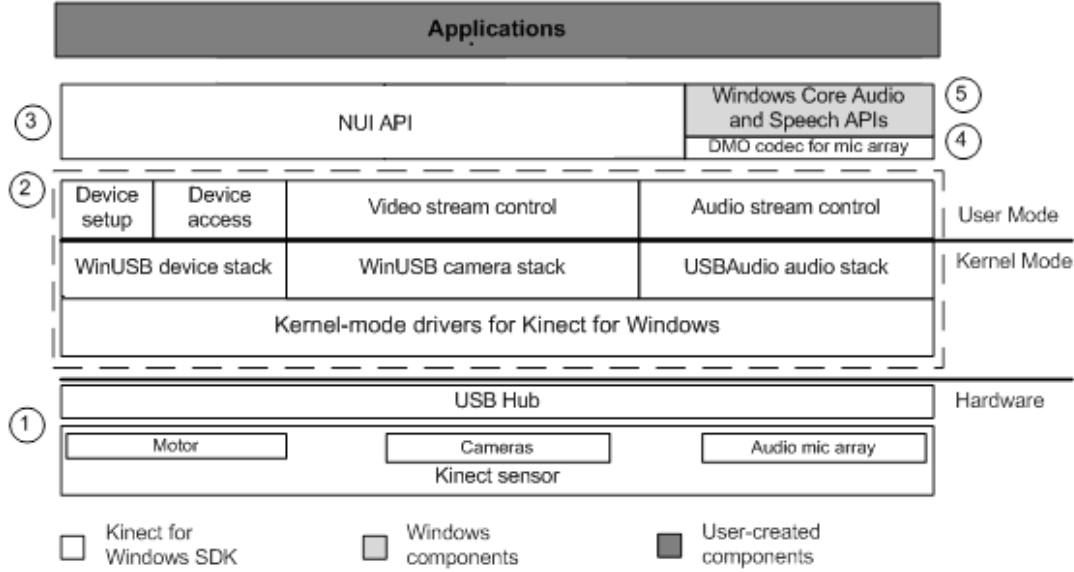


Şekil 2.5 Bir Kinect uygulamasının yazılım ve donanım etkileşimi [19]

OpenNI, Microsoft Kinect SDK ve OpenKinect (LibFreeNect) Kinect donanımını tanıyan yazılım araçlarıdır. Başlangıçta PrimeSense firması tarafından geliştirilen OpenNI, NITE adındaki Kinect gibi diğer derinlik algılayıcı kameralarla uyumlu hareket takip ara katman yazılımı ile birlikte çalışmaktadır. Nisan 2014'den itibaren açık kaynaklı halde Occipital tarafından yönetilen OpenNI'nin Mayıs 2014 itibariyle 2.2 sürümü kullanımdadır. OpenKinect açık kaynaklı bir proje olarak Kasım 2010'da başlatılmış ve Kinect'in Xbox oyun konsolu dışında ve bütün platformlarda kullanılmasını isteyen kişilerce geliştirilmiştir. Mayıs 2015 itibariyle GitHub üzerinde LibFreeNect adıyla erişilebilmektedir. Microsoft, OpenKinect ve OpenNI projelerinin yaygınlaşması üzerine, Kinect SDK adıyla kendi yazılım geliştirme aracını kullanıma açmıştır. OpenNI ve OpenKinect Linux ve MacOSX platformlarında da çalışabiliyorken, Kinect SDK .NET platformu üzerinde geliştirildiğinden yalnızca Windows işletim sistemi üzerinde çalışmaktadır. Kinect SDK 1.8 sürümü ve öncesi itibariyle Kinect for Windows ürününe özel tasarlanmış olup; Ekim 2014 tarihi itibariyle son sürüm olan 2.0 sürümü ise Kinect for Windows v2 olarak da bilinen ikinci modeli desteklemektedir. OpenNI projesini sunucularında barındıran Occipital firması kendi ürünleri "Structure Sensor" üzerine yoğunlaştığından, 2014 ve sonrası itibariyle güncel Kinect yazılım geliştirme araçları Microsoft yönetiminde yayınlanmaktadır.

Kinect for Windows SDK Kinect uygulaması geliřtirmek için gereken sürücü, referans API, teknik dokümantasyonu sağlamaktadır. Ayrıca çeřitli Kinect kullanım örnekleri (C# ve C++) kodlarıyla gösterilmektedir.

řekil 2.6’da tez çalışmasında da kullanılan Kinect for Windows SDK 1.8 yazılım mimarisi gösterilmektedir. Kullanıcı tarafından oluşturulan uygulamalar, NUI API ve Windows çekirdek ses/konuşma API üzerinden Kinect donanımıyla iletişime geçmektedir.



řekil 2.6 Kinect SDK Mimarisi [19]

2.3. Kinect’in İnsan Pozu ve İskeletinin Tespiti

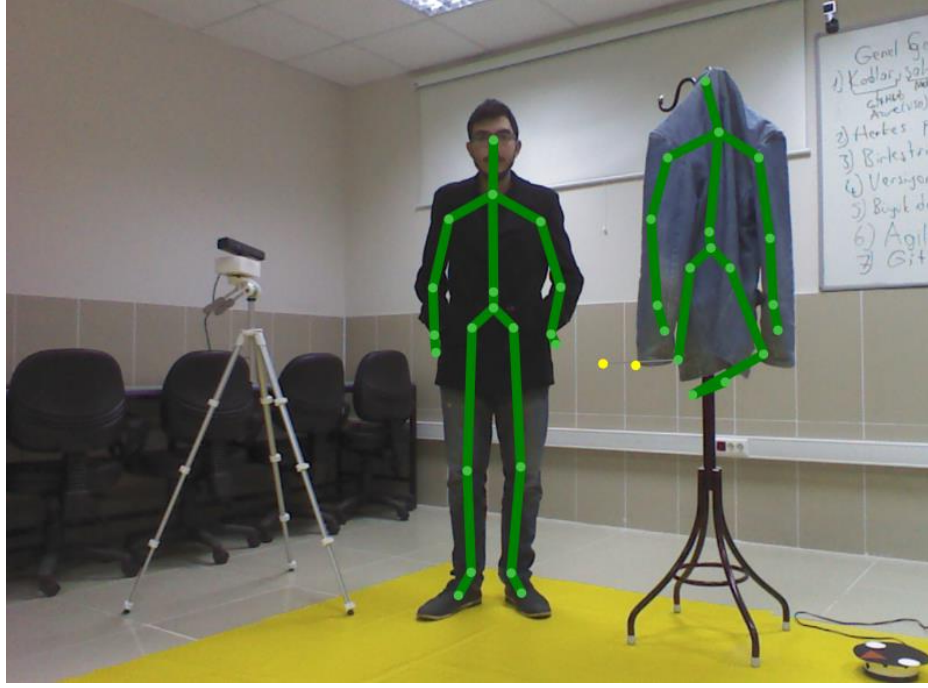
Kinect, görüş alanındaki ortamın ham derinlik görüntülerini vermektedir. Kinect’ in çeřitli bilgisayarlı görme uygulamaları için bu görüntünün işlenmesi gerekir [16]. Ham derinlik görüntüleri, hareketi yakalanacak aktörü tanımlayan piksellerin belirlenmesi ve bu piksellerin vücut bütünlüğünün sürekliliğini sağlamasında kullanılır. Derinlik verilerinden elde edilen bu pozun şekline göre kullanıcının iskeleti çıkarılır ve ilerleyen derinlik görüntü kareleri boyunca bu iskeletin takibi sağlanır. İskelet takibinde bütün vücut (oturma durumunda ise üst vücut) hareketi takip edilerek, eklemler oluşan yeni vücut pozuna göre konumlandırılır. Her bir eklem 3B koordinat verilerine sahiptir ve bir vücut bölgesini temsil eder. Bu sayede kullanıcı uzuvlarının Kinect ekranı uzayında nerede olduğu (görel koordinatlarla) anlaşılabilir.

İnsan vücudu farklı şekildeki duruşlara müsait, esnek bir yapıya sahiptir.

İnsan pozunun bilgisayarlar tarafından algılanmasında kameralar kullanılırken söz konusu esnek yapının yanı sıra; insan boy ve cüsselerindeki çeşitlilik, giyilen kıyafetlerin çeşitliliği problemi karmaşıklaştırmakta ve basit bir algoritma kullanılarak bu problemin genel çözümüne imkân vermemektedir.

Ham derinlik görüntülerinden insan pozisyonları çıkarımı için şu yöntemler önerilmektedir [20], [21].

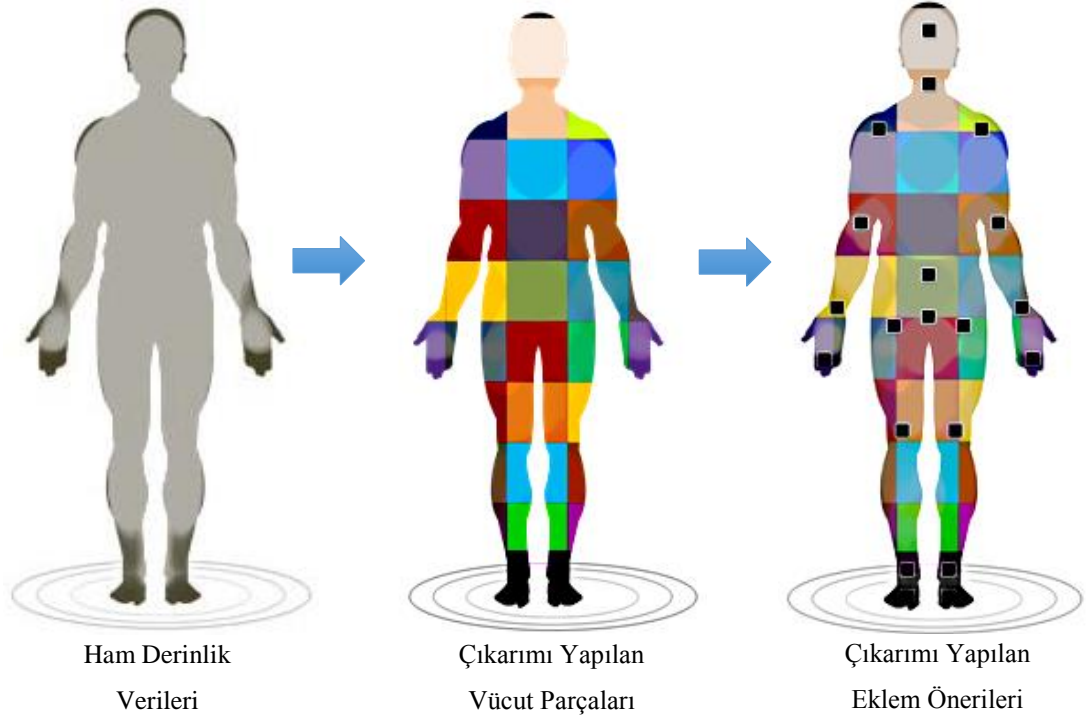
- 1) **Model Tabanlı Çıkarım:** Shotton ve ekibi tarafından geliştirilmiştir [20]. Şu anda Kinect'te kullanılan yöntem olup Kinect SDK ile kullanıcılara 20 eklemlili bir iskelet yapısını hazır olarak sunmaktadır. Çok sayıda derinlik görüntü verilerinin işlenerek çeşitli duruş modellerinin oluşturulması ve yeni pozisyonların bu duruş örnek eğitim verileri ile karşılaştırılmasına dayanır. Modeller oluşturulurken farklı boylar, cüsseler, kıyafetler, saç tipleri vb. etmenler hesaba katılarak çeşitli görünümüne göre eğitim verileri elde edilir. Eğitim verileri üzerinde vücut uzuvlarının konumları ve şekilleri belirlenerek kameradan elde edilen derinlik verisindeki silüetin parçalarının (bütün pikselleri eşleştirilerek) hangi vücut bölgesine denk geldiği tayin edilmektedir. Eğitim verileri sayesinde Kinect insan vücudunu, ham derinlik verileri üzerindeki diğer nesnelere görüntülerinden ayırt edebilmektedir. Ancak şekil olarak insana benzeyen çeşitli nesnelere (balmumu manken, üzerine palto asılmış portmanto vb.) kullanıcı ile birlikte aynı görüntü karesinde yer alınca Şekil 2.7'de görüldüğü üzere Kinect birden fazla kullanıcı olduğu yanlışlığına düşebilmektedir.



Şekil 2.7 Portmantoya asılı ceketin neden olduğu ikinci kullanıcı iskeleti

Kinect, düğümleri vücut kısımlarını birbirinden ayıracak şekilde etiketlenmiş birbirinden farklı karakter modelleri olan karar ağaçlarını kullanarak belirli vücut parçasına ait olabilecek pikselleri belirler. Bu şekilde kameradan elde edilen ham derinlik verilerindeki her pikselin karar ağacındaki bir düğümlerle eşleşmesiyle vücut parçaları oluşturularak vücut uzuvları belirlenir.

Belirlenen vücut uzuvlarına eklem noktaları atanır (Şekil 2.8) ve takip eden derinlik görüntü kareleri boyunca bu eklemler güncellenerek bütün vücut takibi yapılır. Eklem noktalarının Kinect koordinat sistemi üzerindeki koordinat değerlerini bulmak için, vücut silüeti görüntüsü bir küp üzerine Şekil 2.9'deki gibi yerleştirilerek eklem noktalarının 3B görünümü hesaplanır.



Şekil 2.8 Model tabanlı çıkarımla Kinect eklem noktalarını belirleme süreci [18]



Şekil 2.9 3B görünüm [18]

2) **Modelden Bağımsız Çıkarım:** Stommel, Beetz, ve Xu tarafından önerilen bu yöntem herhangi bir kinematik veya yüzey örüt (surface mesh) modeline ihtiyaç duymadığı için modelden bağımsız olarak tanımlanmaktadır. Bu yöntemde model tabanlı yaklaşımın sınırladığı kullanım kapsamını genişletmek maksadıyla ana noktaların mekân-zamansal bölütlenmesi

önerilmektedir. Bu çıkarım sürecinin ana adımları şu şekilde sıralanmaktadır [21].

1) Başlangıçta farklı filtreler kullanılarak Kinect' in sağladığı ortam derinlik haritası üzerindeki kişiler tespit edilir ve arka plan ayrımı yapılır.

2) Tespit edilen kişi silüetleri üzerinde Grassfire dönüşümü algoritmasının probleme uygun geliştirilmiş hali kullanılarak (topolojik) iskeletleştirme yapılır.

3) Tespit edilen kişilerin mekân zamansal bölütleme ile yörüngeleri elde edilir. Bu bölütleme, daha genel (model kısıtlaması olmaksızın) ve düşük işlem maliyetli bir yöntem olarak kullanılmaktadır. Bu yöntem eğitim verisine ihtiyaç duyulmadan ön tanımlı kurallarla görünüm tabanlı çalışmaktadır.

4) Kişilerin duruşunu kodlamak (encoding) için duruş benzerliğini koruyan ve duruş görünümleri uzayının örneklenmesine imkân veren görünüm tabanlı özellik tanımlayıcılar hesaplanır. Bu şekilde insan-bilgisayar etkileşimi uygulamalarında durum farkındalığı artırılmış olur.

5) Özellik uzayı Hilbert eğrisi boyunca hizalanarak ve lineerleştirilerek görselleştirilir.

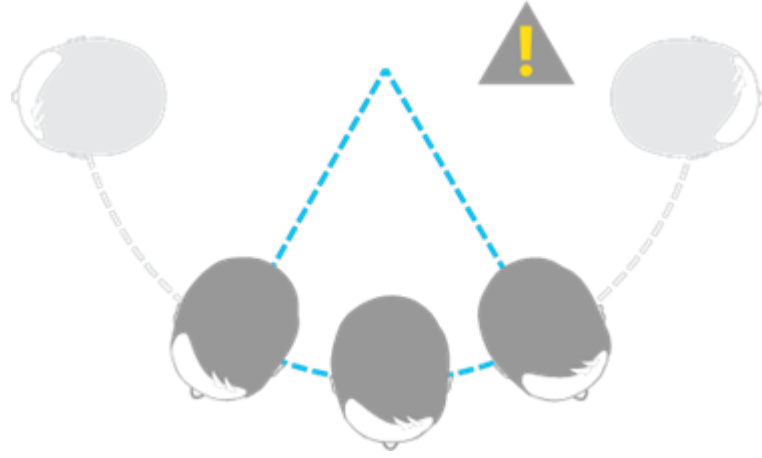
Bu iki yöntem arasındaki farklar Çizelge 2.1'de gösterilmektedir[21].

Çizelge 2.1 Modelden bağımsız ve model tabanlı çıkarımların karşılaştırılması

Kriter	Modelden Bağımsız	Model Tabanlı
Farkedilen Şekiller	Verilen boyuttaki ön plan nesnelere	İnsanlar ve bütün vücut uzuvları
Nesne Sayısı	Eşleşen bütün nesnelere dâhil eder	Genelde 1 veya 2 olarak kullanıcı tarafından ön tanımlıdır
Çıktı	Tespit edilen nesnelere poz/şekil seçici tanımlayıcıları	Eklem açıları
Başlama Zamanı	Arka plan bir kez ayarlanıncaya kadar	Görüntü karesi tabanlı
Eğitim Seti	Küçük video dizisi kümeleri	Etiketlenmiş büyük görüntü kümeleri
Eğitim	Nesne boyutunu elle ayarlayarak	Küme bilgisayar kullanarak makine öğrenmesi ile
Diğer duruşlara uygulanabilirlik	Yardımcı/televizyon amaçlı cihaz veya çanta taşıyan insanlar, hayvanlar, robotlar	Sadece insan vücudu
Gürültü	Benzer gürültülü pozlar ilişkilendirilebilir.	Mümkün olduğunca bastırılır.

Kullanıcıların Kinect tarafından algılanabilmeleri ve iskeletlerinin oluşturulmaları için kafa ve üst gövdelerinin kameranın görüş alanı (güvenilir tespit aralığı) içinde olması yeterli olup, özel bir poz vermeleri veya kalibrasyonu gerekmemektedir.

Kinect iskelet takibi Şekil 2.10'da gösterildiği gibi, kullanıcıların ayakta veya oturur vaziyette ve yüzlerinin Kinect'e dönük haldeki vücut pozisyonlarına göre optimize edilmiştir. Yan pozisyonlarda kamera kullanıcıların bazı uzuvlarını tespit edemeyebilir. Microsoft, geliştiricilerin bu durumu göz önünde tutarak uygulama geliştirmesini önermektedir [17].

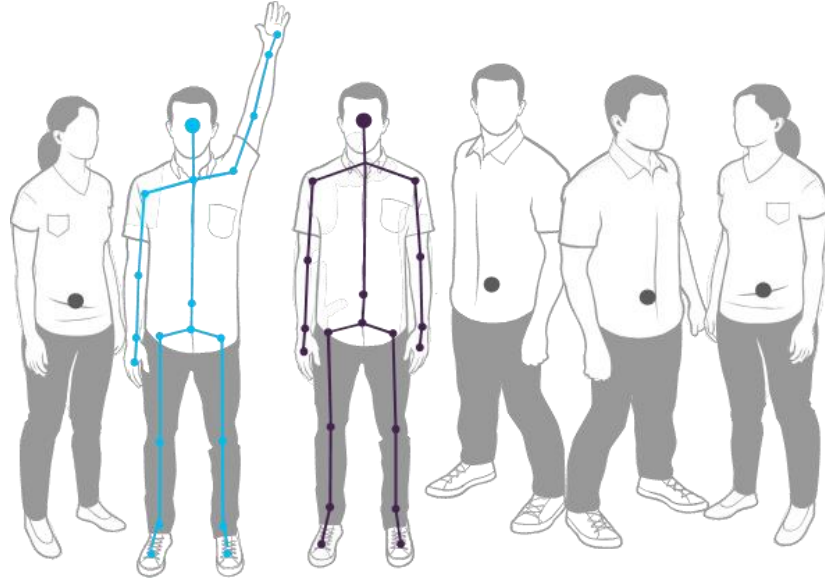


Şekil 2.10 İskelet takibi için Kinect' e yönelmek gerekir.

Kinect SDK v1.8 ile kameranın görüş alanındaki 6 kişi takip edilebilmektedir. Bunların ikisinin hareketleri, eklem noktaları 3B koordinatlar olarak tanımlanan iskelet yapısı üzerinden takip edilebilmektedir. Bu durum Şekil 2.11 ile gösterilmiştir. Diğer 4 kişinin iskeletleri çıkarılmayıp, sadece varlıklarını ifade eden bir kullanıcı kimlik bilgisi atanmaktadır. Kullanıcıların tanımlamaları kullanıcı kameranın görüş alanında yer aldığı süre boyunca geçerli olup, kullanıcı görüş alanından çıkıp tekrar girdiğinde farklı bir kimlik atanmaktadır.

Bir iskelet 20 eklem noktası ve bu noktaların hiyerarşik yapısı ile ifade edilmektedir. Bunun yanı sıra üst vücut bölgesini belirten 10 eklem noktasından (kollar, baş ve boyun) oluşan oturma modu adında ikinci bir iskelet takip yöntemi de bulunmaktadır. Vücut mimiklerini algılama için çeşitli hareketler ön tanımlı olarak yer almakta ve kullanıcılara özel mimik tanımlama imkânı vermektedir. El hareketleri kavrama (grip) ve itirme (push/press) şeklinde tanımlanmış olup parmak hareketlerini tanımlama düzeyinde bir hassasiyet söz konusu değildir.

Kullanıcıların yüzleri ve yüz ifadeleri tanınabilmekte ve gerçek zamanlı takip edilebilmektedir. Buna ilaveten Kinect SDK kullanıcının yüzünü 3B olarak modelleyip ikinci bir yüz görüntüsü oluşturabilmektedir.



