

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

WEB TABANLI CBS, SOKAK GÖRÜNÜMÜ VE SANAL GERÇEKLIK

TOLGA BAKIRMAN

**YÜKSEK LİSANS TEZİ
HARİTA MÜHENDİSLİĞİ ANABİLİM DALI
UZAKTAN ALGILAMA VE CBS PROGRAMI**

**DANIŞMAN
DOÇ. DR. M. ÜMİT GÜMÜŞAY**

İSTANBUL, 2013

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

WEB TABANLI CBS, SOKAK GÖRÜNÜMÜ VE SANAL GERÇEKLİK

Tolga BAKIRMAN tarafından hazırlanan tez çalışması 13.06.2013 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Harita Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

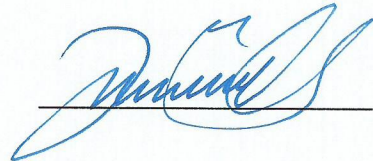
Doç. Dr. Ümit GÜMÜŞAY
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Ümit GÜMÜŞAY
Yıldız Teknik Üniversitesi

Prof. Dr. Fatmagül KILIÇ
Yıldız Teknik Üniversitesi

Doç. Dr. Şinasi KAYA
İstanbul Teknik Üniversitesi



ÖNSÖZ

“Sensör Yardımı ile 3 Boyutlu Algılama Sisteminin Oluşturulması” konulu tezimin danışmanlığını üstlenen, bana maddi ve manevi her türlü destek olan, fikirlerime hep olumlu ve yapıcı bakış açısıyla bakan hocam Doç.Dr. M. Ümit Gümüşay’a saygılarımla teşekkürlerimi sunarım. Panorama görüntülerini elde etmemi sağlayarak, çalışmanın gerçekleşmesinde büyük rol oynayan Prof. Dr. Fatmagül Kılıç’a ve Başarsoft firmasının sahibi Sayın Alim Küçükpehlivan’a teşekkür ederim.

Ayrıca, desteklerini sürekli yanımda hissettiğim aileme, bu çalışmaya başlamamda ve gerçekleştirmemde büyük emekleri olan arkadaşlarım Ömer Harmanşa ve Büşra Demiral’a teşekkürlerimi sunarım.

Mayıs, 2013

Tolga BAKIRMAN

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	vii
KISALTMA LİSTESİ.....	viii
ŞEKİL LİSTESİ.....	x
ÇİZELGE LİSTESİ	xiii
ÖZET	xiv
ABSTRACT.....	xv
BÖLÜM 1	
GİRİŞ.....	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	2
1.3 Bulgular	2
BÖLÜM 2	
İNTERNET TABANLI COĞRAFİ BİLGİ SİSTEMLERİ	3
3.1 İnternet Tabanlı CBS Yöntemleri.....	3
3.1.1 Sunucu Tarafı İnternet Tabanlı CBS	3
3.1.2 İstemci Tarafı İnternet Tabanlı CBS	4
3.2 ArcGIS for Server.....	5
3.2.1 Çalışma Prensipleri.....	5
3.2.2 Web Servisleri	6
3.2.3 Web Servis Fonksiyonları.....	6
3.2.3.1 Harita Servisleri	6
3.2.3.2 Veri Servisleri	6
3.2.3.3 Analitik (Çözüm) Servisler	7
3.2.4 Web Servisi Türleri.....	7
3.2.4.1 SOAP Tabanlı Web Servisleri	8
3.2.4.2 REST Tabanlı Web Servisleri	9

3.2.4.3 SOAP ve REST Tabanlı Servislerin Karşılaştırılması.....	12
3.3 Web Servislerini Kullanma (API)	13
3.3.1 ArcGIS API for JavaScript	13
3.3.2 ArcGIS API for Flex	13
3.3.3 ArcGIS API for Silverlight	14
BÖLÜM 3	
SOKAK GÖRÜNÜMÜ.....	17
3.1 Sokak Görünümünün Oluşumu.....	17
3.2 Sokak Görünümü Nedir? Nasıl Çalışır?	19
3.2.1 Panoramaların Oluşumu.....	19
3.2.1.1 Hata Kaynakları	23
3.2.2 Projeksiyonlar	25
3.2.2.1 Küresel (Equirectangular – Plate Carrée) Projeksiyon	25
3.2.2.2 Silindirik Projeksiyon	27
3.2.2.3 Kubbe Projeksiyon.....	27
3.2.2.4 Kübik Projeksiyon	29
3.2.3 Panoramik Döşemeler (Tile)	30
3.2.4 Yasal Boyut	31
3.3 Google Maps Javascript API v3 ve Street View Servisi.....	32
BÖLÜM 4	
HAREKET SENSÖRÜ VE DERİNLİK HARİTALARINDAN HAREKETİN ALGILANMASI.....	34
4.1 Microsoft Kinect.....	34
4.2 Derinlik Haritaları	35
4.3 İskelet Çıkarımı.....	39
4.4 Hareket Tanımlama.....	40
BÖLÜM 5	
UYGULAMA	43
5.1 Uygulama Alanı	43
5.2 Veri Toplama	44
5.3 Görüntülerin İşlenmesi	46
5.4 Sokak Görünümünün Oluşturulması.....	48
5.5 Sokak Görünümü Analizi	59
5.6 İnternet Tabanlı Coğrafi Bilgi Sisteminin Oluşturulması	61
6.6.1 Verinin Hazırlanması.....	61
6.6.2 Verinin Servis Edilmesi.....	63
6.6.3 Web Uygulamasının Oluşturulması	66
5.7 Sokak Görünümünün Vücut Hareketleriyle Kontrol Edilmesi.....	70
BÖLÜM 6	
SONUÇ VE ÖNERİLER	80
KAYNAKLAR	82

EK-A

İSİM DEĞİŞTİRME PROGRAMI C# KODLARI 85

EK-B

SORGULAMA KODLARI..... 90

ÖZGEÇMİŞ..... 93

SİMGE LİSTESİ

K	Kamera sabiti
R	Yarıçap
X	Küresel koordinat sisteminde noktanın X doğrultusundaki koordinatı
Y	Küresel koordinat sisteminde noktanın Y doğrultusundaki koordinatı
Z	Küresel koordinat sisteminde noktanın Z doğrultusundaki koordinatı
Z	Kameradan objeye olan mesafe
Z_c	Varsayılan mesafe
θ	Düzleme indirgenmiş yatay koordinat
φ	Düzleme indirgenmiş düşey koordinat

KISALTMA LİSTESİ

ADF	Application Development Framework
API	Application Programming Interface
BMP	Bitmap Image File
CBS	Coğrafi Bilgi Sistemleri
CM	Santimetre
ED50	European Datum 1950
ESRI	Enviromental Systems Research Institute
EXIF	Exchangable Image File Format
FAAST	Flexible Action and Articulated Toolkit
FPS	Frame Per Second
GB	Gigabayt
GIF	Graphics Interchange Format
GPS	Global Positionaing System
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identification
IDE	Integrated Development Enviroment
IP	Internet Protocol Address
İBB	İstanbul Büyükşehir Belediyesi
JPEG	Joint Photographic Experts Group
KB	Kilobayt
MB	Megabayt
MS	Milisaniye
MYO	Meslek Yüksek Okulu
ÖSYM	Öğrenci Seçme ve Yerleştirme Merkezi
PNG	Portable Network Graphics
REST	Representational State Transfer
RGB	Red Green Blue
S	Saniye
Sd	Service Definition
SDK	Software Development Kit
SOAP	Simple Object Acces Protocol
SWF	Small Web Format

TIFF	Tagged Image File Format
URL	Uniform Resource Locator
UTM	Universal Transverse Mercator
WSDL	Web Service Description Language
WWW	World Wide Web
XML	Extensible Markup Language
XAML	Extensible Applications Markup Language
YTÜ	Yıldız Teknik Üniversitesi

ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1	Sunucu taraflı internet tabanlı CBS mimarisi 4
Şekil 2. 2	İstemci taraflı internet tabanlı CBS mimarisi 4
Şekil 2. 3	Web tabanlı CBS'nin çalışma mimarisi 5
Şekil 2. 4	ArcGIS Server yazılımının işlevleri 6
Şekil 2. 5	SOAP tabanlı web servislerinin mimarisi 8
Şekil 2. 6	SOAP tabanlı web servisinin yanıtı 8
Şekil 2. 7	REST tabanlı web servislerinin mimarisi 9
Şekil 2. 8	ArcGIS Server yazılımının REST arayüzü 9
Şekil 2. 9	REST tabanlı web servisinin yanıtı 10
Şekil 2. 10	REST tabanlı web servisinin katman isteği yanıtı 10
Şekil 2. 11	ArcGIS Server REST API ile gerçekleştirilmiş sorgu yanıtı 11
Şekil 2. 12	ArcGIS Server yazılımında SOAP ve REST tabanlı web servisleri 12
Şekil 2. 13	ArcGIS API for Flex mimarisi 14
Şekil 2. 14	ArcGIS API for Silverlight mimarisi 15
Şekil 3. 1	Google haritaların ilk kullanım arayüzü 17
Şekil 3. 2	Sokak görünümü kapsama alanı 18
Şekil 3. 3	Kişisel panorama ile oluşturulmuş sokak görünümü 18
Şekil 3. 4	Küresel projeksiyona izdüşürülmüş 360 derece panoramik görüntü 19
Şekil 3. 5	Ladybug 3 ve Ladybug 2 kameraları 19
Şekil 3. 6	Görüntü birleştirme işlemi şeması 20
Şekil 3. 7	RGB görüntüler 20
Şekil 3. 8	Renk işleme yöntemine göre sonuçlar 21
Şekil 3. 9	Renk işleme yöntemlerinin süreye ve donanıma göre karşılaştırılması 21
Şekil 3. 10	Rektifiye edilmiş görüntüler 22
Şekil 3. 11	2 boyutlu mesh, 3 boyutlu mesh 22
Şekil 3. 12	Çift görüntü hatası 23
Şekil 3. 13	Kaynaşma genişliği = 0 durumu 24
Şekil 3. 14	Kaynaşma genişliği = 100 durumu 24
Şekil 3. 15	Parlaklık düzeltmesi kapalı 25
Şekil 3. 16	Parlaklık düzeltmesi açık 25
Şekil 3. 17	Küresel projeksiyon geometrisi 26
Şekil 3. 18	Küresel projeksiyona indirgenmiş panoramik görüntü 26
Şekil 3. 19	Silindirik projeksiyon geometrisi 27