Şekil 2.11 Kinect 2 kişinin iskeletini olmak üzere 6 kişiyi aynı anda takip edebilmektedir

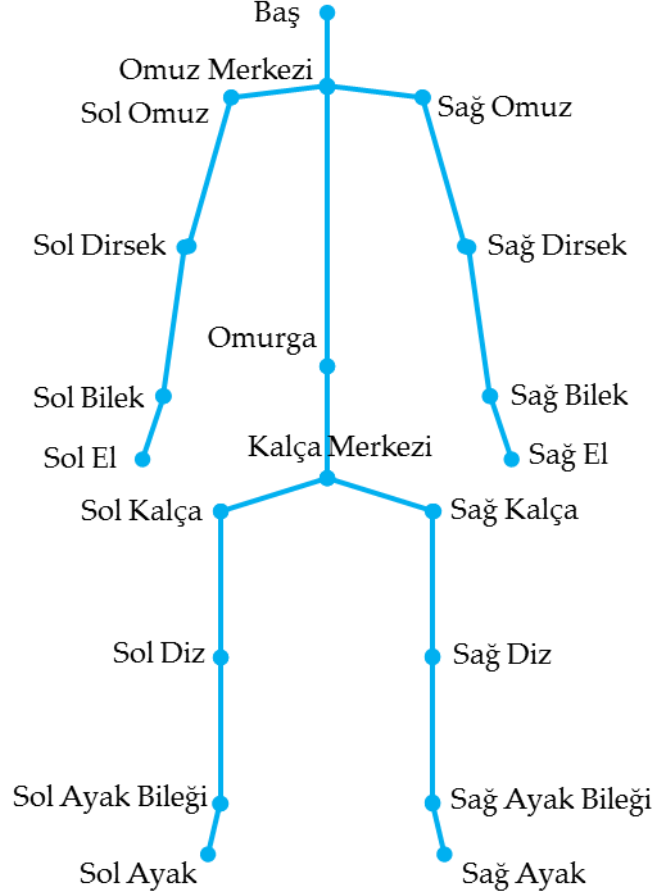
2.4. Kinect ile Hareket Yakalama

Kinect, kullanıcı iskeletini 20 eklem noktasından Şekil 2.12'te belirtilen hiyerarşiye göre oluşturmaktadır. Her bir eklem verisi ön tanımlı bir vücut bölgesine ait olup (Kinect koordinat uzayında yer alan) 3B koordinatlar içermekte ve zaman serisi şeklinde gösterilmektedir. Kullanıcı görüntü alanından çıkmadığı veya silüeti bozulmadığı sürece eklem noktaları aynı vücut bölgesini göstermeye devam eder. İnsan uzuvlarını temsil eden bu eklem verileri bir bütünlük (Kinect iskelet) içerisinde zamanla (yakalanan derinlik görüntü kareleri) güncellenmektedir.

Kinect ile elde edilen hareket yakalama verilerinin bir grafik uygulamasındaki 3B modele giydirilebilmesi için eklem noktalarının 3B koordinat bilgisine ek olarak eklemlerin yönelim bilgisine (joint orientation) de ihtiyaç duyulmaktadır.

Kinect for Windows SDK eklem yönelimi bilgisini iki biçimde sağlamaktadır [22]:

1. Bir eklem yakınındaki diğer eklemlerle ilişkisini ifade eden kemik üzerinde tanımlı (hiyerarşik olarak görel) dönme açıları
2. Kinect kamera koordinatları cinsinden mutlak yönelim bilgisi



Şekil 2.12 Kinect iskeleti eklem noktalarının isimlendirilmesi

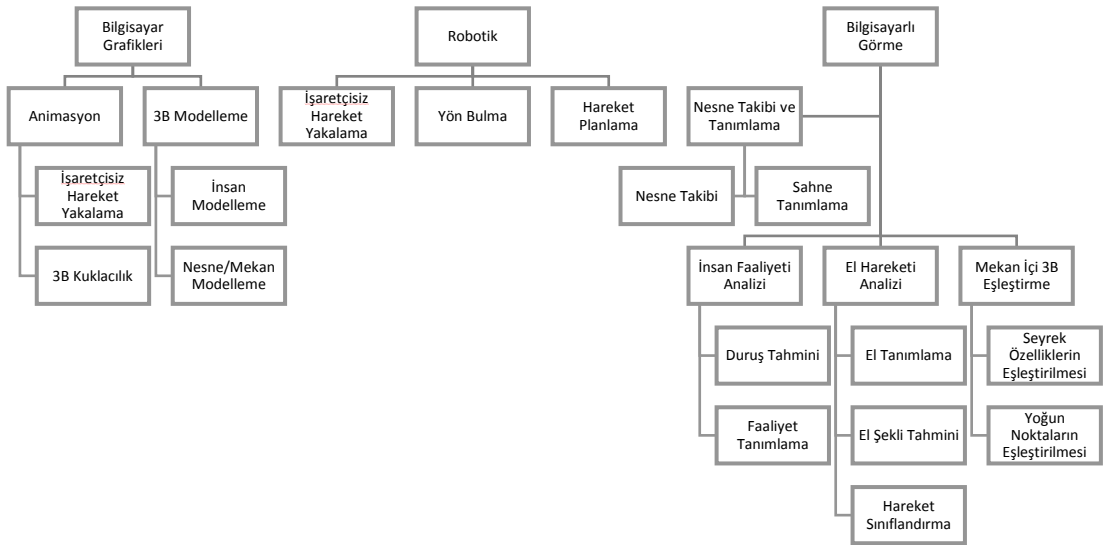
Kinect ile hareket yakalamak için kullanıcı iskelet verisinden yakalanan hareketi tanımlayan eklem noktaları belirlenerek bu noktaların hareket boyunca yakalanan derinlik görüntü karelerindeki 3B koordinatları zaman serisi üzerinde sıralanır. Böylece hareketi tanımlayan eklem noktalarının hareket yörüngesi elde edilir.

Kinect' in hareket yakalama başarımı işaretçi tabanlı sistemlerle mukayese edilmiş ve belli durumlar için muadil olarak kullanılabilceği gösterilmiştir [23].

Bonnechère, Sholukha, Moiseev, Rooze ve Van Sint Jan; Kinect kullanılarak yapılan işaretçisiz hareket yakalamada elde edilen verilerin anatomik olarak eksik ve hatalarının olduğunu tespit etmişlerdir. Canlı uzuvlar ve kadavra (yumuşak doku olmaksızın doğrudan kemik üzerine işaretçi yerleştirerek) üzerinde işaretçi tabanlı sistemler kullanılarak eklem hareketlerinin modellenmesiyle oluşturulan model tabanlı yaklaşımla ek veriler elde edilmiştir. Anatomik açıdan daha gerçekçi olan bu verilerin Kinect iskelet verisine eklenmesiyle Kinect'in hareket yakalama başarımı artırılmıştır [24].

2.5 Kinect Araştırma Konuları ve Akademik Çalışmalar

Microsoft'un Kinect for Xbox 360 ürününü 2010' da satışa çıkarmasıyla birlikte, yüksek çözünürlükte derinlik algılamayı muadil ürünlere göre düşük bir fiyatta sağlayan bu sensör, bilgisayarlı görme alanındaki araştırmacılar başta olmak üzere geliştirici ve araştırmacıların ilgisini çekmiştir.



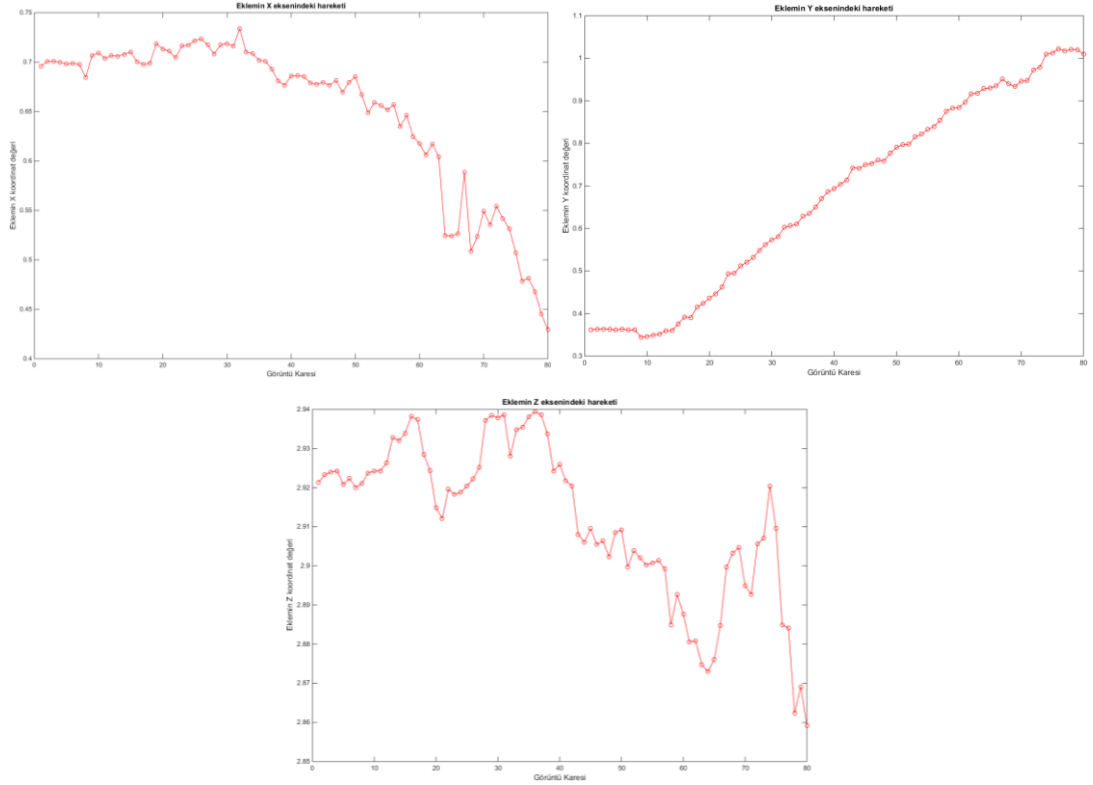
Şekil 2.13 Kinect araştırma konularına genel bakış

Han, Shao, Xu ve Shotton; Kinect araştırma konularıyla ilgili bir inceleme sunmuşlardır. [16]. Bu incelemedeki araştırma konuları genel başlıklarıyla Şekil 2.13' de gösterilmiştir. Kinect başta olmak üzere derinlik algılayıcı sistemler; görüntü işleme ve bilgisayarlı görme alanında güncel araştırma konuları arasında yer almaktadır.

Fossati, Gall, Grabner, Ren ve Konolige; ICCV 2011 içerisinde yer alan "IEEE Bilgisayarlı Görme için Tüketim Derinlik Kameraları" atölyesine sunulan bildirileri genişleterek kitap halinde yayınlamışlardır [25]. Bu kitapta Kinect üzerine yapılan 3B kayıt ve yeniden oluşturma, insan vücudu analizi ve RGB-D veri setleri konulu çalışmalar bulunmaktadır.

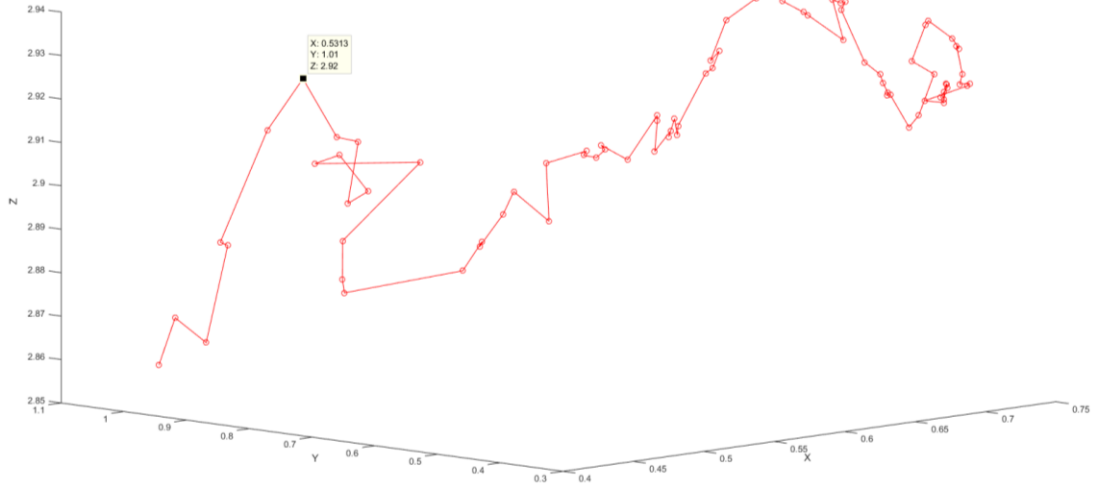
3. HAREKET YÖRÜNGELERİNİN BENZERLİK ÖLÇÜMÜ

Kinect ile yakalanan hareketlerin benzerliğinin ölçümü için öncelikle hareketlerin geometrik gösterimi belirlenmelidir. Optik hareket yakalama sistemleri ile yakalanan bir hareket kesikli (discrete) bir yapıdadır. Hareketi takip edilen işaretçi veya işaretçisiz sistemlerde eklem noktasının herhangi bir t anındaki konumu görüntü karesi tabanlı belirlendiğinden zaman değişimi Δt 1 görüntü karesidir. Yüksek frekanslı kameralarla saniye başına ne kadar daha çok görüntü karesi elde edilirse, mevcut kesikli yapı sürekli bir yapıya daha fazla yakınsayacaktır. Kinect'in saniyede 30 derinlik görüntü karesi verdiği göz önüne alındığında, 80 görüntü karesi boyunca takip edilen bir eklem hareketinin sürekliliğe yakınsama düzeyi Şekil 3.1' de görülmektedir.



Şekil 3.1 Örnek eklem koordinatlarının 80 görüntü karesi boyunca kesikli hareketi

Grafikteki eklem noktaları doğru parçaları ile birleştirildiğinde Şekil 3.2'de görüldüğü üzere 3B çokgensel eğri (polychain) biçiminde bir yörünge elde edilmektedir. Eğrinin başlangıç ve bitiş noktaları, yakalanan hareketin başlangıç ve bitiş anlarına tekabül ettiğinden, yakalanan bir hareketin geometrik gösteriminde kullanılabilir.



Şekil 3.2 80 noktadan oluşan çokgensel eğri yörüngesi

Bu tez çalışmasında hareketlerin benzerliği; yakalanan hareketi tanımlayan eklem noktalarının izlediği yörüngenin, referans olarak kabul edilen hareketin oluşturduğu yörünge ile bir benzerlik ölçütü kullanılarak karşılaştırılması şeklinde tanımlanmaktadır. Literatürdeki yaklaşımlardan farklı olarak, hareketler bir benzerlik ölçütü kullanılarak birbirleriyle kıyaslanmaktadır.

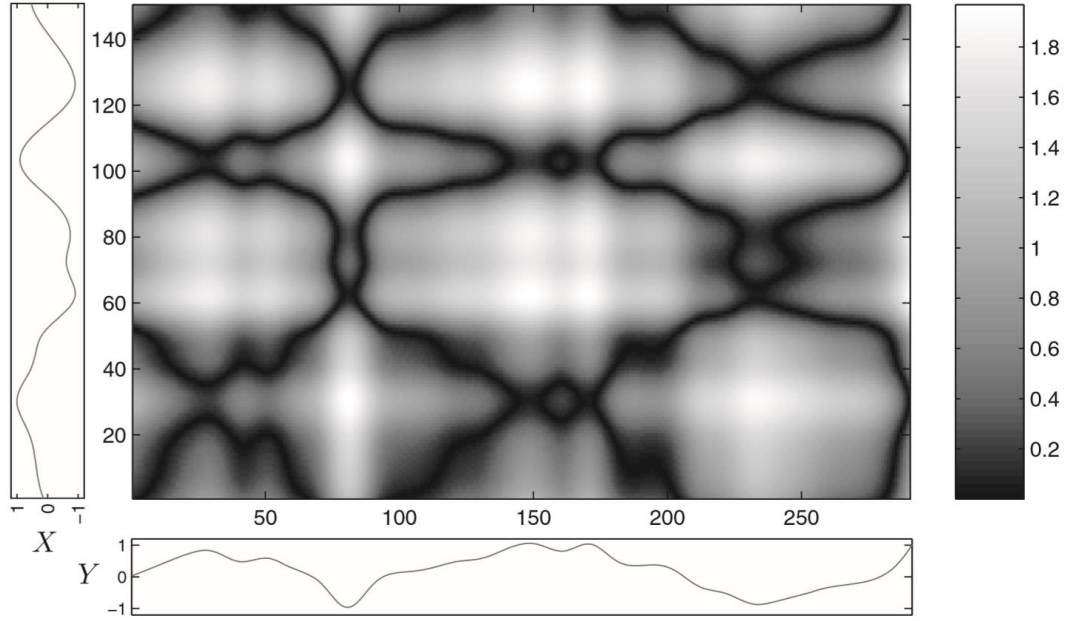
Eğri karşılaştırma; (hesapsal geometri ve görüntü işleme) şekil eşleştirme, imza benzerliği vb. pek çok alanda çalışılan ve uygulamaları bulunan bir araştırma konusudur. Uygulamadaki ihtiyaca ve eğri karşılaştırma probleminin karmaşıklığına göre literatürde çeşitli eğri karşılaştırma yöntemleri sunulmaktadır [26].

3.1 Dinamik Zaman Bükmesi

Dinamik zaman bükmesi (Dynamic Time Warping - DTW), konuşma tanıma için öne sürülen optimum dinamik programlama tabanlı zaman normalizasyonu algoritması olarak tanımlanmaktadır [27]. İlerleyen yıllarda bu algoritma zaman veya hız serilerini hizalama ve benzerliklerini ölçmek amacıyla da kullanılarak, lineer bir diziye dönüşebilen bütün verilerin analiz edilebildiği genel bir algoritmaya dönüşmüştür [28]. Dinamik zaman bükmesi ve çeşitlerinin sinyal işleme, veri madenciliği, bilgi erişimi, biyoinformatik, robotik, bilgisayar grafikleri ve kimya mühendisliği alanlarında uygulamaları bulunmaktadır [29].

$X = (x_1, x_2, \dots, x_n)$ ve $Y = (y_1, y_2, \dots, y_n)$ zamanda eş uzaklıktaki noktalarla örneklenmiş iki özellik serisi olmak üzere DTW ile bu serileri karşılaştırmak için bir

yerel maliyet ölçütüne ihtiyaç duyulmaktadır. Bu ölçüt değerinin küçük olması özelliklerin birbirine daha çok benzediği anlamına gelmektedir. Dizilerdeki her bir eleman ikilileri için bu ölçüt hesaplandığında Şekil 3.3’ te görüldüğü üzere bir maliyet matrisi elde edilmektedir. X ve Y için en küçük toplam maliyeti sağlayan bir hizalama, bu matriste bir “vadi” üzerinde yer almaktadır. Bu hizalama bükme yolu (warping path) olarak belirtilmektedir.



Şekil 3.3 X ve Y dizilerinin Manhattan uzaklığı yerel maliyet ölçütü olarak kullanıldığında oluşan maliyet matrisinde koyu renkler düşük maliyeti göstermektedir [29]

Matematiksel olarak

$$p = (p_1, \dots, p_L) \quad (3.1)$$

$$l \in [1:L] \text{ ve } p_l = (n_l, m_l) \in [1:N] \times [1:M]$$

şeklinde tanımlanan bir p bükme yolu şu şartları sağlamalıdır:

Sınır şartı: Bükme yolunun başlangıç ve bitiş noktaları köşe noktalarına denk gelmelidir.

$$p_1 = (1,1) \text{ ve } p_L = (N, M) \quad (3.2)$$

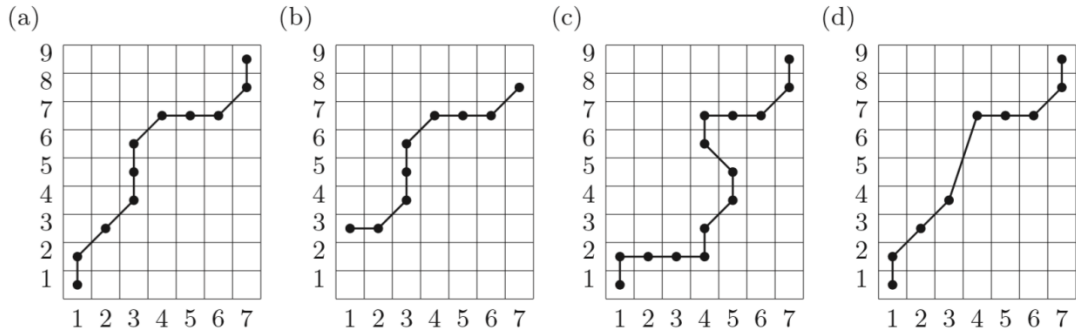
Monotonluk şartı: Bükme yolunda önceki değer, bir sonraki değerden küçük veya eşit olmalıdır.

$$n_1 \leq n_2 \leq \dots \leq n_L \text{ ve } m_1 \leq m_2 \leq \dots \leq m_L \quad (3.3)$$

Adım boyu şartı: Bükme yolu dizisindeki bir ikili, kendisinden bir önceki ikiliden farklı olmalıdır ve ikili değerleri arasındaki fark en çok 1 olabilir.

$$l \in [1:L - 1] \text{ ve } p_{l+1} - p_l \in \{(1,0), (0,1), (1,1)\} \quad (3.4)$$

Adım boyu şartı aynı zamanda monotonluk ve bir tür sürekliliği de ifade etmektedir. Monotonluk ve süreklilik, X ve Y serilerindeki hiçbir elemanın atlanmamasını ve hizalamada tekrarın önüne geçmeyi sağlamaktadır. Şekil 3.4' te bu şartlar göz önüne alınarak geçerli ve geçerli olmayan bükme yolları gösterilmektedir. Bu yollardan (a) geçerli bir bükme yolu olup; (b) sınır şartını, (c) monotonluk şartını, (d) ise adım boyu şartını ihlal etmektedir.