Şekil 3. 20	Silindirik projeksiyona indirgenmiş panoramik görüntü.....	27
Şekil 3. 21	Kubbe projeksiyon geometrisi	28
Şekil 3. 22	Kubbe projeksiyona indirgenmiş panoramik görüntü	28
Şekil 3. 23	Kübik projeksiyon geometrisi.....	29
Şekil 3. 24	Kübik projeksiyona indirgenmiş panoramik görüntü.....	30
Şekil 3. 25	Görüntü parçaları.....	31
Şekil 3. 26	Bulanıklaştırma işleminin uygulandığı bölgeler	32
Şekil 4. 1	Kinect Sports oyununun oynanışı	35
Şekil 4. 2	Microsoft Kinect.....	35
Şekil 4. 3	Derinlik haritası	36
Şekil 4. 4	Derinlik ölçüm teknikleri	36
Şekil 4. 5	Kızılötesi projektör kesiti , Paralaks oluşum, Kızılötesi görüntü	37
Şekil 4. 6	Derinlik haritası	38
Şekil 4. 7	İskelet.....	39
Şekil 4. 8	Vücut segmentasyonu	39
Şekil 4. 9	Kullanılan karar ağacı sayısının vücut segmentasyonuna etkisi	40
Şekil 4. 10	Üç boyutlu vücut birleşim noktaları.....	40
Şekil 4. 11	Gövde, 1. ve 2. derece bağlantı noktaları	41
Şekil 4. 12	Gövde, 1. ve 2. derece bağlantı noktalarının koordinat sisteminde tanımlanması.....	41
Şekil 4. 13	Hareket grafiği.....	42
Şekil 5. 1	Yıldız Teknik Üniversitesi Davutpaşa Kampüsü.....	43
Şekil 5. 2	Fotoğraf alım yolları	45
Şekil 5. 3	100 Kimlik numaralı görüntü parçaları	49
Şekil 5. 4	İsim değiştirme programı arayüzü	50
Şekil 5. 5	100 Kimlik numaralı görüntü parçaları son hali	50
Şekil 5. 6	Html dosyasının dil, karakter ve stil seti belirlenmesi	51
Şekil 5. 7	Referanslar	51
Şekil 5. 8	Main.css dosyası	51
Şekil 5. 9	Sayfa tasarımı.....	52
Şekil 5. 10	Sayfa görünümü	52
Şekil 5. 11	Sokak görünümü sağlayıcısı ayarları	53
Şekil 5. 12	Sokak görünümü objesinin oluşturulması	53
Şekil 5. 13	Bağlantı objesinin oluşturulması.....	53
Şekil 5. 14	Bütün görüntü fonksiyonları	54
Şekil 5. 15	Parçalı görüntü fonksiyonları	54
Şekil 5. 16	Switch – Case fonksiyonu ile durumların tanımlanması	56
Şekil 5. 17	Fotoğrafları arasındaki geçiş bağlantılarını ekleme	57
Şekil 5. 18	Heading özelliğinin değer aralığı.....	58
Şekil 5. 19	Sayfanın son durumu	58
Şekil 5. 20	Sayılaştırılan noktalar	62
Şekil 5. 21	Sokak Görünümü başlangıç noktaları öznitelik tablosu.....	62
Şekil 5. 22	Service Definition (.sd) dosyası oluşturma arayüzü	63
Şekil 5. 23	ArcGIS Server Manager giriş sayfası.....	64
Şekil 5. 24	Oluşturulan servisin REST ve SOAP adresleri	64