Şekil 3.4 9 elemanlı X ve 7 elemanlı Y dizileri için geçerli ve geçersiz bükme yolu çizimleri [29]

X ve Y dizileri arasındaki bir p bükme yolunun toplam maliyeti $c_p(X, Y)$, c yerel maliyet ölçütüne göre

$$c_p(X, Y) := \sum_{l=1}^L c(x_{n_l}, y_{m_l}) \quad (3.5)$$

denklemleriyle ifade edilmektedir. Bütün olası bükme yollarında en düşük toplam maliyetli olana optimal bükme yolu denmektedir. X ve Y dizileri arasındaki DTW uzaklığı bu optimal bükme yolunun toplam maliyetine eşittir.

N ve M elemana sahip X ve Y dizileri için optimal bükme yolunu olası bütün bükme yollarını deneyerek bulmaya çalışmak; çok zahmetli ve gereksiz işlemlere yol açabilecek bir yöntemdir, bu şekildeki bir kaba kuvvet algoritması eksponansiyel hesaplama karmaşıklığına yol açabilmektedir. Dinamik programlama kullanılarak $O(N.M)$ zamanda çalışan bir algoritma ile optimal bükme yolu bulunabilmektedir [29]. Bu algoritmanın girdi değeri olarak $N \times M$ boyutlarındaki birikimli maliyet matrisi D ;

$$D(n, m) := DTW(X(1:n), Y(1:m)) \quad (3.6)$$

değerlerinden oluşturulmaktadır. Burada $X(1:n) := (x_1, \dots, x_n), n \in [1:N]$ ve $Y(1:m) := (y_1, \dots, y_m), m \in [1:M]$ birer ön ek dizileridir. D birikimli maliyet matrisini kullanarak $p^* = (p_1, \dots, p_L)$ şeklindeki p^* optimal bükme yolunu bulan söz konusu algoritmanın adımları şu şekildedir:

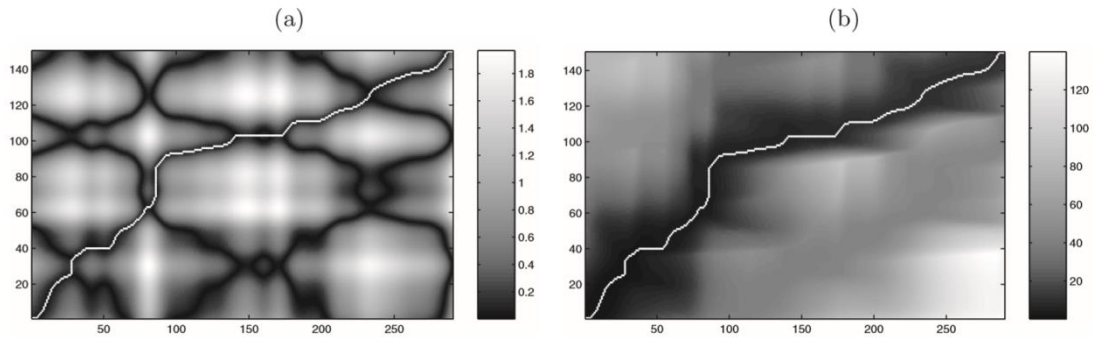
1) $p_L = (N, M)$ elemanından başlayarak optimal bükme yolunun elemanları tersten ele alınır.

2) $(n, m) = (1, 1)$ ise $l = 1$ olduğundan işlem sona erer. (özyineleme için taban durumu)

3) Diğer durumlarda p_{l-1} değeri şu şekilde bulunur:

$$p_{l-1} := \begin{cases} (1, m-1), & n = 1 \text{ ise} \\ (n-1, 1), & m = 1 \text{ ise} \\ \operatorname{argmin}\{D(n-1, m-1), D(n-1, m), D(n, m-1)\}, & \text{diğer} \end{cases} \quad (3.7)$$

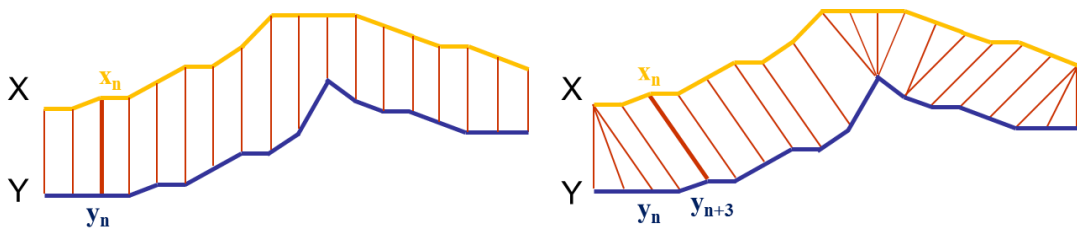
4) “argmin” sonucunda benzersiz bir değer bulunamazsa, sözlüksel biçimde en küçük ikili alınır.



Şekil 3.5 (a) Şekil 3.3’ teki maliyet matrisi ve (b) birikimli maliyet matrisi [29]

Şekil 3.3’ te ele alınan dizilerin maliyet matrisi ve hesaplanan birikimli maliyet matrisi Şekil 3.5’ te gösterilmektedir. Elde edilen optimal bükme yolu, her iki matris üzerindeki beyaz çizgilerle belirtilmektedir. Şekil 3.5(a)’ da görüldüğü üzere optimal bükme yolu, düşük maliyet değerleri üzerinden geçmektedir.

Öklid, Manhattan vb. uzaklıklarla X ve Y serileri üzerindeki aynı indisli noktalar arasında hizalama yapılması Şekil 3.6’da da görüldüğü üzere benzerlik ölçütünü tanımlamada yetersiz kalacaktır. Bunun yerine DTW kullanılarak esnek bir hizalama yapıldığında aralarında faz farkı olsa bile benzer şekillerin eşleşmesine olanak veren daha gerçekçi bir benzerlik ölçütü elde edilir.



Şekil 3.6 DTW ile daha gerçekçi benzerlik ölçümü

Dinamik zaman bükmesi algoritmasının C, C++, Python, Java ve MATLAB dillerinde açık kaynak kodlu uygulamaları bulunmaktadır.

Çelebi vd. uzuv hareketi algılama konusunda eklemlerin hareketin tanımlanmasında önem sırasına göre sınıflandırılmasına imkân veren ağırlıklı DTW algoritmasının kullanılmasını önermiş ve bu algoritma ile uzuv hareketlerinin Kinect' in komut olarak tanıyabileceği şablon hareketlere benzerliğini ölçerek hareket tanımlaması yapmışlardır [30].

3.2 Hausdorff Uzaklığı

Hausdorff uzaklığı, bir noktalar kümesinin diğer bir noktalar kümesindeki en yakın noktaya olabileceği en büyük uzaklık olarak tanımlanmakta ve iki nokta kümesinin birbirinden farklılığının ölçütü olarak da kullanılmaktadır. Nokta kümeleri birbirinden ayrı noktalardan oluşabileceği gibi geometrik şekiller de olabilir. İlk kez Felix Hausdorff tarafından yazılan *Grundzüge der Mengenlehre* (Küme teorisinin temelleri) kitabında tanımlandığından kendisinin soyadı ile isimlendirilmiştir.

A kümesinden B kümesine olan yönlü Hausdorff uzaklığı

$$h(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\} \quad (3.8)$$

şeklinde tanımlanmaktadır [31] [32]. Bu denklemde a ve b , A ve B kümelerine ait iki nokta; $d(a, b)$ ise bu noktalar arasındaki Öklid uzaklığı gibi herhangi bir metrik fonksiyonudur. Nokta kümeleri genel olarak ele alındığında $h(A, B) \neq h(B, A)$ olduğundan yönlü Hausdorff uzaklığı asimetrik bir fonksiyondur [32].

A kümesi ve B kümesi arasındaki Hausdorff uzaklığı ise

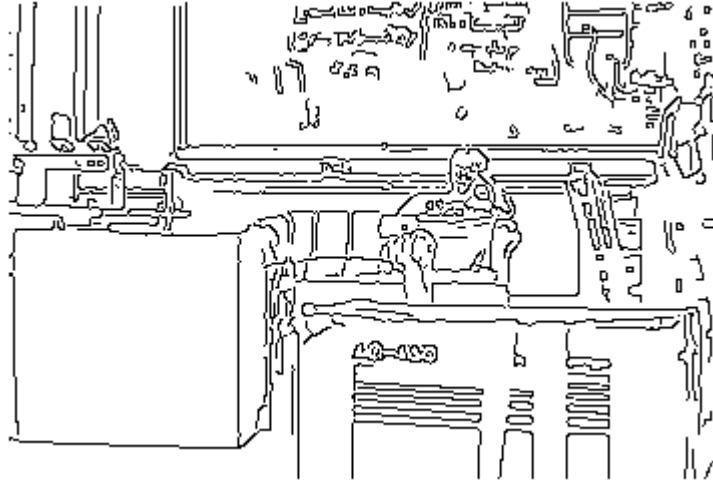
$$H(A, B) = \max \{ h(A, B), h(B, A) \} \quad (3.9)$$

şeklinde tanımlanmaktadır [31], [32]. A ve B 'nin birer geometrik şekil olduğu durumlarda bu fonksiyon şekilleri oluşturan bütün noktalar (sayılamayacak kadar çok) için geçerlidir. Aralarındaki Hausdorff uzaklığı d olan A şeklindeki bir nokta, B şeklindeki herhangi bir noktaya en fazla d kadar uzaklıktadır, bu da A ve B şekillerinin birbirleriyle eşleşmeyen bölümleri konusunda bir ölçüt vermektedir.

Atallah, n ve m köşe sayılarına sahip kesişmeyen dışbükey çokgenler arasındaki Hausdorff uzaklığını $O(n + m)$ zamanda hesaplayan bir algoritma önermiştir [33]. Bu algoritma sınırlı sayıdaki durumu kapsadığından, Hausdorff uzaklığının hesaplanmasında genel olarak kullanılacak bir algoritma niteliğinde değildir.

Taha ve Hanbury; manyetik rezonans görüntüleme'deki hacim bölütlemesi gibi büyük nokta kümeleriyle işlem yapıldığı durumlarda verimli çalışan ve herhangi bir kısıtlama yapılmaksızın genel amaçlı kullanıma elverişli, Hausdorff uzaklığını tam olarak neredeyse lineer zamanda ölçen bir algoritma önermişlerdir [32].

Huttenlocher ve Rucklidge, Hausdorff uzaklığı kullanılarak görüntü karşılaştırılması uygulaması yapmışlardır [34]. Şekil 3.7'teki görüntü içerisinde aranacak görüntü şablonu Şekil 3.8'de gösterilmektedir. Bu görüntüler Canny kenar tespiti yöntemiyle elde edilmiştir.



Şekil 3.7 İçinde arama yapılacak görüntü örneği [34]



Şekil 3.8 Aranacak görüntü şablonu [34]

Hausdorff uzaklığı kullanılarak yapılan eşleştirmede görüntü şablonunun, Şekil 3.7'te bulunan en uygun hali Şekil 3.9'da gösterilmektedir. Şekildeki yeşil çizgiler Canny kenar tespiti ile belirlenen kenar bölgelerini, kırmızı çizgiler geometrik dönüşümü yapılan görüntü şablonunun kenar bölgelerini ve sarı çizgiler de uyuşan bölgeleri göstermektedir.



Şekil 3.9 Görüntü şablonunun ana görüntüde bulunması [34]

3.3 Fréchet ve Kesikli Fréchet Uzaklığı

Fréchet uzaklığı eğriler arasındaki benzerlikte eğriler boyunca noktaların sıra ve konumlarını hesaba katan bir ölçüt olarak ifade edilmektedir. İlk kez Maurice Fréchet tarafından 1906'da tanımlandığından, kendisinin soyadı ile isimlendirilmiştir.

Bir metrik uzayda verilen iki eğri arasındaki Fréchet uzaklığı; köpeğini tasmayla gezdiren birinin, kendisi bir yolda (1. eğri) köpeği diğer yolda (2. eğri) olacak şekilde geriye dönmeksizin iki yolun tamamlanmasına imkân veren en kısa tasma uzunluğunu bulma problemi olarak ifade edilmektedir. S metrik uzayında A ve B tanımlanan iki eğri olmak üzere,

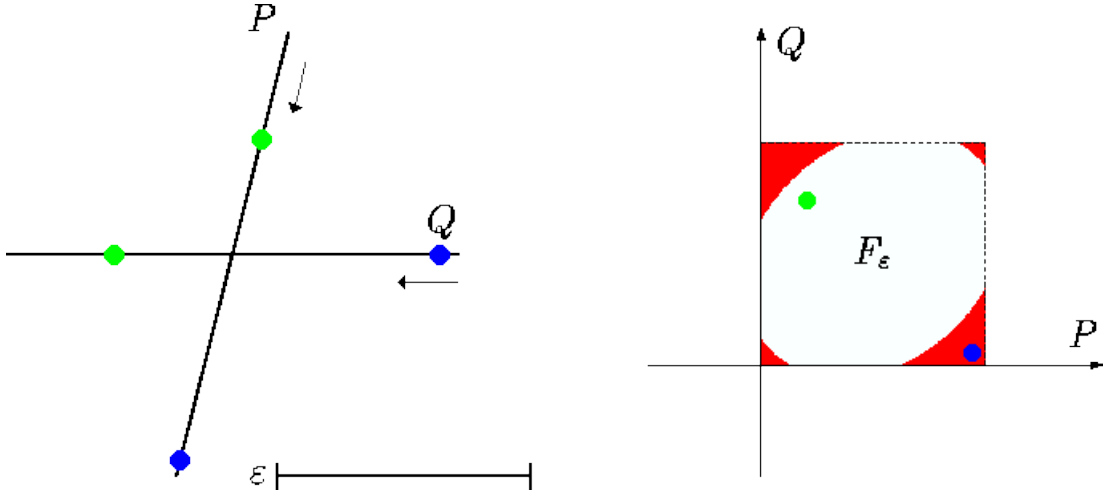
$$F(A, B) = \inf_{\alpha, \beta} \max_{t \in [0,1]} \{d(A(\alpha(t)), B(\beta(t)))\} \quad (3.10)$$

fonksiyonu ile A ve B arasındaki Fréchet uzaklığı hesaplanmaktadır. Burada α ve β , A ve B eğrilerini $t \in [0,1]$ ortak parametresini kullanarak tekrar parametrelendiren fonksiyonlar; d ise mesafe fonksiyonudur.

Alt ve Godau Fréchet uzaklığının hesaplanması için boş alan diyagramı kavramını öne sürmüşlerdir [35]. P ve Q doğru parçaları olmak üzere; $s \in P$ ve $t \in Q$ olmak üzere aralarındaki mesafe en fazla ε olan bütün (s, t) nokta ikilileri

$$F_\varepsilon = \{(s, t) \in [0,1]^2 \mid d(P(s), Q(t)) \leq \varepsilon\} \quad (3.11)$$

şeklinde tanımlanmaktadır. F_ε bir düzlem üzerinde Şekil 3.10'daki gibi gösterildiğinde birim kare içerisindeki boş alanı ifade etmektedir.

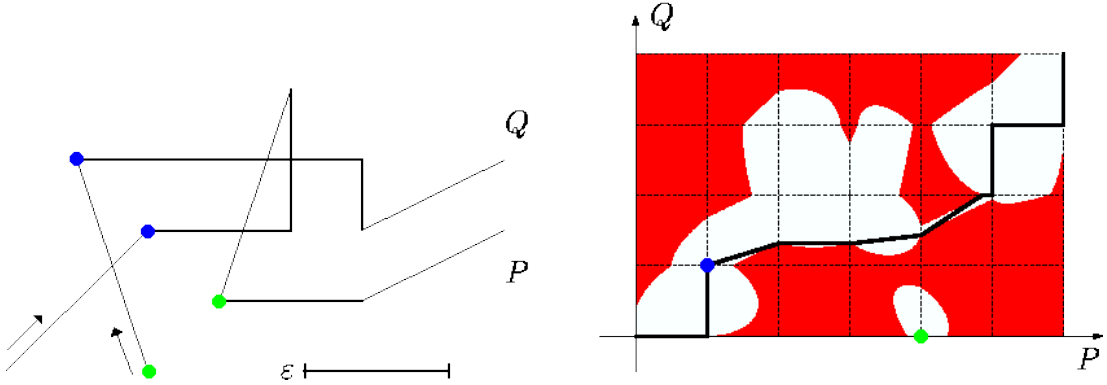


Şekil 3.10 İki doğru parçası üzerindeki “tasma” için boş alanın gösterimi [35]

P ve Q ; uzunlukları p ve q olan birer çokgensel eğri olarak yeniden düzenlediğinde aralarındaki mesafe en fazla ε olan bütün (s,t) nokta ikililerini gösteren denklem

$$F_\varepsilon = \{(s, t) \in [0, p] \times [0, q] \mid d(P(s), Q(t)) \leq \varepsilon\} \quad (3.12)$$

haline gelmektedir. Bu durumda F_ε bir düzlem üzerinde Şekil 3.11’deki gibi gösterildiğinde, verilen ε eşik değeri için iki eğri arasındaki boş alan diyagramı elde edilmiş olur.



Şekil 3.11 İki çokgensel eğri arasındaki boş alan diyagramı [35]

İki eğri arasındaki Fréchet uzaklığı; ancak ve ancak F_ε içerisinde $(0,0)$ noktasından (p,q) noktasına doğru her iki eksen de yalnızca bir yönde ilerlenebilen bir yol tanımlanabiliyorsa ε veya daha düşük bir değere sahiptir.

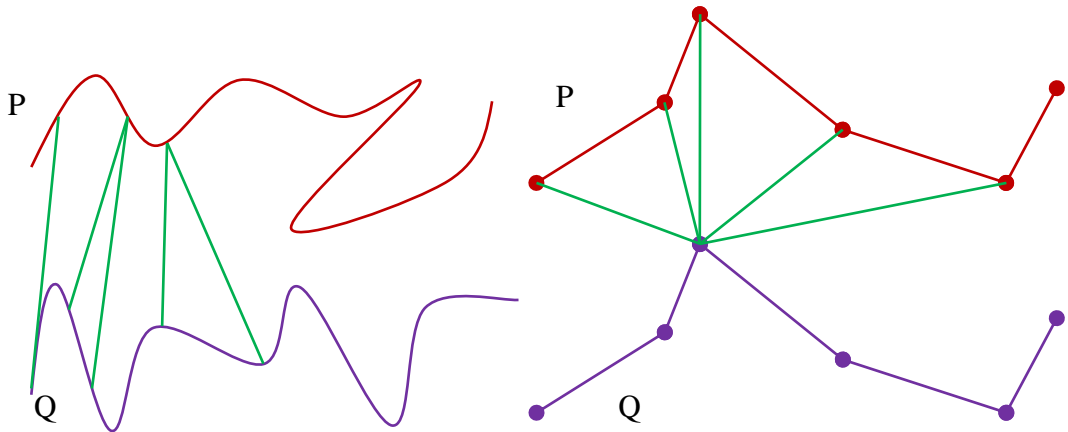
Zayıf Fréchet uzaklığı, uzaklığı belirten doğru parçasının uç noktalarının eğriler üzerinde serbestçe hareket edebildiği (tasmayı kısa tutmak için geriye doğru

harekete izin veren) bir Fréchet uzaklığı çeşidi olarak tanımlanmaktadır. Alt ve Godau bu uzaklığın hesaplanmasında çokgensel eğriler için bir algoritma önermektedir [35].

Jeodezik Fréchet uzaklığı, eğrilerin çok yüzlü zemin üzerinde olması veya eğriler arası engeller olduğu durumda uzaklığı belirten doğru parçasının jeodezik özellikte olduğu Fréchet uzaklığı türüdür. Cook ve Wenk bu uzaklığın hesaplanmasında basit bir çokgen içerisinde yer alan çokgensel eğriler için P zamanda çalışan bir algoritma önermektedir [36].

Alt ve Godau Fréchet uzaklığının hesaplama özellikleri konusunda, iki çokgensel eğri arasındaki tam Fréchet uzaklığını $O(pq \log^2 pq)$ zamanda hesaplayan bir algoritma tanımlamışlardır [35]. Burada p ve q çokgensel eğriler üzerindeki parça sayılarını ifade etmektedir.

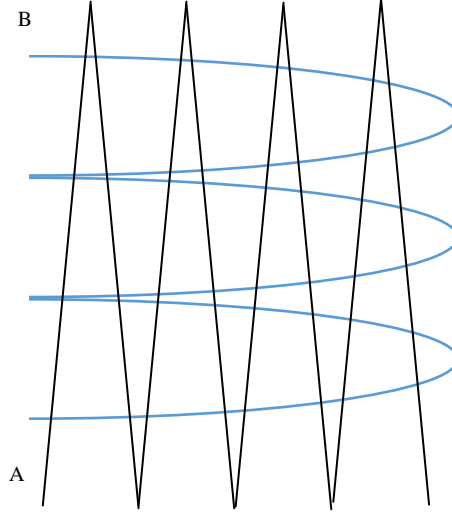
Eiter ve Mannila kesikli Fréchet uzaklığını (coupling uzaklığı), Fréchet metriğinin çokgensel eğriler için bir yaklaşımı olarak tanımlamaktadır [37]. Bu uzaklık hesaplanırken “tasmanın” ilk ve son noktalarının çokgensel eğriler üzerindeki köşe noktaları üzerinden seçilmesine müsaade etmektedir (Şekil 3.12). Bu kısıtlama sayesinde Eiter ve Mannila kesikli Fréchet uzaklığının bir dinamik programlama algoritması ile P zamanda hesaplanabileceğini göstermişlerdir. Uygulamada kullanılan MATLAB fonksiyonu bu yaklaşım ele alınarak kodlanmıştır.



Şekil 3.12 Fréchet ve kesikli Fréchet uzaklıkları

3.4 Probleme Uygun Yöntemin Belirlenmesi

Kinect eklem noktalarının yörüngeleri kesikli bir yapıdadır. Ayrıca bu yörüngeler farklı eğrilik yönlerinde ilerleyebilmekte ve görüntü kareleri farklı uzunluklarda olabilmektedir.



Şekil 3.13 Aralarındaki Hausdorff mesafesi düşük, Fréchet mesafesi yüksek olan A ve B eğrileri

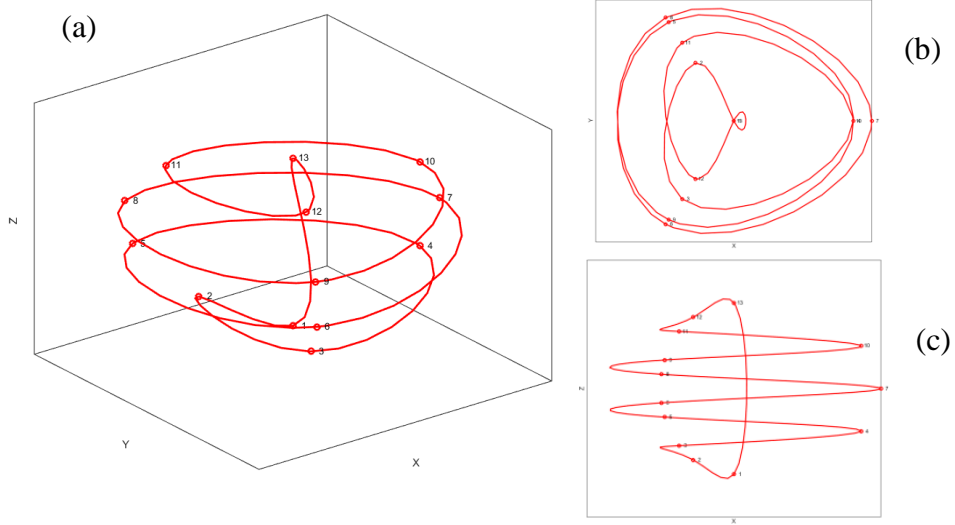
DTW yöntemi zaman serisi üzerinde geriye doğru harekete izin vermediğinden, hareket yörüngelerinin DTW algoritması için geçerli hale getirilmesi ek işlemler gerektirmektedir. Hausdorff uzaklığı şekil karşılaştırma probleminde düzlemsel eğrilerin benzerlik ölçütünü hesaplamak için önerilen diğer bir yöntemdir. Ancak Hausdorff uzaklığı düşük olmasına rağmen, eğrilerin birbirine benzemediği durumlar görülebilmektedir. Bu durum Şekil 3.13' de gösterilmiştir. Hausdorff uzaklığında yörüngelerin eğrilik yönleri dikkate alınmadığından, benzerlik ölçütü açısından Fréchet uzaklığı daha iyi sonuç vermektedir. DTW uygulamalarında yörüngelerin eşit uzunlukta olmaları gerekmemektedir. Bu durum Fréchet uzaklığı için de geçerlidir.

Bütün bu durumlar göz önünde tutulduğunda, tezde ele alınan hareket yörüngelerini karşılaştırma probleminde en uygun çözümün Fréchet uzaklığı algoritması olduğu kanaatine varılmıştır.

3.5 3B Eğrileri Karşılaştırmak

Yukarıda ele alınan eğri karşılaştırma yöntemleri düzlemdeki eğriler içindir. 3B eğrilerin karşılaştırılmasında eğrilerin üçüncü boyutunun da hesaba katılması gerekmektedir. Karşılaştırılması için geometrik dönüşüm (taşımaya, döndürme, ölçeklendirme vb.) uygulanmasına gerek olmayan 3B eğrilere bu yöntemleri uygulayabilmek için eğrilerin XY ve XZ düzlemlerine izdüşümlerinin (Şekil 3.14) alınarak boyut indirilmesi yapılmalıdır. Bu izdüşümler üzerinden karşılaştırmaların yapılması, geometrik dönüşümlerin dikkate alınmadığı eğrilere doğru sonucun bulunması için yeterlidir [38].

Yapılan uygulamada elde edilen yörüngeler, uygulama ortamının koordinat sistemine göre konumlanmaktadır. Karşılaştırılan bütün yörüngeler aynı koordinat sisteminde ve aynı oktantta yer aldığından karşılaştırma öncesi eğrilerin geometrik dönüşümlerle koordinat sistemleri arası taşınmasına veya döndürülmesine gerek yoktur.



Şekil 3.14 (a) 3B bir spline eğrisinin (b) XY izdüşümü ve (c) XZ izdüşümü

4. ÇOK SAYIDA KINECT İLE VERİ ELDESİ

Kinect nesnelerin sadece bir yönden görünümünü vermektedir. Bir eklemkin Kinect tarafından algılanamaması durumunda, Kinect eklem koordinatını çıkarımla belirlemeye çalışmaktadır. Bu durum eklem koordinatlarının yanlış tahminiyle sonuçlanmaktadır. Nesnelerin birbirini örtmesi (occlusion) veya daha detaylı yüzey verisi elde etme gibi ihtiyaçlardan dolayı, çok sayıda Kinect kamera kullanılarak veri elde etme yaklaşımı ortaya çıkmıştır.

Bu bölümde çoklu Kinect ortamının oluşturulması ve Kinect'lerden elde edilen iskelet verilerini birleştirmede parçacık filtresi algoritmasının kullanılması önerilmiştir. Çoklu Kinect sensörleri kullanılarak hareket yakalama üzerine yapılan önceki çalışmalar Kinect'lerin iskelet verilerini kullanma veya duruş tahmini için siluet veya şablon eşleştirme optimizasyonu olarak gruplandırılabilir.

Li vd., fizik tedavi gözetiminde iskelet verilerini birleştirmek için Kalman filtrelerinin kullanımını önermişlerdir [39]. Bu çalışmada çoklu Kinect sistemi bir ana Kinect ve ona bağımlı N adet Kinect şeklinde oluşturularak ana Kinect'in yerel Kartezyen koordinat sistemi dünya koordinat sistemi olarak tanımlanmıştır. Kinect'lerin derinlik sensörleri dama tahtası örüntüsü kullanmak yerine insan vücudundaki 3B eklem konumları kullanılarak kalibre edilmiştir. Kalibrasyon sürecinden sonra ölçüm hassasiyetini geliştirmek amacıyla iskelet verilerini girdi olarak alan ve birleşmiş iskelet verilerini çıktı olarak veren Kalman filtresi kullanılmıştır.

Baek ve Kim, dört Kinect sensöründen elde edilen iskelet verilerini birleştirmek ve farklı vücut ebatlarındaki duruşları karşılaştırmak için yöntemler sunmuşlardır [40]. Bu çalışmada eklem konumları girdisindeki gürültünün azaltılmasında Kalman filtresi tabanlı hareket filtreleme gerçek zamanda uygulanmıştır.

Kitsikidis vd. iskelet verilerini birleştirmek için eklem konum nokta bulutlarını kullanarak dönüştürülmüş çoklu iskelet verilerinin ana Kinect'e kalibrasyonunu önermişlerdir [41].

Asteriadis vd. eklem noktalarını tahmin etmek için enerji fonksiyonlarının kullanımını önermişlerdir. Ayrıca bulanık çıkarım sistemi vasıtasıyla belirli bir eklemkin gürültü tahmininin hesaplanması için bir yöntem sunmuşlardır [42]. Bilinmeyen hareketler için tanım kümesi tabanlı bilgi yapısının (knowledge

framework) belirtilmesi mümkün olmadığı için, her vücut parçasının tahmininde olası duruş, dışavurumculuk ve hareket geçmişini göz önünde tutan ve güven tabanlı mantığı temel alan yerel bir yaklaşım öne sürülmüştür. j eklemının beklenen 3B konumunun önceki konumlar, dışavurumculuk, vaziyet ve her Kinect'in derinlik verileriyle belirlendiği ileri sürülerek; j eklemının beklenen 3B konumunu hesaplamak için, E_j^k enerji fonksiyonlarının toplamının (üç Kinect' ten elde edilen) \mathbf{p} aday konumlarının bir kümesine maksimizasyonu şu şekilde tanımlanmıştır.

$$x_j = \arg \max_x \left(\sum_{k \in K} E_j^k(\mathbf{p}, \mathbf{x}) \right) \quad (4.1)$$

Burada K ; bütün Kinect sensörlerinin kümesi, $E_j^k(\mathbf{p}, \mathbf{x})$; aday konumların takip edilen eklemlere uzaklık çekirdeklerinin Mamdani bulanık girişim şemasının çıktısıyla çarpımının ağırlıklı kombinasyonunu belirtmektedir.

Yeung vd. çoklu Kinect'lerden alınan iskelet verilerini birleştirme problemini kısıtlı bir optimizasyon yapısı altında tanımlamışlardır [43]. Tanımladıkları bu optimizasyon yapısında iki tür kısıtlama belirtmişlerdir. Kemik uzunluklarının korunmasını ağır kısıtlama, tutarsız eklem konumları arasındaki dengeyi de hafif kısıtlama olarak ifade eden bir algoritma sunmuşlardır. Uygulamada iki Kinect'i birbirine dik yerleştirmişler, kullanıcıyı kolları ve bacakları açık bir duruşla Kinect'lere 45° bakacak şekilde konumlandırarak sistemi başlatmışlardır. Bu sayede her iki Kinect'in, kullanıcının bütün eklem noktalarını takip edebilmesini sağlamışlar ve güvenli kemik uzunluklarının kontrolü için bir taban oluşturmuşlardır. Daha sonra aşağıdaki denkleme göre eklem noktalarını yeniden konumlandırmışlardır.

$$\min_{s.t.} \sum_{i \in S_A} w_i^A \|\mathbf{p}_i^* - \mathbf{p}_i\|^2 + w_i^B \|\mathbf{p}_i^* - (\mathbf{R}\mathbf{q}_i + \mathbf{t})\|^2 \quad (4.2)$$

$$\sum_{(i,j) \in S_A} (\|\mathbf{p}_i^* - \mathbf{p}_j^*\| - I_{i,j})^2 = 0$$

Burada (\mathbf{p}_i^*) 'ye optimize edilecek olan) \mathbf{p}_i , ilk Kinect'in iskeleti S_A 'daki bir eklem konumunu; \mathbf{q}_i , ikinci Kinect' in iskeleti S_B 'deki bir eklem konumunu; \mathbf{R} bir döndürme matrisini ve \mathbf{t} 'de iki Kinect'in koordinat sistemleri arasındaki geometrik dönüşümler için bir taşıma vektörünü göstermektedir. w_i^A ve w_i^B ise \mathbf{p}_i ve \mathbf{q}_i 'nin güvenilirliğine göre ayarlanan ağırlık değerleridir.

Williamson vd. çoklu Kinect kullanan ve kullanıcılar 360 derece döndüklerinde bile hareket yakalayabilen bir asker eğitim sistemi önermişlerdir [44]. Bu çalışmada yönelimsiz ve yönelim tabanlı birleştirme sistemleri öne sürülerek performans ve kararlılık açısından kıyaslanmıştır.

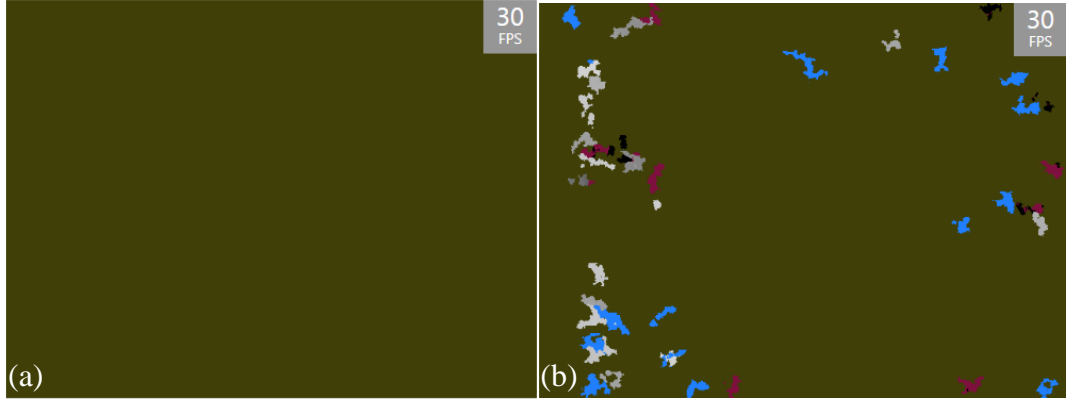
Berger vd. işaretçisiz hareket yakalama için dört Kinect' in aynı zamanlı kullanımını incelemişler ve derinlik verilerinden çıkarılan silüetleri birleştirerek hareket yakaladıklarını belirtmişlerdir [3]. Elde edilen sonuçları doğrulamak için RGB verileri kullanılmıştır. Ayrıca bu çalışmada çoklu derinlik ve RGB sensörlerini eşzamanlı kalibre etmek için yöntemler önerilmiş ve değişik malzemeler içeren bir hareket yakalama düzeneğindeki değişken sayıda Kinect'in girişim hataları araştırılmıştır. Doğrudan derinlik verileri üzerinde çalışma yapıldığından dolayı girişimden kaynaklanan hataları azaltmak için dört Kinect'e kadar zaman çoklayıcılı bir donanım çözümü öne sürülmüştür.

Zhang vd. yüksek boyutlu olarak tanımlanan duruşları takip etmek ve bir nokta bulutunda birleştirmek için parçacık filtresi ve bölüntü örnekleme (partition sampling) tekniklerinin bir uygulamasını ele almışlardır [45].

4.1 Çoklu Kinect Ortamının Hazırlanması

Çok sayıda Kinect kamerasının eş zamanlı kullanımında donanım tabanlı kızılötesi lazerlerin girişimi ve USB veri yolu kullanımı sorunlarıyla karşılaşmaktadır.

1) **Kızılötesi Lazerlerin Girişimi:** Kinect' in derinlik algılama mekanizması kızılötesi ışınların ortama gönderilmesi ve yansıyan ışınlardan derinlik haritası çıkarılması şeklindedir. Birden fazla kameradan gönderilen kızılötesi ışın demetleri dalga girişimi sonucunda birbirlerini sönmlediklerinden ortamın derinlik haritasında derinliği belirsiz bölümler oluşmaktadır. Şekil 4.1(a)'da bütün Kinect kameralarının kızılötesi projektörleri kapalıyken derinlik görüntüsü algılanamamaktayken, Şekil 4.1(b)'de diğer Kinect kameralarının kızılötesi projektörleri açılmakta ve söz konusu Kinect kendi kızılötesi projektörü kapalı olsa dahi diğer Kinect'lerin yaydığı ışın demetlerini algılayarak bir görüntü oluşturmaktadır.



Şekil 4.1 Kinect IR kamerası, diğer Kinect'lerin yaydığı ışın demetlerini algılar



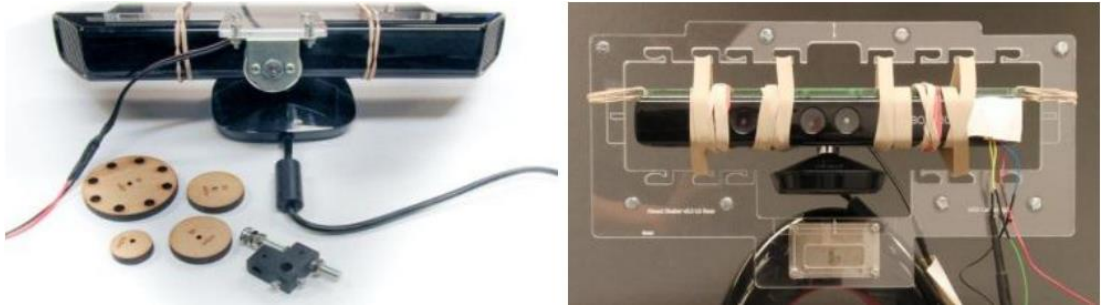
Şekil 4.2 Tek Kinect ile elde edilen derinlik görüntüsüne (üstte) ikinci Kinect' in neden olduğu ışın girişiminin etkisi (altta)

Şekil 4.2'de ise birinci Kinect'in oluşturduğu ortamın derinlik görüntüsüne ikinci Kinect' in olumsuz etkileri görülebilmektedir.

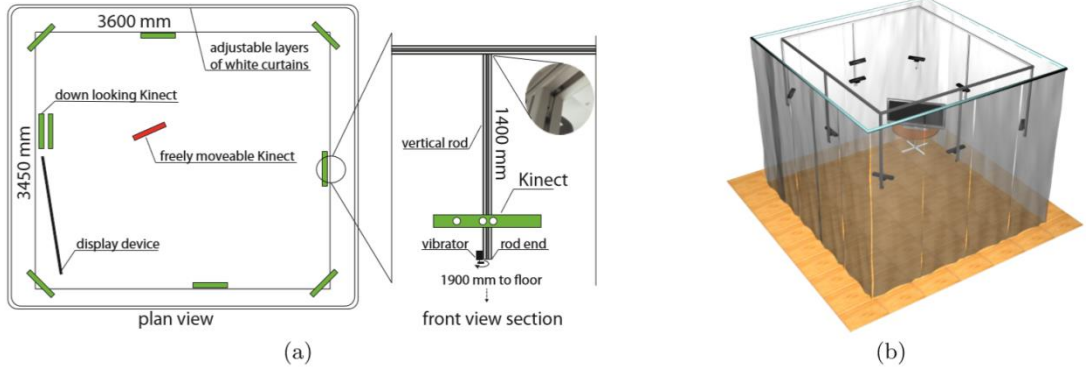
Bu soruna karşın görüntüleme hacimleri kesişen her Kinect'in ortama gönderdiği ışın demetini kiplererek (modulation), yansıyan demetlerden kendi gönderdiğini ayırt edebilmesi şeklinde bir çözüm ileri sürülebilir. Ancak Kinect for

Windows SDK v1.8 içerisinde böyle bir çözüm geliştirilmemiştir. Kinect' in kızılötesi projektörü ve IR kamerası kiplenmediği için, diğer Kinect kızılötesi projektörlerinden gelen kızılötesi ışın demetlerini kendisinin gönderdiği ışın demetlerinden ayırt edememektedir.

Butler vd. “Shake’n’Sense” adını verdikleri yöntemin çoklu Kinect kullanımının yol açtığı ışın girişimi ve sönümlenmeyi büyük ölçüde azaltılabileceğini ve sağlıklı derinlik verisi elde etmeyi sağlayacağını öne sürmüşler ve yaklaşımlarının Kinect'in elektronik yapısına veya iç yazılımına herhangi bir müdahale içermediğini ve FPS değerini düşürmediğini belirtmişlerdir [46]. Bu yöntemde kamera; üzerine oturtulan titreşim motoru (Şekil 4.3) ile titretilerek, ortama gönderilen ışın demetlerinin konumları değiştirilmektedir. Bu sayede her Kinect ortamdan yansıyan kendi gönderdiği ışın demetini düzgün bir biçimde alabilmektedir. Kainz vd. yoğun hacimsel verilerin çok sayıda Kinect'ten gerçek zamanlı elde edilmesi ve uygulamaları için geliştirdikleri “OmniKinect” yapısında girişimi engellemek için söz konusu yöntemi Kinect'lerin oturtulduğu destek ve çubukları titreterek uygulamışlardır. “OmniKinect” sisteminin genel görünümü Şekil 4.4' te gösterilmekte olup titreşen Kinect'ler yeşille ve titreşmeyen Kinect'ler kırmızı ile gösterilmektedir [47].



Şekil 4.3 Titreşim motorunun Kinect'e takılması (solda) Kinect'in serbest titreşim için akrilik çerçeveye oturtulması (sağda) [46]



Şekil 4.4 OmniKinect kurulumunun (a) plan görünümü (b) 3B görünümü [47]

2) **USB Veri Yolu Kullanımı:** Şekil 4.5’ de görüldüğü üzere Kinect, USB bant genişliğini başka bir Kinect’in aynı USB yöneticisi tarafından kontrolüne izin veremeyecek ölçüde tüketmektedir. Birinci Kinect bağlandığında sürücüler belleğe yüklenirken kullanılacak USB bant genişliği de rezerv edilmektedir. Diğer kameralar Şekil 4.6’da görüldüğü gibi aynı USB yöneticisi üzerinden bağlandığında bant genişliği ilk kameraya (kullanılmıyor olsa bile) rezerv edildiğinden diğer kameralar ilk kamera çıkarılıp rezerv edilen bant genişliği serbest kalıncaya kadar kullanılamaz. Bu nedenle çok sayıda Kinect kullanılması durumunda her Kinect’in farklı bir USB yöneticisi tarafından yönetilecek şekilde bağlanması gerekir. Aygıt yöneticisi üzerinden Şekil 4.7’ de görüldüğü üzere Kinect’lerin sorunsuz bağlandığının kontrolü yapılabilir.

Bant genişliği tüketen aygıtlar:

Açıklama	Kullanılan Bant Genişliği
Sisteme Ayrıldı	20%
USB Bileşik Aygıt	1%
Kinect for Windows Camera	61%
USB Bileşik Aygıt	1%

Listeyi güncelleştirmek için Yenile'yi tıklayın.