Şekil 5. 25	Oluşturulan servisin REST yanıtı.....	65
Şekil 5. 26	Oluşturulan servisin SOAP yanıtı.....	65
Şekil 5. 27	Referanslar	66
Şekil 5. 28	XAML referansları	66
Şekil 5. 29	Grid kaynağı XAML ve görünümü (ağda)	67
Şekil 5. 30	Harita tepsisi	67
Şekil 5. 31	Servisin detayları (REST arayüzü).....	68
Şekil 5. 32	Sorgu sonucu.....	68
Şekil 5. 33	Katman listesi tasarımı ve görünümü	69
Şekil 5. 34	Harita kontrolü ve görünümü	69
Şekil 5. 35	Oluşturulan Web tabanlı CBS'nin kullanıcı arayüzü.....	70
Şekil 5. 36	FAAST arayüzünde iskelet görünümü	72
Şekil 5. 37	Hareketlerin XML'de görünümü	75
Şekil 5. 38	İleri hareketi XML dosyası	76
Şekil 5. 39	Sağa dönme hareketi FAAST, RGB, derinlik, üç boyutlu	77
Şekil 5. 40	Sokak görünümünde sağa dönme	78
Şekil 5. 41	Sağa döndürme, sola döndürme, aşağı döndürme, yukarı döndürme, sonraki görüntü, önceki görüntü	79

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2. 1	ArcGIS Server REST API ile dışa aktarma sorgusu 11
Çizelge 2. 2	ArcGIS API for Silverlight Kütüphanesi 16
Çizelge 3. 1	Yakınlık seviyesine göre görüş alanları..... 31
Çizelge 5. 1	Ladybug 2 kamera özellikleri..... 44
Çizelge 5. 2	Lenslerin çekim anındaki odak uzaklıkları 45
Çizelge 5. 3	PGR dosyası içeriği..... 46
Çizelge 5. 4	Çözünürlük, renk işleme ve dosya türü karşılaştırılması 48
Çizelge 5. 5	Görüntülerin numaralandırılması 49
Çizelge 5. 6	Sokak görünümü internet adresleri 59
Çizelge 5. 7	Parçalı görüntü yöntemi internet adresleri 60
Çizelge 5. 8	Yükleme sürelerinin karşılaştırılması..... 61
Çizelge 5. 9	Algı noktaları 71
Çizelge 5. 10	Sokak görünümü klavye komutları..... 73
Çizelge 5. 11	Vücut hareketleri..... 74
Çizelge 5. 12	“input” tanımlayıcıları 75
Çizelge 5. 13	“output” tanımlayıcıları..... 75
Çizelge 5. 14	İleri ve geri “input” tanımlayıcıları 76
Çizelge 5. 15	İleri ve geri “Output” tanımlayıcıları 77

WEB TABANLI CBS, SOKAK GÖRÜNÜMÜ VE SANAL GERÇEKLİK

Tolga BAKIRMAN

Harita Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. M. Ümit GÜMÜŞAY

Google'ın 2007 yılında kullanıma sunduğu Sokak Görünümü servisi, yeryüzünde bir yer ve mekanı keşfetmek için ideal bir araçtır. Bu servis, kullanıcılara mekanlar hakkında hem bilgi vermekte, hem de sanal gerçeklik hissi sağlamaktadır. Maalesef bu servis, ülkemizde hizmet vermemektedir. Bunun yanında, Google'ın sağladığı Google Maps API for Javascript ile, kişisel panoramik görüntülerle, sokak görünümü sisteminin oluşturulması mümkündür. Bu çalışmayla, ülkemizde mevcut olmayan bu servis, Yıldız Teknik Üniversitesi Davutpaşa Kampüsü'nün tanıtımına katkı sağlayacak şekilde kullanıma sunulmuştur. Serviste farklı görüntü tipleri kullanılmış, bu görüntü tiplerinin servisin performansına etkisi irdelenmiştir. Sokak görünümünün kullanımlarını daha kolay bir hale getirmek ve kullanıcıya daha fazla bilgi sağlamak amacıyla, bu sistem internet tabanlı coğrafi bilgi sistemine entegre edilmiştir. Ayrıca, sokak görünümünün sanal gerçeklik boyutuna yeni bir anlam kazandırmak amacıyla, hareket sensörü kullanılarak, sokak görünümünün vücut hareketleriyle kontrolü de sağlanmıştır.