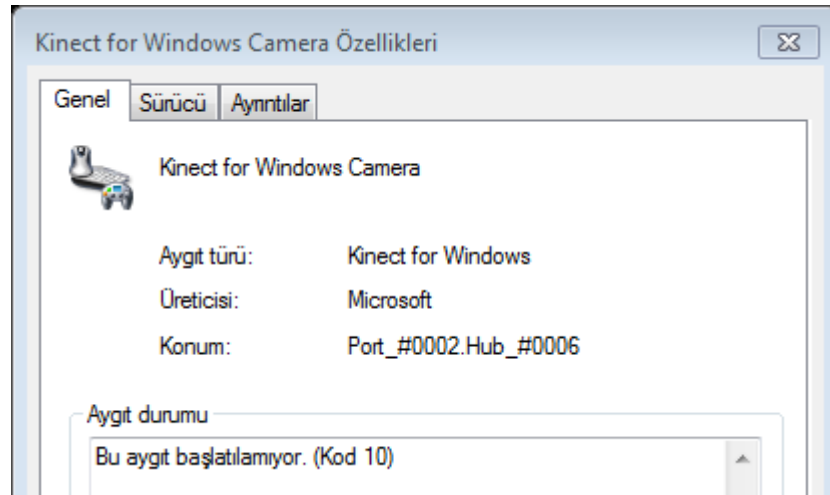
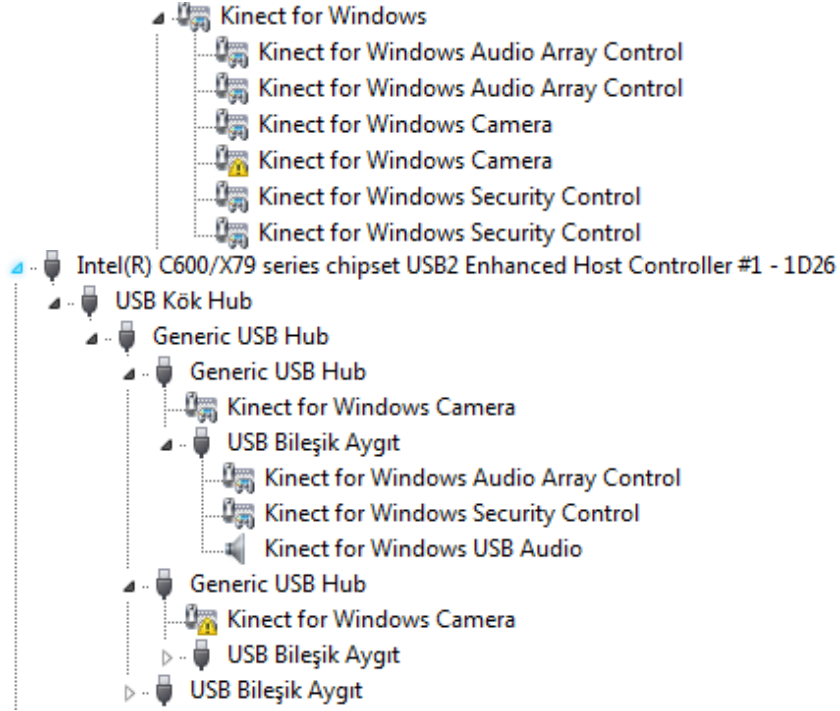
Yenile

Şekil 4.5 Kinect' in tükettiği bant genişliği

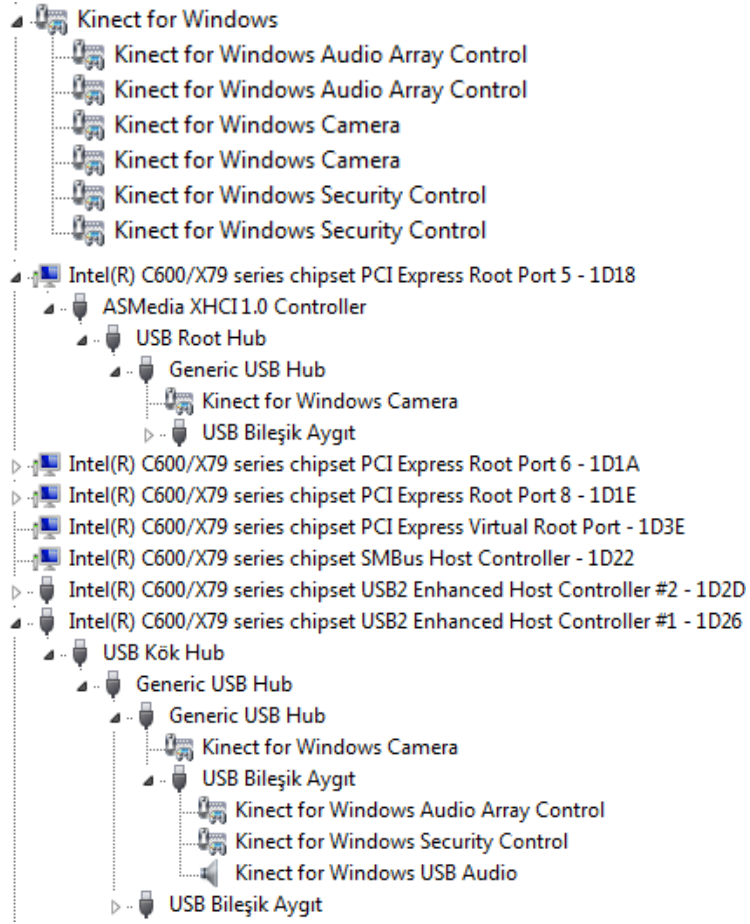
Yukarıda belirtilen sorunların yanı sıra çok sayıda Kinect kullanımının yazılım geliştirme aracı (SDK) desteğinin de bulunması gerekir. Kinect for Windows SDK v1.8 bir bilgisayar sisteminde aynı anda 4 Kinect sensöre kadar çoklu Kinect kullanımını desteklemektedir.

Kinect for Windows v2 kameralarda donanım kısıtlamaları (PCI Express bant genişliğinin yetmemesi vb.) nedeniyle aynı bilgisayar üzerinde çok sayıda kamera

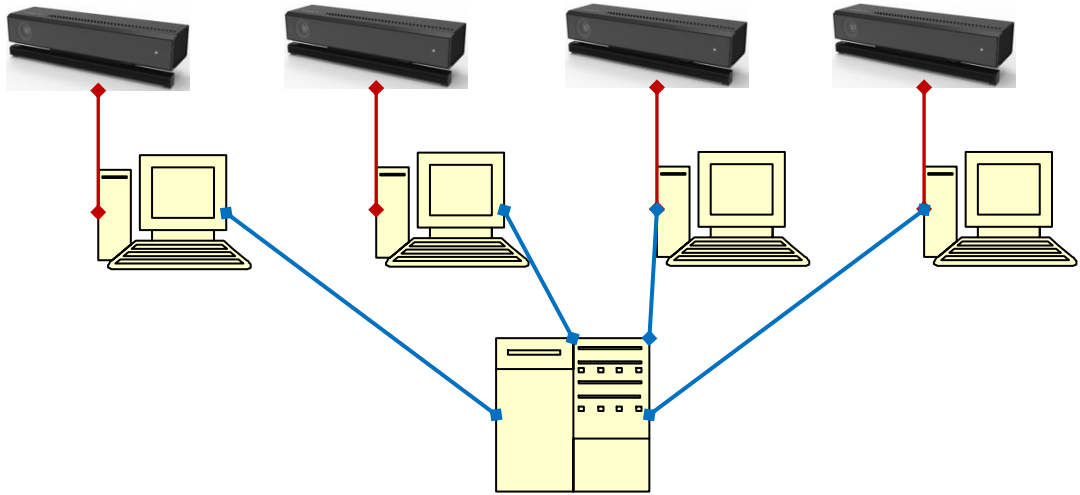
kullanımı Kinect for Windows SDK v2.0 itibariyle desteklenmemektedir. Kameraları farklı bilgisayarlara Şekil 4.8'deki gibi ayrı ayrı bağlayıp verileri ağ üzerinden birleştirilmesi mümkün olmasına karşın, ağ üzerinden merkezi bilgisayara gelen verilerin senkronizasyonu sorununun çözülmesi gerekmektedir. Ayrıca uygulamanın üzerinde çalışacağı bilgisayar donanımı ve ağ altyapısının renk bilgisi içeren nokta bulutu verileri gibi büyük miktarda verinin mümkün olabilecek en düşük gecikmeyle aktarımını sağlayabilecek seviyede olmalıdır.



Şekil 4.6 Aynı USB denetleyiciye bağlı ikinci Kinect'in çalışmaması



Şekil 4.7 Farklı USB denetleyicilerine bağlı sorunsuz çalışan Kinect' ler



Şekil 4.8 Çoklu Kinect for Windows v2 sistemi örneği

4.2 Kinect İskelet Verilerinin Birleştirilmesi

Bu bölümde Kinect iskelet verilerinin birleştirilmesinde parçacık filtresinin kullanımı önerilmiştir. Kinect iskelet verilerinin birleştirilmesinde SLAM (Simultaneous Localization and Mapping) problemi ve bir SLAM yöntemi olan (Monte Carlo Konumlaması olarak da adlandırılan) parçacık filtresi algoritması kullanılmıştır.

Parçacık filtresi eklem düzeyinde uygulanmakta ve her eklem için yeni yörüngesi, algoritmanın çıktısı olarak elde edilen birleşmiş eklem koordinat değerlerinden oluşturulmaktadır. Bir hareket, hareket eden uzuvu ait bir eklem grubu ile gösterilmektedir. Bir eklem için hareket bilgisi, hareketin yer aldığı görüntü karelerindeki koordinat verilerinin (normalize edilebilen ve diğer referans sistemlerine taşınabilen) bir araya getirilmesiyle oluşturulmaktadır.

Her bir Kinect'ten ayrı ayrı yakalanan hareket verilerinin birleştirilmesi tamamlandığında, birleşmiş hareket verileri birleşmiş eklem grubu olarak gösterilmektedir. Uygulama iki Kinect kullanılarak yapılmış olsa da, eş zamanlı çalışan daha fazla Kinect için genişletilebilir.

4.2.1 SLAM ve İskelet Eklemi Takibi Problemine Uyarlanması

SLAM; robotik alanında temel bir problem olan, Türkçesi "Eşzamanlı Konumlama ve Eşleştirme" anlamına gelen "Simultaneous Localization and Mapping" ifadesinin kısaltmasıdır. Bir robotun önceden bilinmeyen bir ortamda ilerlediği durumlarda SLAM hareket kontrolü için çok önemli olmaktadır. SLAM, herhangi bir ortam harita bilgisi olmadığında, tamamen (hatalar da hesaba katılarak) ölçüm ve tahminlere dayanarak uygulanabilmektedir.

SLAM algoritmaları aynı zamanda bilgisayarlı görme alanındaki nesne takibi problemlerine de uygulanabilmektedir. Takip edilen nesnenin hareketinin kestirilemediği durumlarda, nesnenin konumunu belirlemek ve hareket yörüngesini oluşturmak da SLAM problemleri kapsamındadır. Bu tez çalışmasındaki uygulamada takip edilen nesne iskelet eklemleri olup, ölçüm verileri Kinect tarafından sağlanmaktadır.

4.2.2 Parçacık Filtresi Algoritması

Parçacık filtreleri, filtreleme problemi çözümünde Kalman filtrelerinden sonraki aşamada bulunmaktadır. Liggins'in çoklu sensör verilerini birleştirme konulu kitabında parçacık filtrelerinin bu alanda kullanımı hakkında detaylı bilgiler ve uygulamalar yer almaktadır [48]. Thrun robotik alanındaki SLAM problemlerinin çözümünde parçacık filtrelerinin kullanımını yaygınlaştırmış ve parçacık filtreleri hareketli robotların konumunu saptamada altın standart haline gelmiştir [49].

Bu çalışmada, parçacık filtresinin hareketli robot takibindeki kullanımına benzer bir algoritma önerilmiş ve parçacık filtresinin çoklu Kinect iskeleti eklem takibine nasıl uyarlanacağı gösterilmiştir. Her iki durumda da takip edilen nesnelere ortak özelliklere sahip olmakla birlikte, iskelet eklem hareketi rastgele olabilmekte ve daha detaylı bir hareket modeli tanımlamasına ihtiyaç duymaktadır. Bu uyarlamada parçacık filtresi algoritmasının dört aşamasının tez kapsamındaki ele alınan probleme göre detayları şu şekildedir.

1. **Ölçüm:** Bir parçacık kümesi oluşturulur ve sınırlandırılmış dünya koordinat sistemi üzerine rastgele dağıtılır. Deney ortamına yerleştirilen iki Kinect' in dünya koordinat sistemine göre koordinatları referans noktaları (landmark) olarak tayin edilir. Eklemin başlangıç konumu bilinmediğinden, sensörün koordinat sistemi ile eklem arasındaki görece uzaklığa ait ölçüm verileri her iki Kinect'ten elde edilir. Bir sonraki adımda kullanılacak olan ağırlık değerlerinin atanması için her parçacığın referans noktalarına olan uzaklığı hesaplanır.
2. **Ağırlık Güncelleme:** Bu aşamada parçacıklar, parçacığın takip edilen eklem olma olasılığını belirten ağırlık değerleriyle gösterilmektedir. Ölçüm verilerine yakın uzaklık değerlerine sahip parçacıklar eklem adayları olarak belirlenir ve kendilerine diğer parçacıklardan daha büyük ağırlık değeri atanır. Bütün parçacıklar rastgele oluşturulduğu ve pek çoğu ölçüm verilerinden oldukça uzakta konumlandırıldığı için parçacıkların çoğu aday olamayacak ve yeniden örnekleme aşamasına (ölçüm verilerine yakın olan) çok az parçacıkla devam edilecektir.
3. **Yeniden Örnekleme:** Bu aşamada yeni bir parçacık kümesi rulet çarkı seçimi usulüyle oluşturulur. Daha büyük ağırlık değerlerine sahip parçacıklar çarkta daha büyük dilimi kaplayacak şekilde yerleştirilir, bu sayede oluşturulan yeni

parçacığın güçlü adayların komşuluğunda konumlanması şansı artmaktadır. Her parçacık için rulet çarkı çevrilerek parçacıkların hangi aday komşuluğunda konumlandırılacağı belirlenir ve yeni parçacık kümesi oluşturulur.

4. **Hareket Güncelleme:** Takip edilen eklem hareketine uygun olarak parçacıkların hareket etmesi için bir hareket modeli oluşturulur ve parçacıklar belirlenen bu hareket modeline göre hareket ettirilir. Hareket modeli takip edilen nesnenin hareketine uygun biçimde belirlenmelidir. Takip edilen eklemlerin hareketleri her yöne olabileceğinden ve kestirilemeyeceğinden konum değişimi, hız ve ivmeyi hesaba katan bir kinematik modelin eklem hareketi için daha uygun olacağı ve daha iyi sonuçlar vereceği düşünülmüştür.

Bu aşamalar yeni ölçüm verileriyle tekrarlanır. Takip edilen eklem konumu, aday parçacıkların konumlarının ortalaması alınarak belirlenir.

Parçacık filtresi algoritmasının sözde kodu şu şekildedir:

Algoritma parcacik_filtresi(X_{t-1}, u_t, z_t):

- 1: $\bar{X}_t = X_t = \emptyset$
- 2:
- 3: **for** $m = 1$ to M do **Yeni parçacıklar oluşturma**
- 4: sample $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$
- 5: $w_t^{[m]} = p(z_t | x_t^{[m]})$ **Ağırlık değerlerini hesaplama**
- 6: $\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
- 7: **endfor**
- 8: **for** $m = 1$ to M do
- 9: draw i with probability $\propto w_t^{[i]}$
- 10: add $x_t^{[i]}$ to X_t
- 11: **endfor**
- 12: **return** X_t

Burada $X_t := x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}$; $x_t^{[m]}$ ($1 \leq m \leq M$ olmak üzere) parçacıktan oluşan bir parçacık kümesini belirtmektedir. Parçacıkların her biri takip edilen iskelet eklemine gerçek konumu hakkında birer hipotez önermektedir. M toplam parçacık sayısını belirtmekte ve bu sayı her iterasyonda hassasiyeti arttırmak için 1000 veya daha büyük seçilmektedir. Her parçacığın olabilirlik kestirimini tanımlamak için önem ağırlığı $w_t^{[m]}$ hesaplanır. Daha büyük önem ağırlığına sahip parçacıklar iyi adaylar olarak değerlendirilir ve yeniden örnekleme aşamasında takip edilen robotun

sonraki durumunu daha iyi temsil edeceği öngörülür. Yeniden örnekleme aşamasından sonra bu aday parçacıklar etrafında yeni bir parçacık kümesi oluşturulur ve algoritma bu güncellenmiş küme ve yeni ölçüm değerleriyle ilerletilir.

Algoritmanın kodlanmasında, hareket güncelleme ve yeniden örnekleme aşamaları takibin genel performansını doğrudan etkilemektedir.

5. UYGULAMALAR

Tez çalışmasında önerilen Fréchet uzaklığı kullanılarak hareket yörüngelerinin benzerlik ölçümü ve parçacık filtresi algoritması ile iskelet eklemi birleştirme yöntemleri iki ayrı uygulama içerisinde gösterilmektedir.

Fréchet uzaklığı kullanılarak hareket yörüngelerinin benzerlik ölçümü uygulamasında kullanıcıların belirli bir süre içerisinde yaptıkları uzuv hareketleri kaydedilmiş ve bu kayıtlardan uzuv hareketlerini tanımlayan eklem noktalarının (kullanıcı iskeletinin eklemleri) her bir görüntü karesindeki konumu 3B olarak elde edilmiştir.

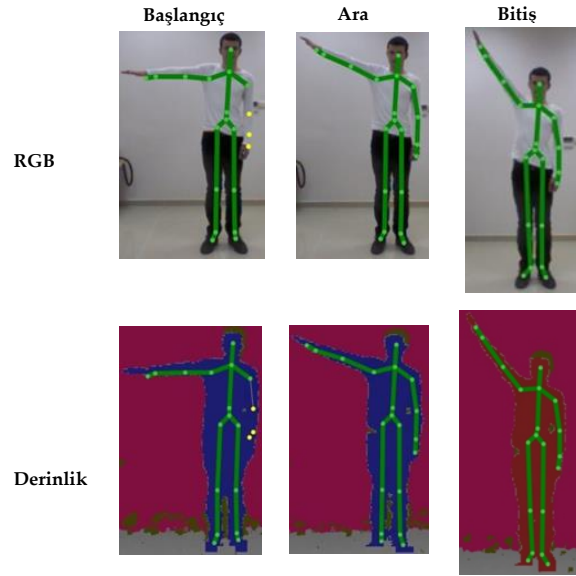
Hareket algılamada Kinect' in başarımı, uygulamada kullanılan mesafe için yeterli olup, kameranın derinlik sensörünün yapısından ve kayıt alınan ortamdaki kaynaklanan nedenler dolayısıyla, kullanıcılarının ham iskelet (kullanıcıyı tanımlayan eklem noktalarının bütünü) verisi üzerinde gürültü bulunmaktadır [50]. Kinect eklem koordinatları ölçümünde var olan bu gürültü nedeniyle, parçacık filtresinde ölçüm verileri olarak kullanılacak bu değerlere ayrıca gürültü eklemeye gerek duyulmamaktadır.

Bu uygulamanın işleyişi genel olarak Şekil 5.1' de gösterilmiştir. İlk olarak; kullanıcının hareketlerinin 3B sanal ortamda takip edilebilmesi için, kullanıcının eklem noktalarının dünya koordinat sisteminden (world coordinate system) 3B sanal koordinat sistemine (view coordinate system) dönüştürülmesi gerekir. Bu işlem Microsoft Kinect SDK kullanan MATLAB' ın "Image Acquisition Toolbox" kütüphanesi ile yapılabilmektedir.



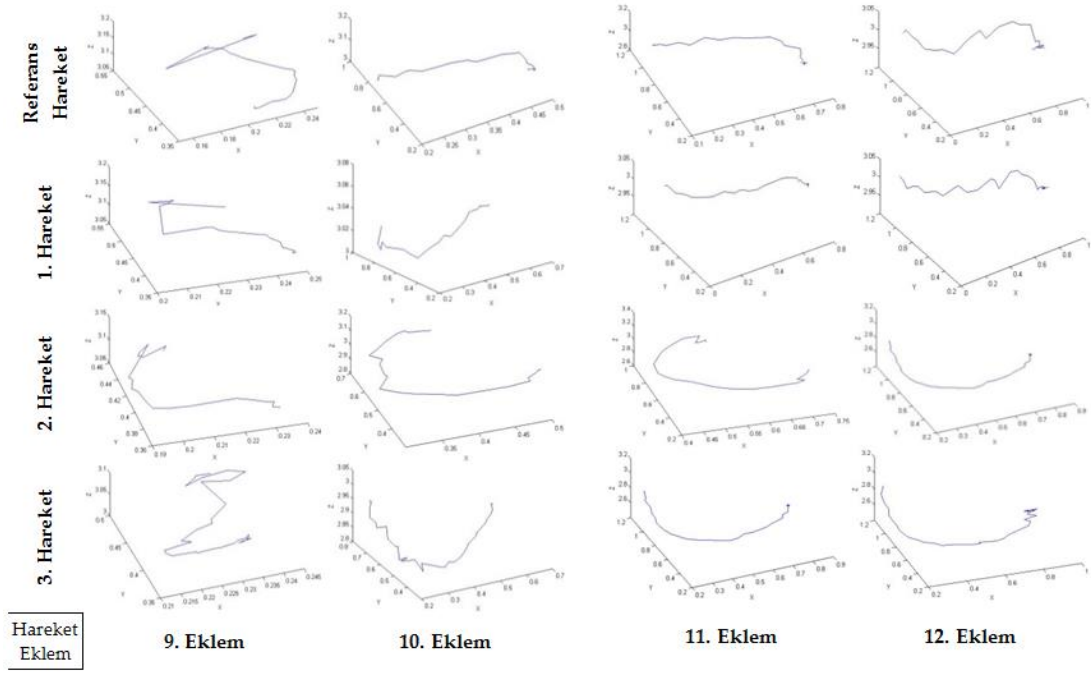
Şekil 5.1 Fréchet uzaklığı ölçümü uygulamasının genel işleyişi

Örnek uzuv hareketi olarak kullanıcının sağ kolu yatay eksen boyunca uzatıldıktan sonra omuz hizasına doğru kaldırılması şeklinde belirlenmiştir. (Şekil 5.2) Bu hareketin yakalanması için 9, 10, 11 ve 12. eklem noktalarının koordinat değişimleri takip edilmiştir. Kinect kendi koordinat sisteminde (view coordinate system) kameranın fiziksel konumunu (dünya koordinat sistemi içerisindeki) orijin olarak kabul ettiğinden, uygulama süresince hareketleri yakalanacak kişi kameraya göre aynı noktada konumlandırılmıştır. Bu şekilde elde edilen ilk hareket referans hareketine benzer, diğer iki hareket ise referans hareketine benzemeyen hareketler olarak kaydedilmiştir.



Şekil 5.2 Örnek hareketin başlangıç ara ve bitiş pozisyonları

Referans hareketin ve bu hareketle karşılaştırılacak diğer üç hareketin verileri Şekil 5.3' de gösterilmektedir.



Şekil 5.3 Referans hareket ve karşılaştırılacak hareketlerin verileri

Uygulamadaki uzuv hareketini tanımlayan eklem noktaları bazında, 3 hareketin yörüngelerinin XY ve XZ izdüşümlerinin referans hareketin yörüngesinin XY ve XZ izdüşümüyle arasındaki Fréchet uzaklığı değerleri Çizelge 5.1’ de verilmektedir.

Çizelge 5.1 Hareketlerin XY izdüşümlerinin aralarındaki Fréchet uzaklığı

Hareket	Eklem	XY Fréchet Uzaklığı	XZ Fréchet Uzaklığı
1	9	0.047	0.051
1	10	0,048	0.024
1	11	0.050	0.040
1	12	0.108	0.107
2	9	0.065	0.061
2	10	0.216	0.187
2	11	0.420	0.391
2	12	0.502	0.467
3	9	0.066	0.094
3	10	0.221	0.205
3	11	0.333	0.380
3	12	0.378	0.422

Parçacık filtresi algoritması ile iskelet eklemi birleştirme uygulamasında kullanıcı hareketlerinin herhangi başka bir nesnenin engellemesine maruz kalmadan yakalanabilmesi için kamera bakış doğrultusu kullanıcının bakış doğrultusu ile 45° açı yapacak şekilde iki adet Kinect kamerası bakış doğrultularının kesiştiği noktanın 2,4 metre uzaklığına Şekil 5.4' te görüldüğü şekilde yerleştirilmiştir. Aynı anda çalıştırılan iki kameralı düzenek ile kullanıcıların belirli bir süre içerisinde yaptıkları uzuv hareketleri eşzamanlı olarak kaydedilmiş ve elde edilen iki iskelet verisinde de uzuv hareketlerini tanımlayan eklem noktalarının her bir derinlik görüntü karesindeki konumu 3B olarak elde edilmiştir. İki iskeletin verileri eklem noktaları bazında parçacık filtresi kullanılarak birleştirilmiştir.



Şekil 5.4 Çoklu Kinect deney düzeneği

Çok sayıda Kinect kullanılarak iskelet verisi elde edilen diğer çalışmalarda; iskeletlerin sağlıklı bir şekilde birleştirilebilmesi için bir Kinect' in ana Kinect olarak belirlenmesi, diğer Kinect iskeletlerinin ana Kinect' in referans sistemine taşınması önerilmektedir. Bunun için diğer iskeletlerin eklem noktaları bir döndürme matrisi ile çarpılarak ana Kinect' in referans sistemine göre konumlandırılmaktadır. Bu çalışmada ise Kinect kameraları, parçacık filtresi için ölçüm verileri sağlamak amacıyla kullanıldığından, uygulamada Kinect koordinat sistemini kullanmaya gerek duyulmamıştır. Bunun yerine Kinect koordinat sisteminde bulunan eklemlerin koordinatlarının dünya koordinat sisteminde karşılığı geometrik dönüşümler kullanılarak hesaplanmıştır.

Bu deney düzeneğinde kullanıcı ve Kinect'ler arasında görüşü engelleyecek herhangi bir nesne yoksa, hareketi her iki Kinect de kaydedebilmektedir. Kullanıcı sağ kolunu kaldırdığından, kullanıcının yönelimine bağlı olarak birinci Kinect'den alınan sağ kol eklem verilerinin hatalı olması muhtemeldir.

Öte yandan eğer kullanıcı ve Kinect'lerden birinin arasında bir nesne varsa, kullanıcının derinlik haritasındaki silüetini bloklayabilir ve buna bağlı olarak harekete ilişkin bazı eklemleri örtebilir. Bu durumda nesnenin görüş hacmini blokladığı Kinect'ten alınan eklem verileri eklem hakkında güvenilir bilgi veremez ve dolayısıyla parçacık filtresi için gereken ölçüm verileri diğer Kinect' ten alınana dayanmak durumundadır.

Uygulama ortamı $4 \times 4 \times 4$ kübik hacimle sınırlandırılmış olup birinci Kinect (2.4, 0, 0) konumuna ve ikinci Kinect de (0, 0, 2.4) konumuna yerleştirilmiştir.

Uygulama ortamı ve iki Kinect farklı koordinat sistemlerine sahiptir. Uygulama ortamının koordinat sisteminin (dünya koordinat sistemi) eksenleri sağ el kuralıyla belirlenirken, Kinect' in koordinat sisteminin temel eksenleri sol el kuralı ile belirlenir. Ayrıca aynı eklem her bir Kinect'e göre konumu ve Kinect'lerin ortam koordinat sistemi içindeki yönelimleri de farklı olduğundan, ortam koordinat sistemine göre hizalanması gerekir. Eklem koordinatlarının doğru hesaplanması için her iki Kinect'in uygulama ortamı referans sistemiyle arasındaki geometrik dönüşümler belirtilmelidir.

İki koordinat sistemi arasındaki bir geometrik dönüşüm, dönüşüm matrisi şeklinde ifade edilmektedir. Kinect koordinat sistemindeki her P noktasına ortamın referans sisteminde karşılık gelen P' noktası denklem (5.1)'de verilen dönüşüm matris çarpımı ile bulunabilmektedir.

$$P' = MP \quad (5.1)$$

Burada; M taşıma ve döndürme matrislerinin bir kombinasyonu olan bir dönüşüm matrisidir. 3B noktalar için taşıma ve döndürme matrisleri homojen koordinatlar kullanılarak 4×4 hale gelmektedir. Bu sayede M , matris çarpımları biçiminde (chain matrix product) yazılabilmektedir. Ortamdaki Kinect'lerin dönüşüm matrisleri birinci ve ikinci Kinect için sırasıyla M_1 ve M_2 olarak tanımlanmaktadır.

Birinci Kinect kamerası için geometrik dönüşüm matrisi M_1 şu şekilde tanımlanmaktadır.

$$M_1 = T_1 R_1 = \begin{bmatrix} 1 & 0 & 0 & 2.4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

Burada; T_1 birinci Kinect'in koordinat sistemine ait orijinin taşınma matrisi, R_1 temel eksenleri hizalamak için gereken döndürme matrisidir.

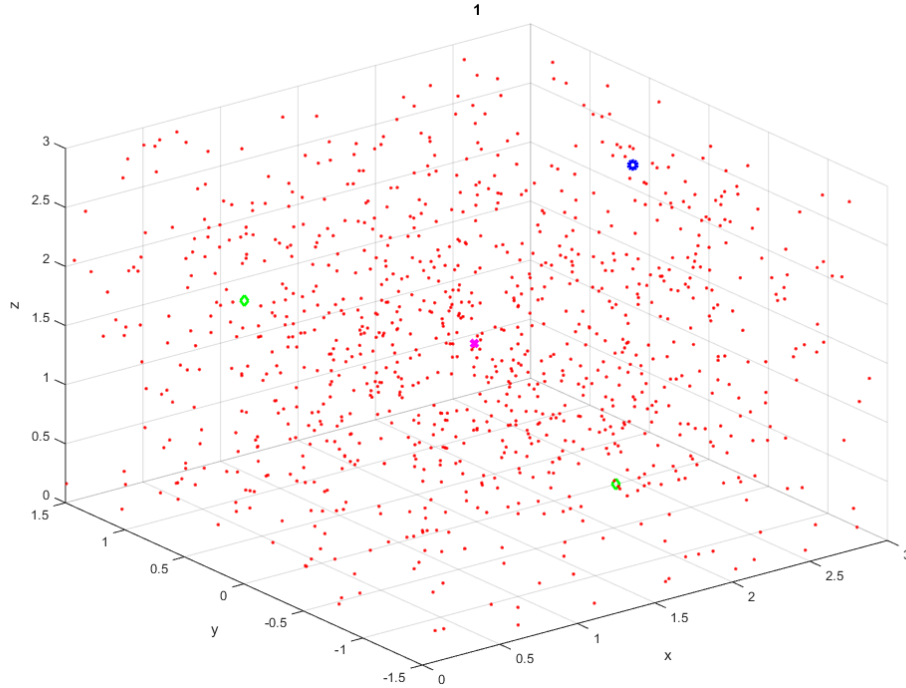
İkinci Kinect kamerası için geometrik dönüşüm matrisi M_2 şu şekilde tanımlanmaktadır.

$$M_2 = T_2 R_{21} R_{22} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2.4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

Burada; T_2 ikinci Kinect'in koordinat sistemine ait orijinin taşınma matrisi, R_{21} ve R_{22} temel eksenleri hizalamak için gereken döndürme matrisleridir.

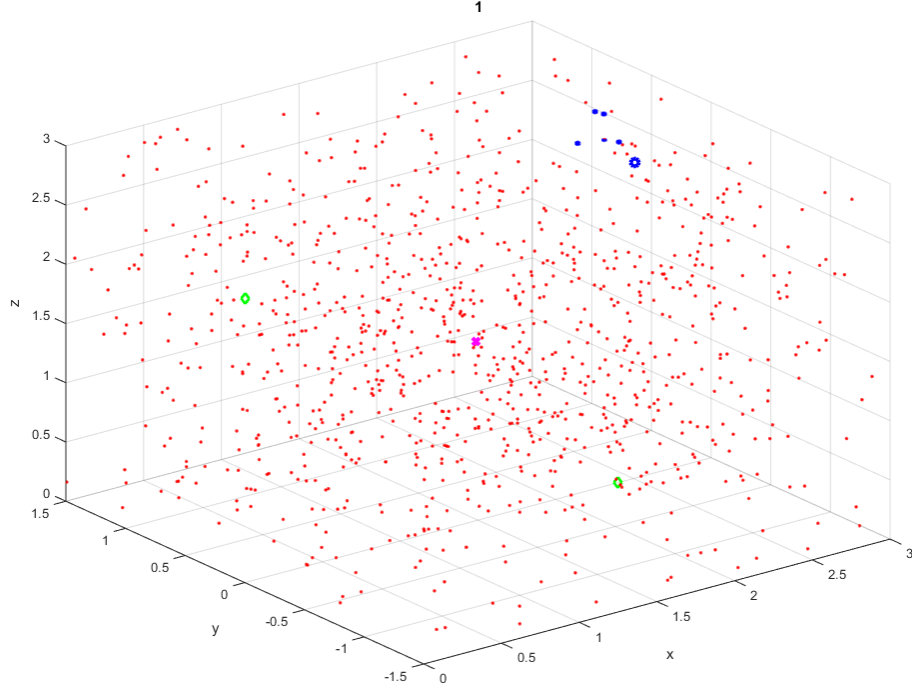
Parçacık filtresinin ağırlık güncelleme aşamasında, parçacıklar 3B uzayda yer aldıklarından parçacıkları filtrelemede iki boyutlu Gauss fonksiyonu kullanılmıştır.

Eklem verilerinin parçacık filtresi algoritması kullanılarak birleştirilmesi uygulaması MATLAB ortamında yapılmıştır. Algoritmanın bir iterasyonunun aşamaları Şekil 5.5 – Şekil 5.9 arasında gösterilmektedir.



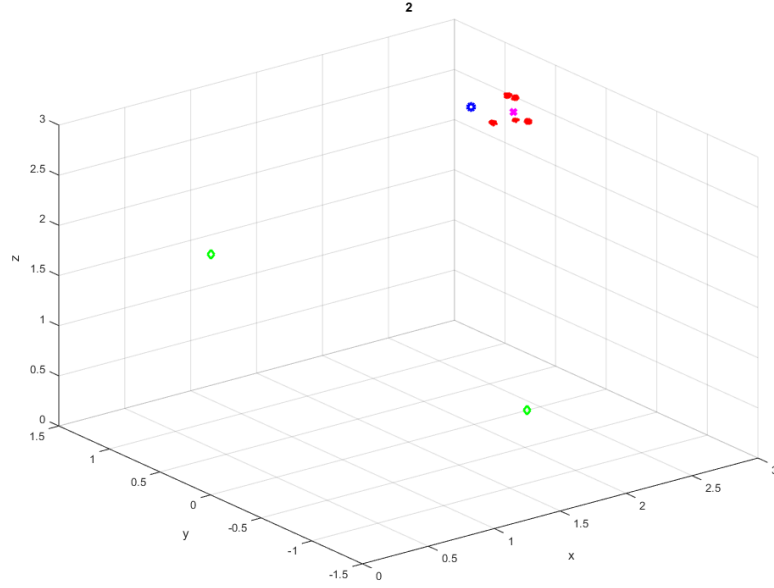
Şekil 5.5 Ölçüm aşamasında parçacıklar ortama rastgele dağıtılarak ölçüm değerleri alınır

Ölçüm aşamasının başlangıcında Şekil 5.5’de kırmızı noktalar olarak gösterilen parçacıklar ortama rastgele dağıtılmış ve her iki Kinect’ ten alınan ilk görüntü karesindeki koordinat değerleri ölçüm verileri olarak verilmiştir. Kinect’lerin ortam koordinat sistemi içerisindeki konumları yeşil renkle gösterilmektedir. Ağırlık güncelleme aşamasında parçacıklara ağırlıkları atanmış ve Şekil 5.6’da ölçüm değerlerine en yakın parçacıklar mavi noktalar olarak gösterilerek ‘aday’ parçacıklar olarak belirlenmiştir.



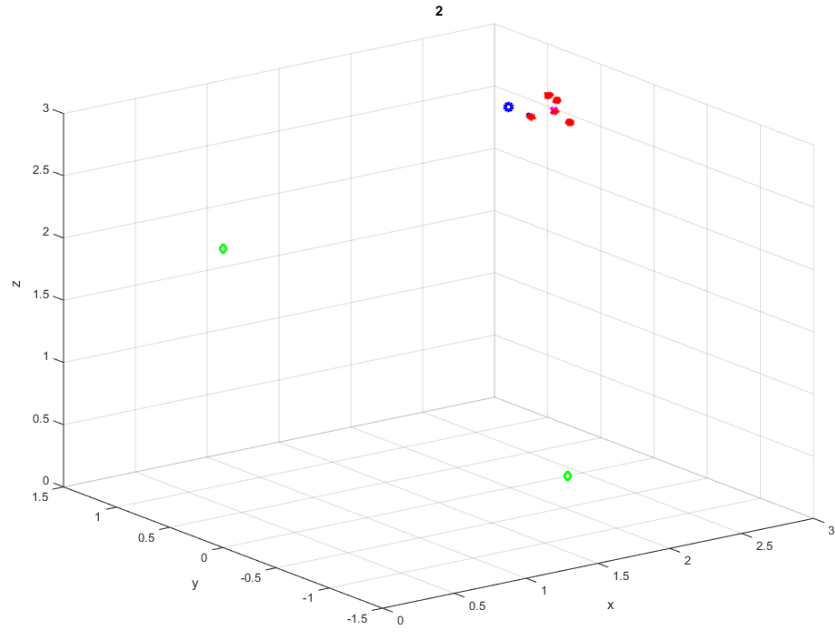
Şekil 5.6 Ağırlık güncelleme aşamasında "aday" olmayan parçacıklar elenir

Şekil 5.7’de yeniden örnekleme aşamasında yeni parçacıkların ‘aday’ parçacıklar etrafında kümelenecek şekilde rastgele oluşturulması gösterilmektedir.



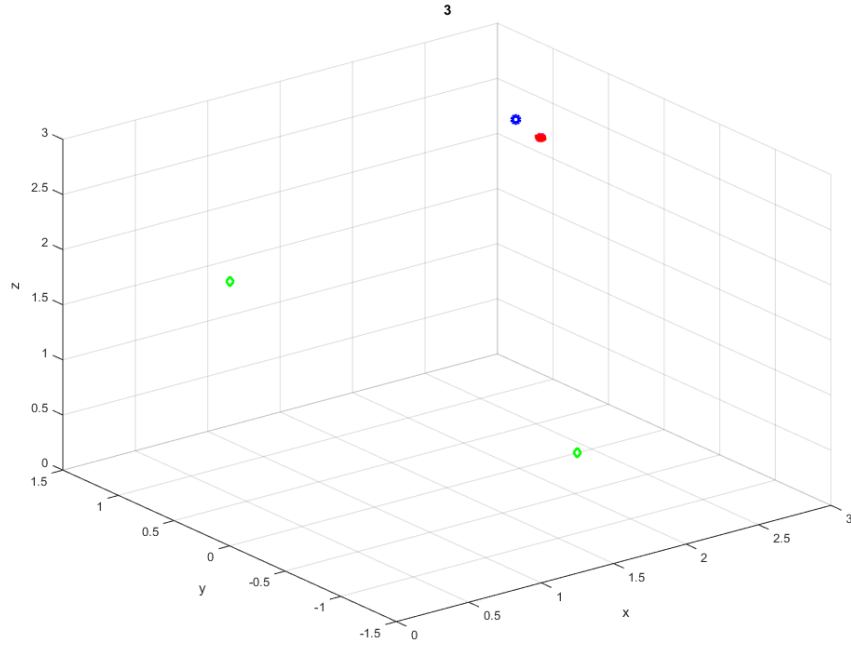
Şekil 5.7 Yeniden örnekleme aşamasında yeni parçacıklar oluşturulur

Oluşturulan bu yeni parçacıklar hareket güncelleme aşamasında göre hareket ettirilmekte ve parçacıkların yeni konumları Şekil 5.8’ de çizdirilmektedir.



Şekil 5.8 Hareket güncelleme aşamasında parçacıklar hareket modeline göre hareket ettirilir

Bir sonraki iterasyonda yeni ölçüm verileri kullanılarak parçacıklara yeni ağırlık değerleri atanmakta ve Şekil 5.9’da görülen bu ölçüm değerlerine en yakın ‘aday’ parçacıklar oluşmaktadır. Bu iterasyondan sonra; oluşturulan bütün parçacıkların koordinat değerlerinin ağırlıklı ortalaması, o görüntü karesi için birleşmiş eklem koordinatlarını vermektedir. Bu koordinatlar görüntü kareleri boyunca bir araya getirilerek yörünge oluşturulur.



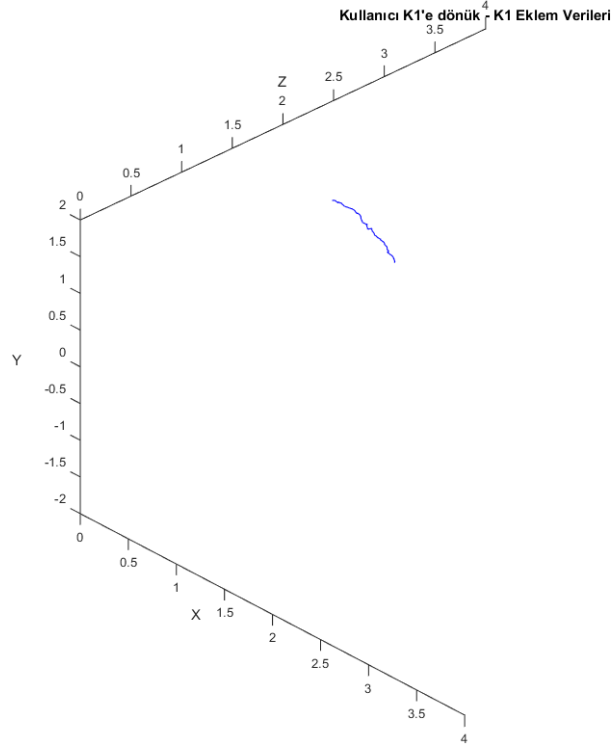
Şekil 5.9 Bir sonraki iterasyonda ölçüm ve ağırlık güncelleme aşamaları

Uygulama başlangıcında hareketi kaydedilecek kullanıcı T pozunu verecek şekilde konumlandırılarak sağ kolunu kaldırması sağlanmıştır. İskelet verileri her iki Kinect'ten eş zamanlı olarak MATLAB'ın görüntü elde etme aracı (image acquisition toolbox) kullanılarak elde edilmiş ve (f bir oturumda yakalanan derinlik görüntü kareleri olmak üzere) $f \times 3$ matrislerde kaydedilmiştir. Takip edilen eklemlerin her iki Kinect'ten elde edilen koordinat verileri ayrı ayrı tanımlanmıştır.

Kullanıcının Kinect'lerden birine tamamen döndüğü durumlarda sağ kolu kaldırma hareketini bir Kinect doğru bir şekilde yakalarken diğeri ise gövdenin sağ kolu örtmesi veya vücut duruşundaki bozukluk nedeniyle hareketi algılayamamaktadır. Bu durum öz kaynaklı örtme problemi olarak nitelendirilmektedir. Önerilen yaklaşımda, parçacık filtresi algoritmasının öz kaynaklı örtme probleminin çözümüne ne düzeyde katkı sağladığı araştırılmıştır.