Anahtar Kelimeler: Sokak görünümü, panorama, internet tabanlı cbs, hareket sensörü, sanal gerçeklik

WEB BASED GIS, STREET VIEW AND VIRTUAL REALITY

Tolga BAKIRMAN

Department of Geomatics Engineering

MSc. Thesis

Advisor: Assoc. Prof. Dr. M. Ümit GÜMÜŞAY

Street View, which was released by Google in 2007, is an ideal tool to discover places and locations. This service doesn't only provide spatial information, but also a virtual reality environment for the user. Unfortunately, this service is not available in our country. But, in addition, Google enable users to create street view with custom panoramic image with the help of Google Maps API for Javascript. In this study, this service, which is not available in our country, was released in Yildiz Technical University Davutpasa Campus. This service was created using different type of images and effects of using different image types to Street view performance were examined. To make street view more user friendly and to provide more information to the user, street view was integrated with a web-based geographic information system. Additionally, to create a greater experience in virtual reality in street view, it was made possible to control street view with body motion using a motion sensor.

Keywords: Street view, panoramic image, web based gis, motion sensor, virtual reality

GİRİŞ

Gelişen teknolojiyle birlikte, dünyada haritalara ve konumsal bilgiye olan ihtiyaç artmaktadır. İnsanlar, bilgisayarlarından veya mobil aygıtlarından, bir mekan hakkında bilgi edinmek ve o mekanı keşfetmeyi istemektedirler. Sokak görünümü bu gereksinime farklı bir çözüm getirmektedir.

Sokak görünümü; panoramik fotoğraflardan oluşan ve kullanıcıya sanal gerçeklik ortamı sunan bir sistemdir. Bu sistem hareket sensörüyle kullanılarak, kullanıcılara bilgisayarla iletişim kurması için yeni bir yol sunulabilir. Ayrıca; sokak görünümü coğrafi bilgi sistemleri (CBS) ile bütünleştirilerek, daha kullanışlı ve daha çok bilgi edinilebilen bir sistem yaratılabilir.

1.1 Literatür Özeti

Coğrafi bilgi sistemleri; konuma ait verilerin depolanması, sunulması, görselleştirilmesi, sorgulanması ve analizlerin yapılmasında güçlü bir araçtır [1]. İnternet ve CBS'nin birleşiminden oluşan, internet tabanlı coğrafi bilgi sistemleri konuma ait bilgilerin sorgulanabilmesi için açık erişim sağlar [2]. Microsoft Silverlight ve Adobe Flash eklentileri veya HTML5 dili sayesinde, internet tabanlı cbs uygulamaları araçlar ve görsellik açısından büyük olanaklar sunmaktadır [3], [4].

Bilgisayarların ve fotoğraf makinelerinin dünyada yaygınlaşmasıyla birlikte, dijital fotoğrafların kullanımı artmıştır. Panoramik görüntüler bu kullanımın büyük bir parçasıdır [5]. Panoramik görüntülere dayanan servisler, sanal gerçeklik yarattığı için oldukça popüler olmaktadır [6].

Google Sokak görünümü, sokak seviyesinde çekilen 360 derece panoramik görüntülerden oluşan ve sıkça kullanılan sistemlerin başında gelir [7]. Bu sistemde, kullanıcıların fotoğraflar arasında geçiş yapabilme ve yatayda 360 derece, düşeyde 290 derece görüş sağlayabilme imkanına sahiptir [8].

1.2 Tezin Amacı

Bu çalışmada geniş bir mekana sahip olan Yıldız Teknik Üniversitesi Davutpaşa Kampüsü için bir sokak görünümü sisteminin oluşturulması, bu sistemin web harita servisleri kullanılarak CBS ile bütünleştirilmesi amaçlanmıştır. Oluşturulması istenen sistem, görüntülerin bütün ve parçalı olarak kullanıldığı iki farklı yöntemle uygulanması, aynı zamanda bu iki yöntemin sayfa yüklenme süresi açısından karşılaştırılmasının yapılması amaçlanmıştır. Sokak görünümünün sanal gerçeklik niteliğine yeni bir boyut kazandırılması için Microsoft Kinect for Xbox 360 hareket sensörü kullanılarak, sokak görünümünün vücut hareketleriyle kontrolü gerçekleştirilecektir. Microsoft Kinect, kullanıcılara oyun kumandası ya da objeye dokunmadan, oyunları interaktif olarak kontrol etme imkanı sunar [9]. Böylece, kullanıcının vücudu bir kontrol kumandasına dönüşerek, yeni bir bilgisayar insan etkileşim tecrübesi yaratmaktadır [10].

1.3 Bulgular

Bu yüksek lisans tez çalışmasında, sokak görünümünde kullanılan görüntü türlerinin, yüklenme sürelerine etkisi irdelenmiştir. Web tabanlı CBS, sokak görünümü ve hareket sensörü gibi üç farklı teknoloji ortak bir uygulamada kullanılarak, bu teknolojilerin daha farklı alanlarda da işlevselliği ortaya konmuştur.