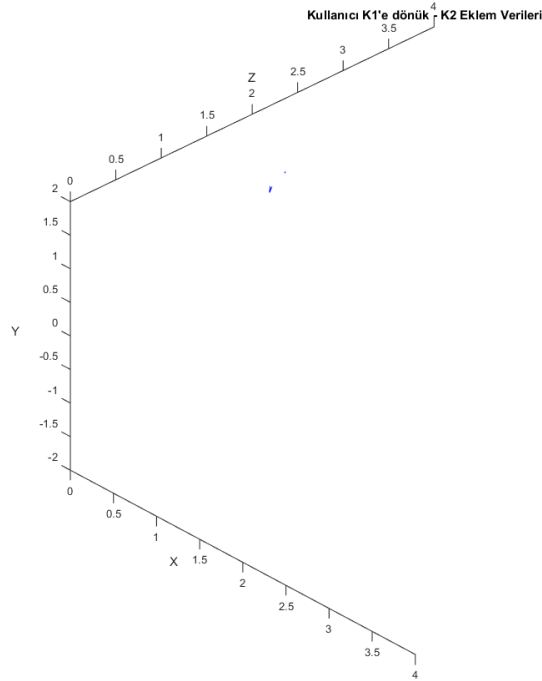
Eklem hareket yörüngelerinin gösterildiği grafikler MATLAB kullanılarak elde edilmiştir. Kinect'in koordinat eksenleri sol el kuralına göre tanımlandığından, Kinect referans sistemine göre hareketi yorumlamak için X eksenini ters yönde ele alınmalıdır.

Kullanıcı birinci Kinect'e dönük bir şekilde sağ kol kaldırma hareketi yaptırılmıştır. Şekil 5.10'da birinci Kinect'ten alınan 12. eklem (sağ el) verileri birinci Kinect'in referans sistemine göre gösterilmektedir.



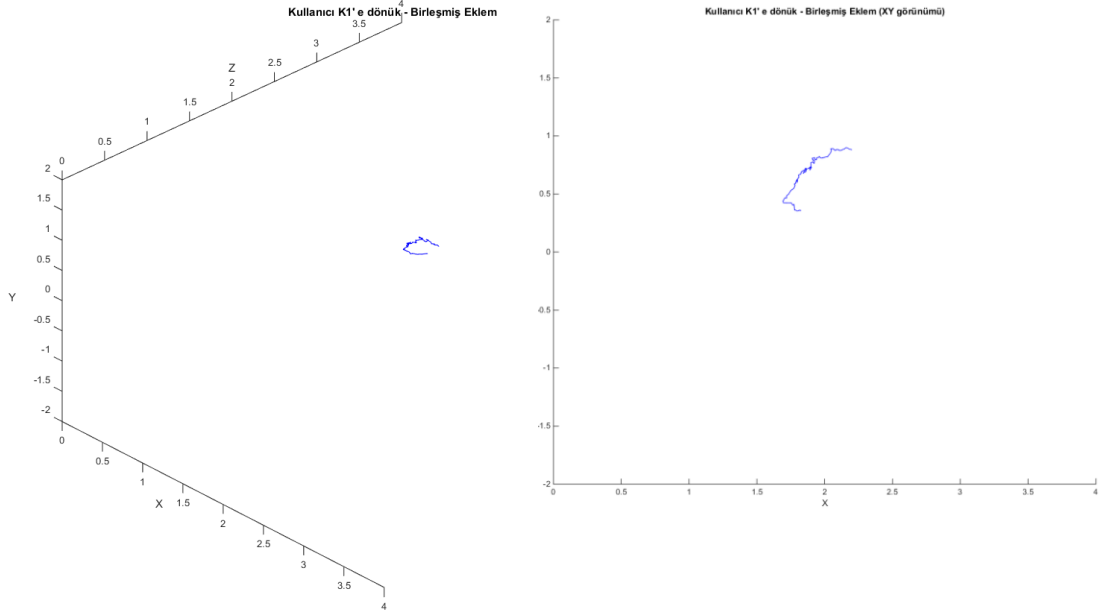
Şekil 5.10 Birinci Kinect'e dönük kullanıcının 12. ekleminin birinci Kinect verileri

Grafikten de anlaşılacağı üzere birinci Kinect hareketi doğru şekilde algılayabilmektedir. Şekil 5.11'de ikinci Kinect'ten alınan 12. eklemin verileri ikinci Kinect'in referans sistemine göre gösterilmektedir.



Şekil 5.11 Birinci Kinect'e dönük kullanıcının 12. ekleminin ikinci Kinect verileri

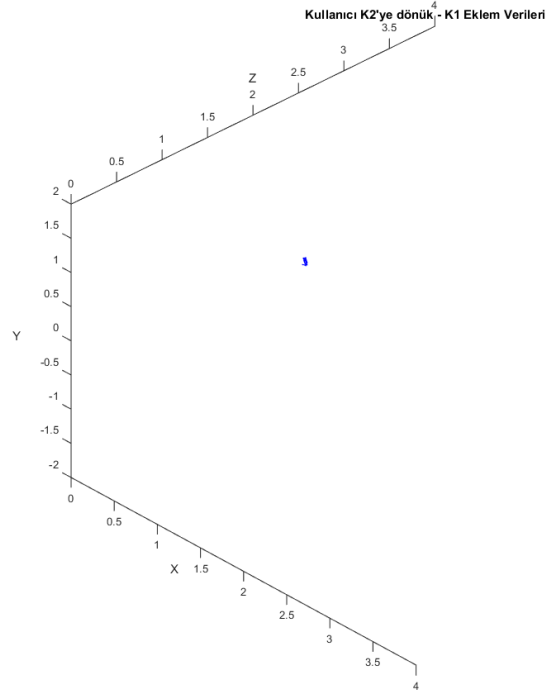
Bu pozisyonda kullanıcının vücudu sağ kolunu örttüğünden ikinci Kinect eklem verilerini Şekil 5.11'den de anlaşılacağı üzere hatalı algılamaktadır. Şekil 5.12'de ise 12. eklemin birleştirilmiş koordinatları ortamın koordinat sistemine göre gösterilmektedir.



Şekil 5.12 Birinci Kinect'e dönük kullanıcının 12. ekleminin birleştirilmiş verileri

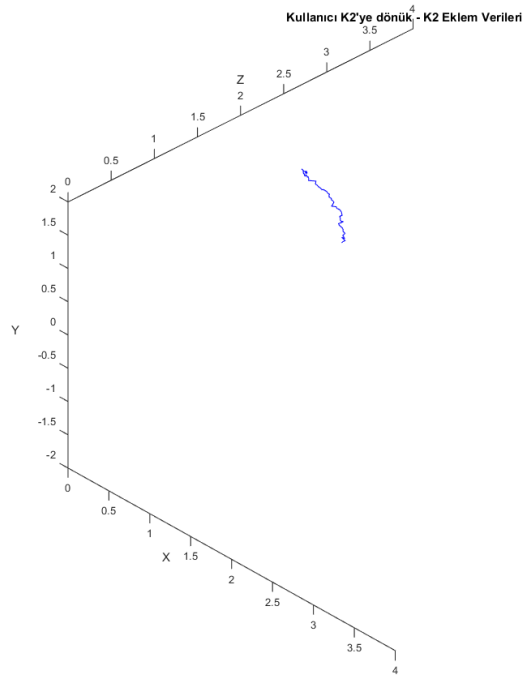
Bu grafikte bazı noktalar düzensiz olmakla birlikte istenen yörünge nitel olarak oluşturulabildiği görülmekte olup, diğer Kinect'ten gelen hatalı veriler söz konusu düzensizliğe sebep olmaktadır. Şekil 5.4 göz önüne alındığında eklemin Y ve X koordinat değerlerinin artması beklentiye uygun olarak gerçekleşmiştir.

Bir diğer uç durum örneği olarak kullanıcı ikinci Kinect'e dönük bir şekilde sağ kol kaldırma hareketi yaptırılmıştır. Şekil 5.13' de birinci Kinect'ten alınan 12. eklemin (sağ el) verileri birinci Kinect'in referans sistemine göre gösterilmektedir.



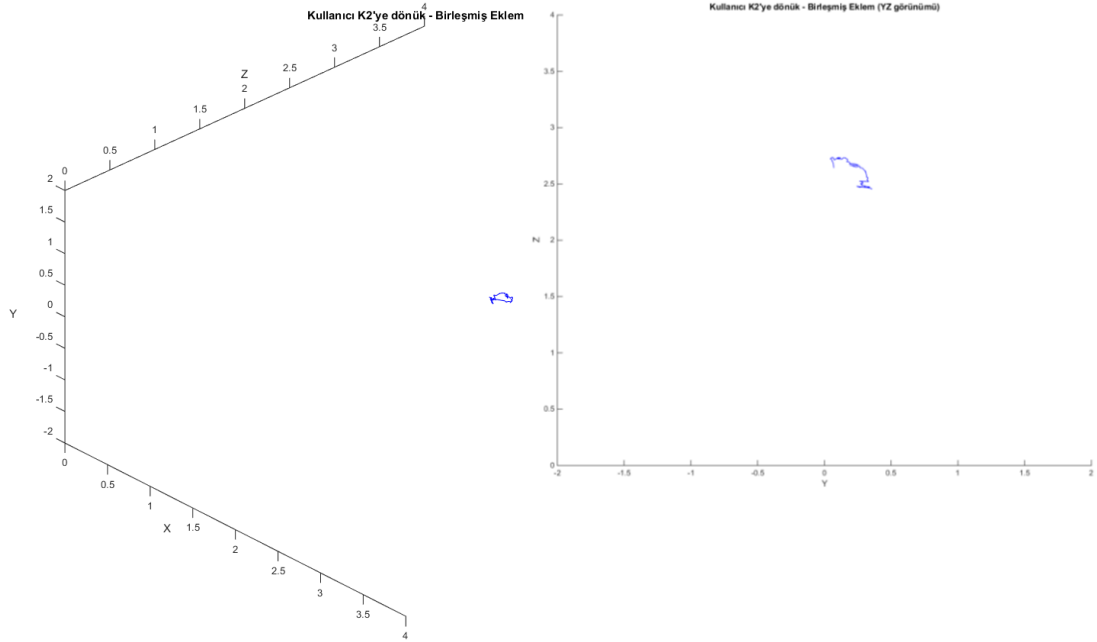
Şekil 5.13 İkinci Kinect'e dönük kullanıcının 12. ekleminin birinci Kinect verileri

Bu pozisyonda kullanıcının sağ kolunun eklemleri birbirini örttüğünden kullanıcı silüeti bozulmakta ve birinci Kinect'in eklem verilerini Şekil 5.13'den de anlaşılacağı üzere hatalı algılamasına neden olmaktadır. Şekil 5.14'de 12. eklemin ikinci Kinect'ten alınan verileri ikinci Kinect'in referans sistemine göre gösterilmektedir.



Şekil 5.14 İkinci Kinect'e dönük kullanıcının 12. ekleminin ikinci Kinect verileri

Grafikten de anlaşılacağı üzere birinci Kinect hareketi doğru şekilde algılayabilmektedir. Şekil 5.15’de ise 12. eklemin birleştirilmiş koordinatları ortamın koordinat sistemine göre gösterilmektedir.

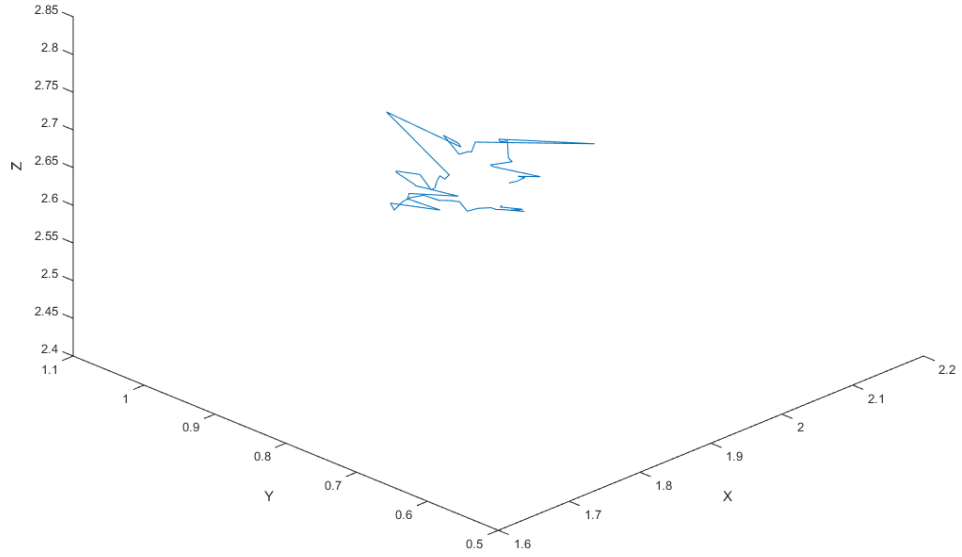


Şekil 5.15 İkinci Kinect'e dönük kullanıcının 12. ekleminin birleştirilmiş verileri

Bu grafikte bazı noktalar düzensiz olmakla birlikte istenen yörüngenin nitel olarak oluşturulabildiği görülmekte olup, diğer Kinect'ten gelen hatalı veriler söz konusu düzensizliğe sebep olmaktadır. Şekil 5.4 göz önüne alındığında eklemin Y koordinat değeri artarken Z koordinat değerinin azalması beklentiye uygun olarak gerçekleşmiştir.

Kullanıcı, parçacık filtresi için ölçüm verilerinin her iki Kinect'ten sağlanabileceği bir durum olan her iki Kinect'e 45 derece açı yaparak yüzü karşıya bakacak şekilde konumlandırılmış ve aynı hareket tekrarlanmıştır. Şekil 5.16'da birinci Kinect'ten alınan 12. eklemin (sağ el) verileri ortamın referans sistemine göre gösterilmektedir.

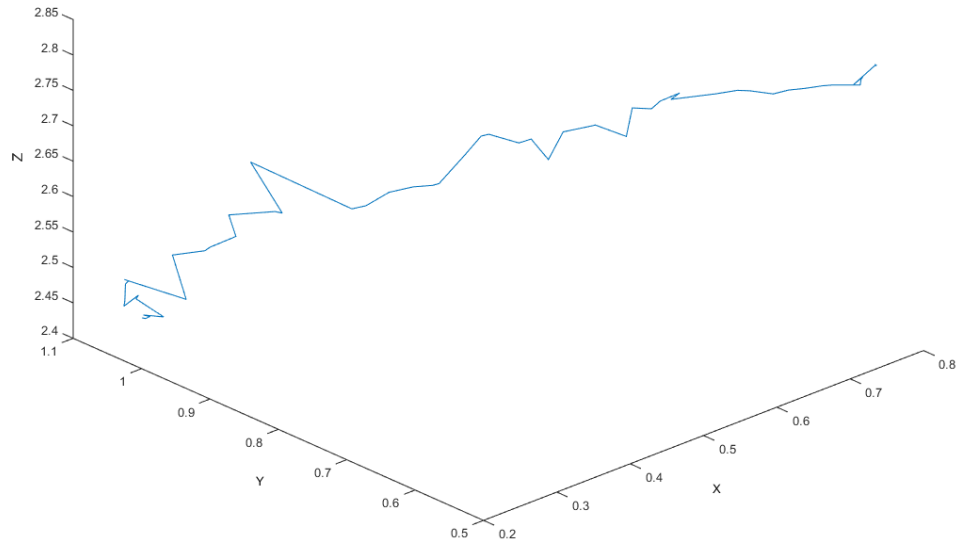
Dünya koor. sistemine göre K1 koordinatları



Şekil 5.16 Kullanıcı 12. ekleminin ortam referans sistemine göre 1. Kinect koordinatları

Şekil 5.17’de 12. eklemin birinci Kinect’ten alınan verileri birinci Kinect’in referans sistemine göre gösterilmektedir.

1. Kinect koordinat sistemine göre K1 koordinatları

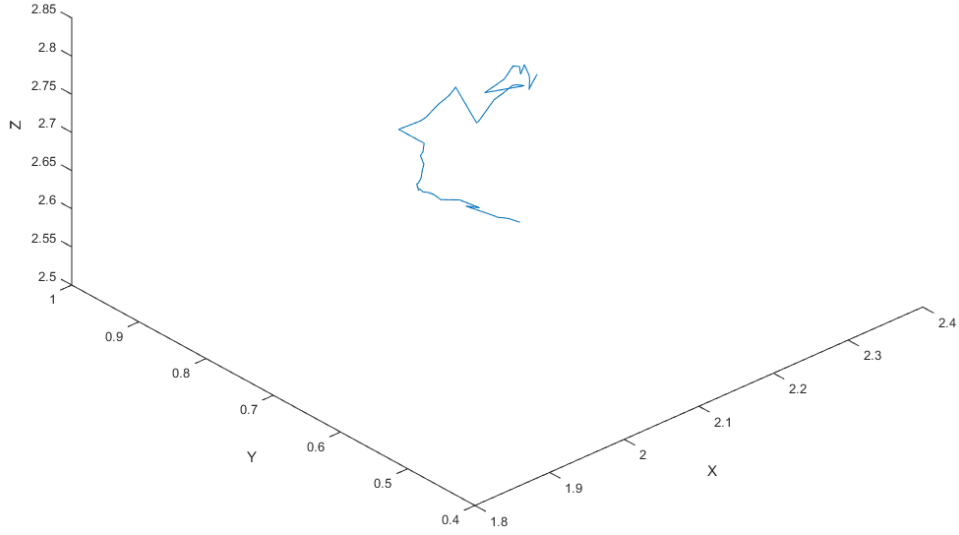


Şekil 5.17 Kullanıcı 12. ekleminin birinci Kinect’e göre 1. Kinect koordinatları

Kullanıcının duruşu birinci Kinect’in Y eksenine etrafında 45 derece dönmüş olduğundan grafikteki yörünge oluşmaktadır. Şekil 5.18’ de 12. eklemin ikinci

Kinect'ten alınan verileri ortamın referans sistemine göre gösterilmektedir.

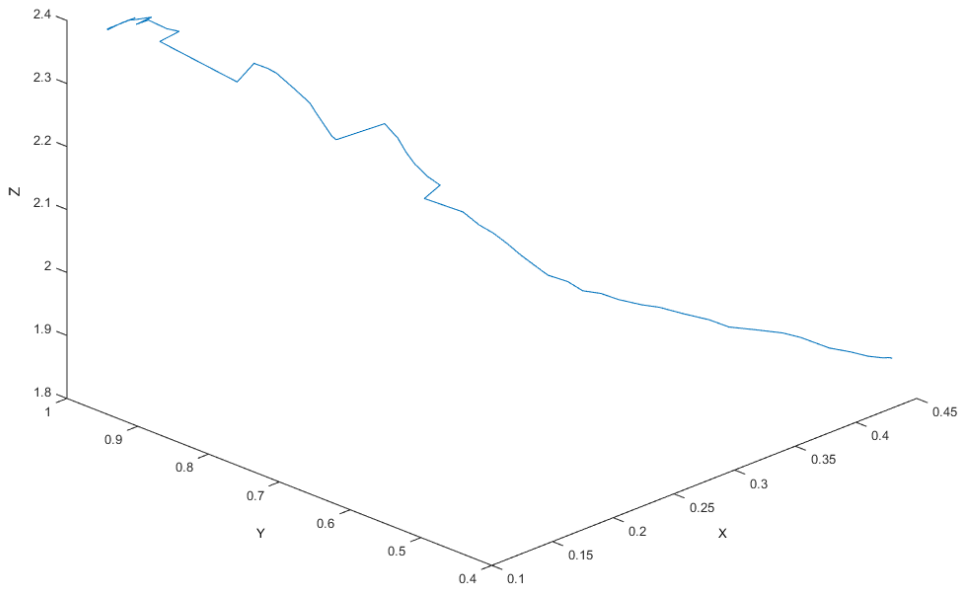
Dünya koor. sistemine göre K2 koordinatları



Şekil 5.18 Kullanıcı 12. ekleminin ortam referans sistemine göre 2. Kinect koordinatları

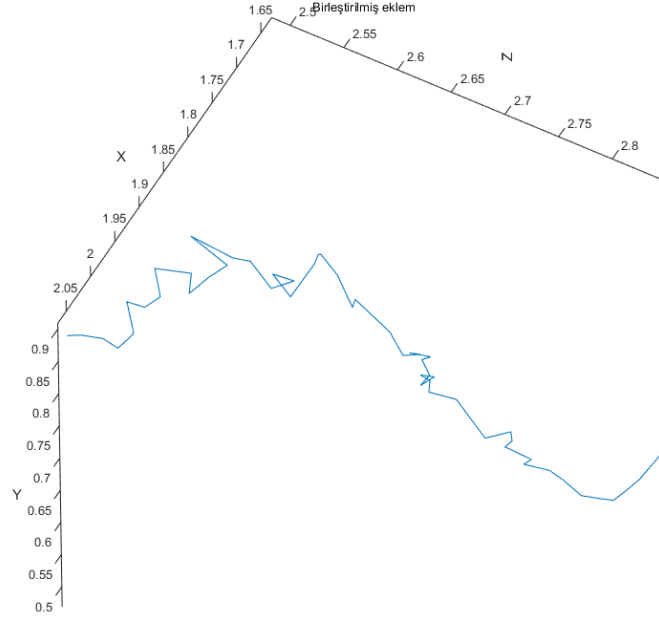
Şekil 5.19'da 12. eklemin ikinci Kinect'ten alınan verileri ikinci Kinect'in referans sistemine göre gösterilmektedir.