Çalışma altı bölümden oluşmaktadır. Çalışmanın ikinci bölümünde, internet tabanlı CBS'nin çalışma prensibi ve kullanılan araçlardan detaylı olarak bilgi verilecektir. Çalışmanın üçüncü bölümünde sokak görünümü için kullanılan kamera ve yazılımlardan detaylı olarak doküman aktarılacaktır. Dördüncü bölümünde hareket sensörünün çalışma prensibi ve detayları açıklanacaktır. Beşinci bölümde çalışmada tasarlanan CBS, sokak görünümü ve vücut kontrol sisteminin oluşturulması için gerçekleştirilen uygulama çalışmaları ve kodlar sunulacaktır. Çalışmadan elde edilen bulgu ve sonuçlara, tezin son bölümünde yer verilecektir.

İNTERNET TABANLI COĞRAFİ BİLGİ SİSTEMLERİ

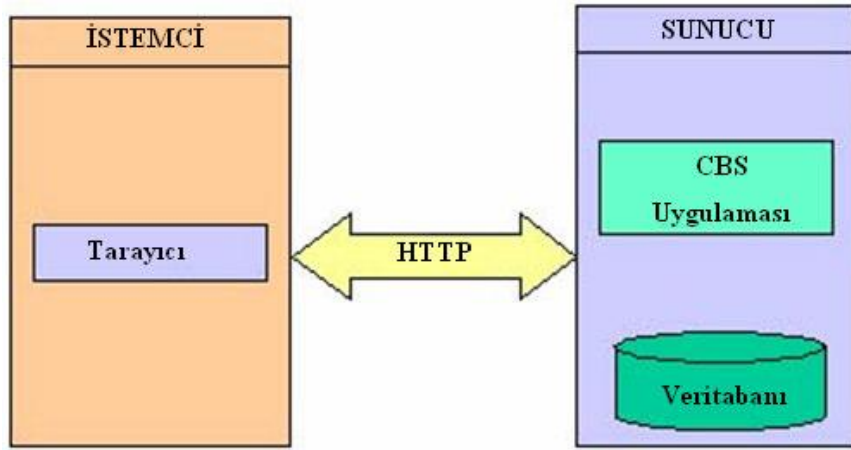
İnternet tabanlı CBS, konumsal veri paylaşmak, CBS analizleri yönetmek, paylaşılan veri ve diğer bilgilere ulaşmak için interneti kullanan ağ merkezli bir CBS aracıdır. İnternet tabanlı CBS istemci/sunucu yapısında, interaktif, paylaşımlı, dinamik ve çapraz platformlu bir sistemdir. Böylece, internet üzerinden birçok konumsal veriye ulaşılabilir ve sunucu/istemci arasındaki teknolojiler farklı olsa dahi veri paylaşımı sürdürülebilir [11], [12].

3.1 İnternet Tabanlı CBS Yöntemleri

İnternet tabanlı CBS uygulaması geliştirmek için sunucu taraflı ve istemci taraflı olmak üzere iki tip mimari vardır [13].

3.1.1 Sunucu Taraflı İnternet Tabanlı CBS

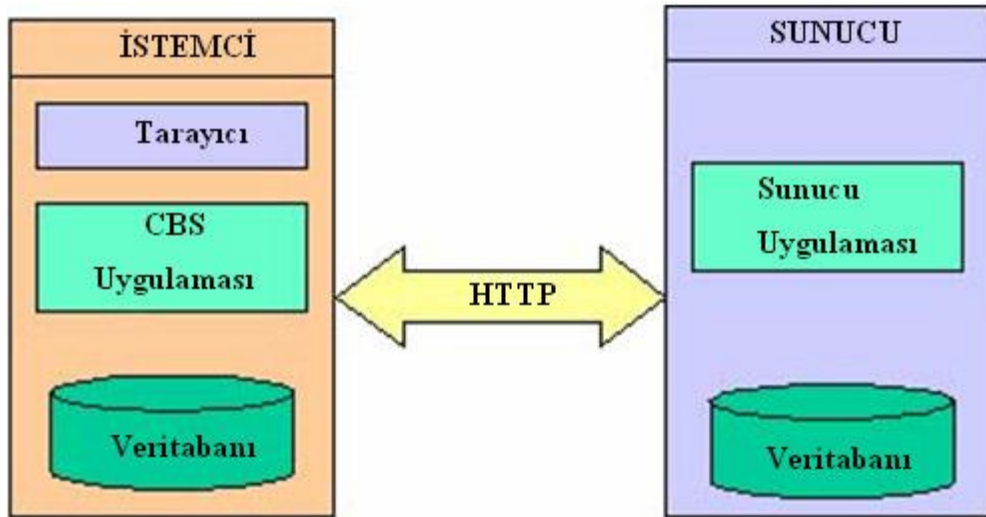
Sunucu taraflı İnternet tabanlı CBS, analizler yapmak ve sonuçlar elde etmek için, CBS sunucusuna bağlıdır. İnternet tarayıcısı bir tür kullanıcı arayüzü görevi görmektedir. Bu uygulamalarda, istemci sunucuya istek gönderir. Sunucu bu isteği işleyerek sonucu istemciye geri iletir. WWW ve HTML CBS vektör verileri desteklemediği için, internet tarayıcıları CBS yazılımlarıyla direkt olarak iletişim kuramazlar. Bu nedenle CBS sunucusuna ihtiyaç vardır [11], [14] (Şekil 2.1).



Şekil 2. 1 Sunucu taraflı internet tabanlı CBS mimarisi [14]

3.1.2 İstemci Taraflı Internet Tabanlı CBS

İstemci taraflı İnternet tabanlı CBS, CBS analizlerinin ve veri işleme işlemlerinin istemcinin yerel bilgisayarında yani internet tarayıcısında yapılmasına imkan sağlamaktadır. Bu yapıda CBS verileri ve analiz araçları sunucuda bulunmaktadır ve istek doğrultusunda istemci tarafından indirilmektedir. İstemci taraflı uygulamalar, Java uygulamaları, ActiveX kumandası veya Silverlight eklentisi kullanılarak oluşturulabilir [11]. İstemci taraflı İnternet tabanlı CBS'nin yapısı Şekil 2. 2 de görülebilir.



Şekil 2. 2 İstemci taraflı internet tabanlı CBS mimarisi [14]

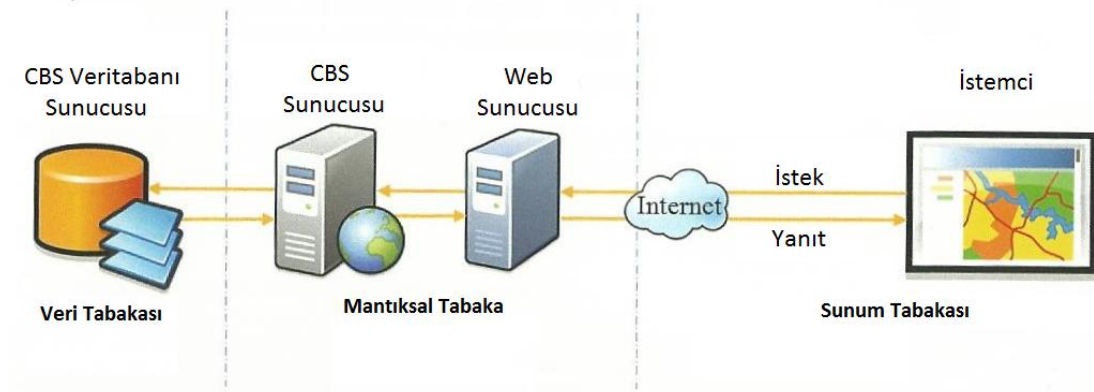
Bu çalışmada, Enviromental System Research Institute (ESRI) tarafından üretilen ArcGIS for Server 10.1 CBS sunucusu yazılımı kullanılmıştır. Bu bölümün diğer kısımlarında bu yazılımdan ayrıntılı olarak bahsedilecektir.

3.2 ArcGIS for Server

ArcGIS for Server, CBS veri kaynaklarını ve analiz fonksiyonlarına web tabanlı olarak kullanıma sunan ve coğrafi analizlerin yapılmasını ve analiz sonuçlarının görüntülenmesini sağlayan bir platform sağlar [15]. Bu platform sayesinde, paylaşılan veri kaynaklarına tüm dünyadan birçok kullanıcı tarafından hızlı bir şekilde ve standart bir arayüz ile erişim sağlanabilir [16],[17]. Bu erişim ile CBS kullanıcıları yerel bilgisayarlarına herhangi bir CBS yazılımı yüklemeyen, internet tarayıcıları üzerinden interaktif olarak CBS veriyle çalışabilmektedirler [18]. Bu sayede hem CBS projeleri hız kazanmış, hem de tekrarlı veri üretiminin önüne geçilmiştir[16].