2. Kinect koordinat sistemine göre K2 koordinatları



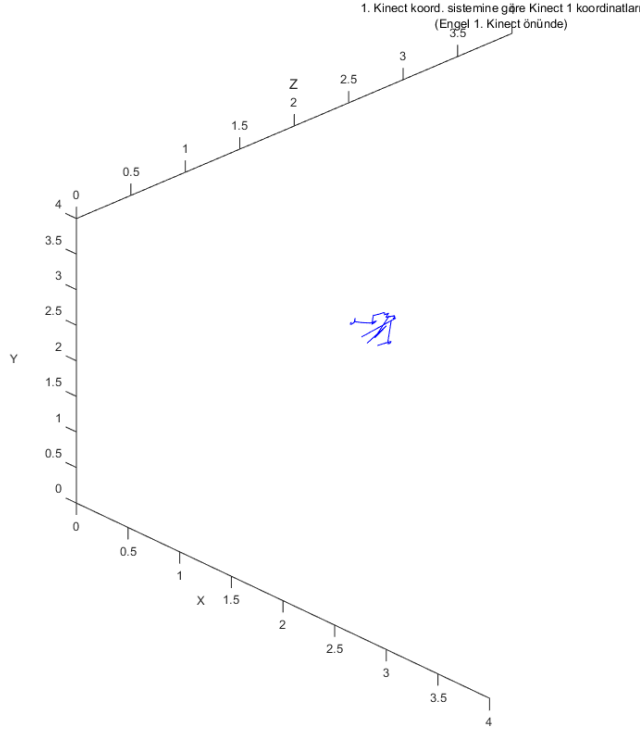
Şekil 5.19 Kullanıcı 12. ekleminin ikinci Kinect'e göre 2. Kinect koordinatları

Kullanıcının duruşu ikinci Kinect' in Y ekseninde 45 derece dönük olduğundan grafikteki yörünge oluşmaktadır. Şekil 5.20'de ise 12. eklem birleştirilmiş koordinatları ortamın koordinat sistemine göre gösterilmektedir.



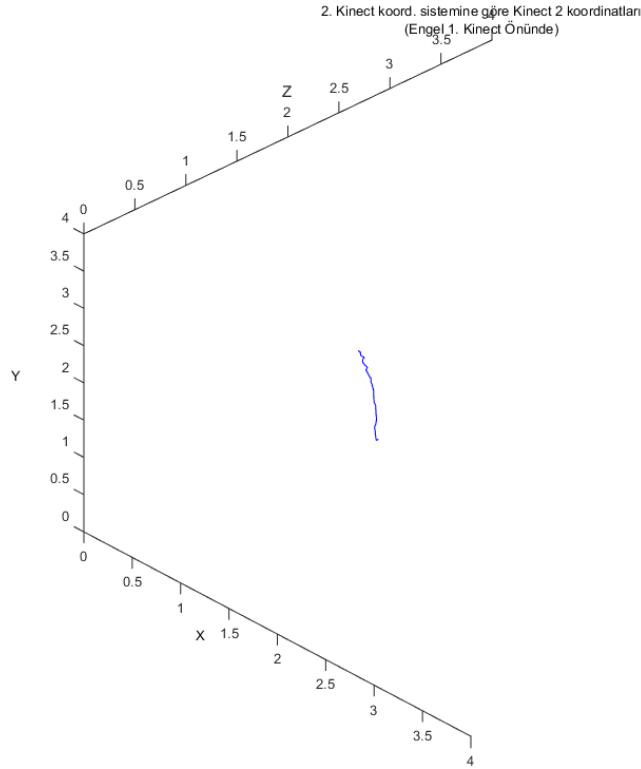
Şekil 5.20 Karşıya dönük kullanıcı 12. eklem birleştirilmiş koordinatları

Bu grafikte bazı noktalar düzensiz olmakla birlikte istenen yörünge nitel olarak oluşturulabildiği görülmektedir. Kameralardan birinin önüne engel konulduğu zaman parçacık filtresi algoritmasının nasıl çalıştığını test etmek için kullanıcı karşıya dönük bir şekilde konumlandırılmış ve birinci Kinect' in önüne engel konulmuştur. Bu şekilde elde edilen hareket kaydı 43 görüntü karesi uzunluğundadır. Şekil 5.21' de 12. eklem birinci Kinect'ten alınan verileri gösterilmektedir.



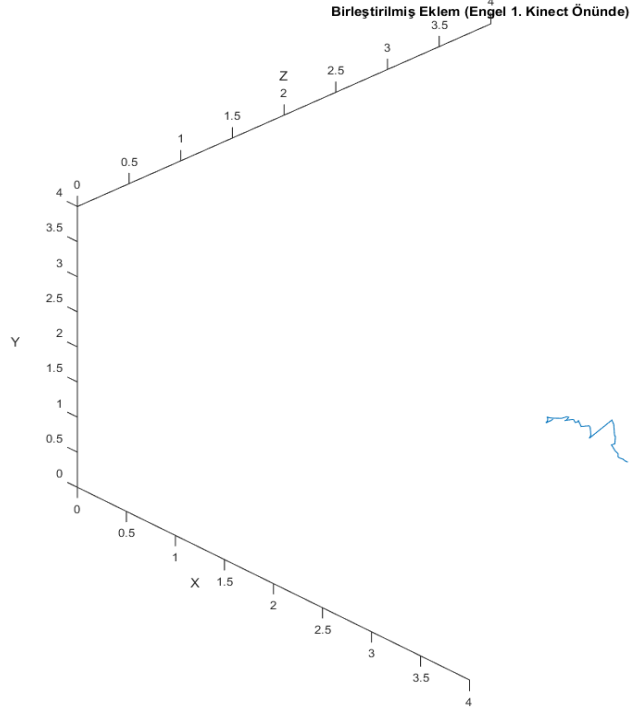
Şekil 5.21 Engel birinci Kinect önüdeyken kullanıcı 12. ekleminin birinci Kinect verileri

Birinci Kinect önündeki engelin kullanıcı silüetini bozmasının bu grafikteki yörüngeye neden olduğu anlaşılmaktadır. Şekil 5.22’de 12. eklemin ikinci Kinect’ten alınan verileri ikinci Kinect’in referans sistemine göre gösterilmektedir.



Şekil 5.22 Engel birinci Kinect önüdeyken kullanıcı 12. ekleminin ikinci Kinect verileri

Önünde engel olmamasına karşın kullanıcının duruşu ikinci Kinect'in *Y* eksenini etrafında 45 derece dönmüş olduğundan grafikteki yörünge oluşmaktadır. Şekil 5.23'de ise 12. eklemin birleştirilmiş koordinatları ortamın koordinat sistemine göre gösterilmektedir.



Şekil 5.23 Karşıya dönük kullanıcı 12. ekleminin birleştirilmiş koordinatları

6. SONUÇLAR VE DEĞERLENDİRME

İnsan hareketlerinin analitik biçimde değerlendirilmesi, bilgisayarlı görme, robotik ve insan-bilgisayar etkileşimi alanlarındaki çeşitli uygulamalarda kullanılmaktadır. Hareket yakalama sürecinde de, insan hareketlerini tanımlayan özellikler çıkarılarak hareketlerin analitik biçimde değerlendirilmesinde kullanılacak veriler elde edilmektedir. Xbox 360 oyun konsollarında bir doğal kullanıcı arabirimi (natural user interface) olarak kullanılmak üzere Microsoft tarafından geliştirilen Kinect; ortamın derinlik görüntülerinden kullanıcı iskeletini elde edebilmesi sayesinde işaretçisiz hareket yakalama amacıyla da kullanılabilir.

Aktivite tanımlama, şekil sınıflandırma ve insansı robotlar için yol bulma gibi uygulamalarda hareket benzerliğinin ölçülebilmesi önemlidir. Hareket benzerliği bir eğri benzerliği problemi olarak ele alındığında, literatürde çok daha geniş bir alanda yapılmış kapsamlı çalışmalara rastlanılmaktadır. Literatürde önerilen yöntemler (DTW, Hausdorff uzaklığı, Fréchet uzaklığı vs.) tez kapsamında incelenen Kinect kullanılarak oluşturulmuş 3B hareket yörüngelerini karşılaştırma problemi için uygunluğuna göre ele alındığında, Fréchet uzaklığının nitel değerlendirmelere uygun benzerlik ölçütünü karmaşık durumlar için önerilen yöntemlere nazaran daha az işlem zamanında sağladığı görülmektedir.

Uygulamada Kinect kamera sisteminin Kinect for Windows modeli kullanılmış olup yakalanan kullanıcı hareketlerinden oluşturulan yörüngelerin ayın referans sisteminde (Kinect koordinat sistemi) yer alması sağlanmıştır. Uzaklık değeri mutlak bir sayı olduğundan sıfırdan büyük sayılar Fréchet uzaklığı için geçerlidir ve bu değer sıfıra yaklaştıkça yörünge benzerliği artmaktadır. İki yörünge arasındaki Fréchet uzaklığının sıfır olması, yörüngelerin %100 benzediği anlamına gelmektedir. Elde edilen deneysel sonuçlarda referans hareketine en benzer hareket olan birinci hareketin bütün eklemlerinin XY izdüşümü ve XZ izdüşümü karşılaştırmaları için Fréchet uzaklığı en düşük olarak ölçülmüştür. Eklem bazlı XY izdüşümü ve XZ izdüşümü karşılaştırmaların ortalamaları sırasıyla 0,06325, 0,30075, 0,2495 ve 0,0555, 0,2765, 0,27525 olmaktadır. Elde edilen bu sonuç nitel değerlendirmeye uygun olmakta ve Fréchet mesafesinin hareket yörüngelerinin benzerliğinde kullanılabilir olduğunu göstermektedir.

Kinect hakkında yapılan akademik çalışmalarda; nesnelerin birbirini örttüğü (occlusion) durumlarda kesintisiz iskelet verileri akışı, nesnelerin 3B taranması ve

modellenmesi için daha detaylı yüzey verisi eldesi vb. çoklu Kinect kullanımı üzerine uygulamalar ortaya konmaktadır. Söz konusu çalışmalarda uygulamanın niteliğine göre kızılötesi ışınların girişiminin etkisi ve bu olumsuz etkiyi zayıflatmak için öneriler de sunulmaktadır. Çoklu Kinect sensörleri kullanılarak hareket yakalama üzerine yapılan akademik çalışmalar, genelde Kinect'lerin iskelet verilerini kullanma veya duruş tahmini için siluet veya şablon eşleştirme optimizasyonu yaklaşımını ele almaktadır.

Bu çalışmada literatürdeki diğer uygulamalardan farklı olarak, çok sayıda Kinect'ten elde edilen verilerin ana Kinect olarak belirlenen bir Kinect'in referans sistemine taşınması yerine Kinect'lerin parçacık filtresi algoritmasına ölçüm verileri sağlayan birer sensör olarak kullanılması önerilmektedir. Ölçüm verilerini oluşturan eklem koordinatları Kinect koordinat sistemine göre konumlandırıldığından, her iki Kinect koordinat sisteminin birleşmiş eklem koordinatlarının yer aldığı referans sistemine taşınması için geometrik dönüşüm matrisleri tanımlanmaktadır.

Uygulamada sırasıyla öz kaynaklı örtme, her iki Kinect' in kullanıcıyı kolaylıkla algıladığı vücut duruşu ve birinci Kinect'in önüne konan engelin neden olduğu örtme durumları ele alınmıştır.

Elde edilen deneysel sonuçlar parçacık filtresi algoritmasının çoklu Kinect'lerden elde edilen iskelet verilerini birleştirmede kullanılabilmesini göstermektedir. Ayrıca bu sonuçlar öz kaynaklı örtme ve engel kaynaklı örtme durumlarında (yörüngede yer yer düzensizlikler ve gürültüler olsa da) benzerlik ölçümü için hareket karşılaştırmada kullanılacak yörüngeler elde edilebileceğini ortaya koymaktadır.

Öz kaynaklı örtme ve engelin neden olduğu örtme durumlarında birleşmiş eklem yörüngesindeki düzensizlikler, kameralardan birinin sağlıklı veri almasından kaynaklanmakta olup algılamada tanımlanabilecek ilave iyileştirmelerle çözülebileceği düşünülmektedir.

KAYNAKÇA

- [1] M. Field, D. Stirling, F. Naghdy, and Z. Pan, "Motion capture in robotics review," in *IEEE International Conference on Control and Automation, 2009. ICCA 2009*, 2009, pp. 1697–1702.
- [2] X. Chen and M. Koskela, "Skeleton-based action recognition with extreme learning machines," *Neurocomputing*, vol. 149, Part A, pp. 387–396, Feb. 2015.
- [3] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholz, and M. Magnor, "Markerless Motion Capture using multiple Color-Depth Sensors."
- [4] Y. Liu, C. Stoll, J. Gall, H. P. Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1249–1256.
- [5] M. Gleicher, "Animation from Observation: Motion Capture and Motion Editing," *SIGGRAPH Comput Graph*, vol. 33, no. 4, pp. 51–54, Nov. 1999.
- [6] Anonymous, "Facilities," *House of Moves*, 13-Jun-2011. [Online]. Available: <http://www.moves.com/facilities-technology/>. [Accessed: 19-Jun-2016].
- [7] Anonymous, "What is motion capture," *VICON*. [Online]. Available: <http://www.vicon.com/what-is-motion-capture>. [Accessed: 19-Jun-2016].
- [8] A. Mazzoldi, D. De Rossi, F. Lorussi, E. P. Scilingo, and R. Paradiso, "Smart textiles for wearable motion capture systems," *AUTEX Res. J.*, vol. 2, no. 4, pp. 199–203, 2002.
- [9] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 755–762.
- [10] M. Grzegorzec, C. Theobalt, R. Koch, and A. Kolb, Eds., *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, vol. 8200. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [11] T. Brox, B. Rosenhahn, and D. Cremers, "Contours, optic flow, and prior knowledge: cues for capturing 3D human motion in videos," in *Human Motion - Understanding, Modeling, Capture, and Animation*, Springer, 2007.
- [12] L. Mündermann, S. Corazza, and T. P. Andriacchi, "The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications," *J. NeuroEngineering Rehabil.*, vol. 3, p. 6, Mar. 2006.
- [13] S. Corazza, L. Mündermann, and T. Andriacchi, "A framework for the functional identification of joint centers using markerless motion capture, validation for the hip joint," *J. Biomech.*, vol. 40, no. 15, pp. 3510–3515, 2007.
- [14] N. Hidayah Bt Ismail and S. N. B. Basah, "The applications of Microsoft Kinect for human motion capture and analysis : A review," in *2015 2nd International Conference on Biomedical Engineering (ICoBE)*, 2015, pp. 1–4.
- [15] Anonymous, "Kinect for Windows Sensor Components and Specifications," *Kinect for Windows Sensor Components and Specifications*. [Online]. Available: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>. [Accessed: 23-Jun-2016].
- [16] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced Computer Vision With Microsoft Kinect Sensor: A Review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.
- [17] Anonymous, *Kinect for Windows Human Interface Guidelines v1.8*. Microsoft.

- [18] A. Jana, *Kinect for Windows SDK programming guide: build motion-sensing applications with Microsoft's Kinect for Windows SDK quickly and easily*. Birmingham: Packt Publ, 2012.
- [19] Anonymous, "Kinect for Windows Architecture," *Kinect for Windows Architecture*. [Online]. Available: <https://msdn.microsoft.com/en-us/library/jj131023.aspx>. [Accessed: 23-Jun-2016].
- [20] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time Human Pose Recognition in Parts from Single Depth Images," *Commun ACM*, vol. 56, no. 1, pp. 116–124, Jan. 2013.
- [21] M. Stommel, M. Beetz, and W. Xu, "Model-Free Detection, Encoding, Retrieval, and Visualization of Human Poses From Kinect Data," *IEEEASME Trans. Mechatron.*, vol. 20, no. 2, pp. 865–875, Apr. 2015.
- [22] "Joint Orientation." [Online]. Available: <https://msdn.microsoft.com/en-us/library/hh973073.aspx>. [Accessed: 21-Jun-2016].
- [23] S. W. Bailey and B. Bodenheimer, "A Comparison of Motion Capture Data Recorded from a Vicon System and a Microsoft Kinect Sensor," in *Proceedings of the ACM Symposium on Applied Perception*, New York, NY, USA, 2012, pp. 121–121.
- [24] B. Bonnechère, V. Sholukha, F. Moiseev, M. Rooze, and J. S. V. Sint, "From Kinect™ to anatomically-correct motion modelling: Preliminary results for human application," in *Games for Health*, B. Schouten, S. Fedtke, T. Bekker, M. Schijven, and A. Gekker, Eds. Springer Fachmedien Wiesbaden, 2013, pp. 15–26.
- [25] A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, Eds., *Consumer Depth Cameras for Computer Vision*. London: Springer London, 2013.
- [26] A. Efrat, Q. Fan, and S. Venkatasubramanian, "Curve Matching, Time Warping, and Light Fields: New Algorithms for Computing Similarity between Curves," *J. Math. Imaging Vis.*, vol. 27, no. 3, pp. 203–216, Nov. 2006.
- [27] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.
- [28] D. J. Bemdt and J. Clifford, *Using Dynamic Time Warping to Find Patterns in Time Series*. 1994.
- [29] M. Müller, "Dynamic Time Warping," in *Information Retrieval for Music and Motion*, Springer Berlin Heidelberg, 2007, pp. 69–84.
- [30] S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici, "Gesture Recognition using Skeleton Data with Weighted Dynamic Time Warping," in *VISAPP (1)*, 2013, pp. 620–625.
- [31] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
- [32] A. A. Taha and A. Hanbury, "An Efficient Algorithm for Calculating the Exact Hausdorff Distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2153–2163, Nov. 2015.
- [33] M. J. Atallah, "A Linear Time Algorithm for the Hausdorff Distance Between Convex.pdf," *Inf. Process. Lett.*, vol. 17, no. 4, pp. 207–209, 1983.
- [34] D. P. Huttenlocher and W. J. Rucklidge, "Hausdorff-Based Matching," *Hausdorff-Based Matching*. [Online]. Available: <https://www.cs.cornell.edu/vision/hausdorff/hausmatch.html>. [Accessed: 22-Jun-2016].

- [35] H. Alt and M. Godau, "Computing the fréchet distance between two polygonal curves," *Int. J. Comput. Geom. Appl.*, vol. 5, no. 01n02, pp. 75–91, Mar. 1995.
- [36] A. F. Cook and C. Wenk, *Geodesic Fréchet distance with polygonal obstacles*. Department of Computer Science, University of Texas at San Antonio, 2008.
- [37] T. Eiter and H. Mannila, "Computing discrete Fréchet distance," Information Systems Department, Technical University of Vienna, Tech. Report CD-TR 94/64, 1994.
- [38] W. Rodriguez, M. Last, A. Kandel, and H. Bunke, "3-Dimensional curve similarity using string matching," *Robot. Auton. Syst.*, vol. 49, no. 3–4, pp. 165–172, Dec. 2004.
- [39] S. Li, P. N. Pathirana, and T. Caelli, "Multi-kinect skeleton fusion for physical rehabilitation monitoring," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 5060–5063.
- [40] S. Baek and M. Kim, "Dance Experience System Using Multiple Kinects," *Int. J. Future Comput. Commun.*, vol. 4, no. 1, pp. 45–49, Feb. 2015.
- [41] A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis, "Dance analysis using multiple Kinect sensors," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014, vol. 2, pp. 789–795.
- [42] S. Asteriadis, A. Chatzitofis, D. Zarpalas, D. S. Alexiadis, and P. Daras, "Estimating Human Motion from Multiple Kinect Sensors," in *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, New York, NY, USA, 2013, p. 3:1–3:6.
- [43] K.-Y. Yeung, T.-H. Kwok, and C. C. L. Wang, "Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications," *J. Comput. Inf. Sci. Eng.*, vol. 13, no. 4, pp. 041007–041007, Oct. 2013.
- [44] B. Williamson, J. LaViola, T. Roberts, and P. Garrity, "Multi-kinect tracking for dismounted soldier training," in *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (IITSEC)*, 2012, pp. 1727–1735.
- [45] L. Zhang, J. Sturm, D. Cremers, and D. Lee, "Real-time human motion tracking using multiple depth cameras," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2389–2395.
- [46] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake’N’Sense: Reducing Interference for Overlapping Structured Light Depth Cameras," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2012, pp. 1933–1936.
- [47] B. Kainz, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter, and D. Schmalstieg, "OmniKinect: Real-time Dense Volumetric Data Acquisition and Applications," in *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*, New York, NY, USA, 2012, pp. 25–32.
- [48] M. Liggins II, D. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition*. CRC Press, 2008.
- [49] S. Thrun, "Particle Filters in Robotics," in *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 2002, pp. 511–518.

[50] M. Azimi, "Skeletal Joint Smoothing White Paper." [Online]. Available: <https://msdn.microsoft.com/en-us/library/jj131429.aspx>. [Accessed: 23-Jun-2016].

TEZDEN ÜRETİLEN YAYIN VE SUNUMLAR

M. O. Şen ve M. Ertürkler, “Kinect ile Elde Edilen Hareket Yakalama Verilerinin Oluşturduğu Yörüngelerin Karşılaştırılması” IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU 2015) , Mayıs 2015, s. 2541-2544.

ÖZGEÇMİŞ

Adı Soyadı: Mehmed Oğuz ŞEN

Doğum Yeri ve Tarihi: Erzurum, 23.11.1985

Adres: İnönü Üniversitesi Merkez Yerleşkesi Batı Lojmanları B1 Blok No:5 44000
MALATYA

E-Posta: oguz.sen@inonu.edu.tr

Lisans: Orta Doğu Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü

Yayın Listesi: M. O. Şen ve M. Ertürkler, “Kinect ile Elde Edilen Hareket Yakalama Verilerinin Oluşturduğu Yörüngelerin Karşılaştırılması” IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU 2015) , Mayıs 2015, s. 2541-2544.