3.2.1 Çalışma Prensipleri

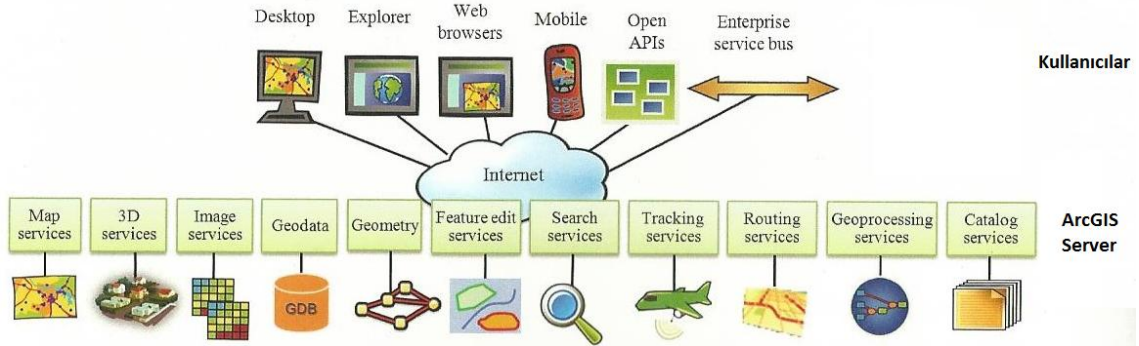
Web tabanlı CBS'nin çalışma prensibi, birkaç CBS bileşeniyle birlikte temel istemci/sunucu mimarisine dayanır (Şekil 2.3). Bir bilgisayar hem Web sunucusu hem CBS sunucusu hem de CBS veritabanı sunucusu olarak görevlendirilebilir. Fakat, mantıksal olarak bu üç bileşen üç ayrı aşamayı temsil etmektedir. Bu temel mimariye, istemci internet üzerinden http protokolü ile Web sunucusuna istek gönderir. CBS ile ilgili istekler, Web sunucusu tarafından CBS sunucusuna iletilir. CBS sunucusu gerekli veriyi CBS veritabanı sunucusundan çekerek, isteği işler. İstek sonucu, Web sunucusu aracılığıyla http protokolü ile istemciye iletilir ve sonuç istemcinin ekranında görüntülenir. Bu şekilde, istek ve yanıt döngüsü tamamlanmış olur. [17]



Şekil 2. 3 Web tabanlı CBS'nin çalışma mimarisi [17]

3.2.2 Web Servisleri

ArcGIS Server, CBS veri kaynaklarının paylaşımını Web servisleri aracılığıyla yapmaktadır (Şekil 2.4).



Şekil 2. 4 ArcGIS Server yazılımının işlevleri [17]

Web servisi, bir Web sunucusu üzerinde çalışan ve Web üzerindeki diğer programlara, yazılımsal arayüzünü sergileyen bir programdır. Web servisleri Web üzerinden diğer yazılım sistemleriyle bilgi alışverişi yapabilir ve fonksiyonlar paylaşabilirler. Bu servislere, kullanıcı, sistemindeki işletim sistemi ne olursa olsun veya hangi programlama dilini kullanırsa kullansın, erişim sağlayabilir. Bu sayade, Web servislerinin kullanımı birlikte çalışabilirlik kavramına yeni bir yaklaşım getirmiştir [17].

3.2.3 Web Servis Fonksiyonları

3.2.3.1 Harita Servisleri

Harita servisleri, kullanıcıların belli bir coğrafi alan için harita isteklerini karşılarlar. Bu istek karşısında harita servisi kullanıcıya, JPEG, PNG ya da GIF görüntü formatlarında harita sunar. Bu Web servisi aynı zamanda öznitelik sorgulamaya ve tanımlamaya olanak sağlar. Konumsal Web servislerinin en çok kullanılan türlerinden biridir.

3.2.3.2 Veri Servisleri

Veri servisleri kullanıcıların sorgulama, düzenleme ve veri senkronizasyonu yapabilmemesine olanak sağlayan servistir. Harita görüntülemeye de imkan veren bazı veri servisleri aynı zamanda bir harita servisidir.

Vektör Veri Düzenleme Servisleri: Düzenleme servisleri kullanarak, son kullanıcı Web üzerinden bir veri tabanına vektör veri ekleyebilir, veri tabanındaki veriyi düzenleyebilir veya silebilir.

Arama Servisleri: CBS veri kaynaklarını dizinlemek ve arama yapabilmek için kullanılır. Kullanıcı anahtar kelimeler kullanarak arama servisinde CBS veri kaynakları arayabilir.

Görüntü Servisleri: Uzaktan algılama görüntüleri ve sayısal yüksek modelleri gibi raster verileri erişim sağlar.

Veri Senkronizasyon Servisleri: İki farklı coğrafi veri tabanı arasında internet üzerinden periyodik olarak veri güncellemelerini otomatik olarak gerçekleştiren servis türüdür.

3.2.3.3 Analitik (Çözüm) Servisler

Analitik servisler, çeşitli CBS analizleri gerçekleştirirler. Bu servisler aynı zamanda kullanıcının tasarladığı iş planını barındırabilir.

Coğrafi Kodlama Servisleri: Coğrafi kodlama, sokak adreslerini coğrafi koordinatlara dönüştürme işlemidir. Coğrafi kodlama servisi, bu fonksiyonu Web'e taşır.

Ağ Analizi Servisleri: En kısa yol analizi, servis alanı hesaplama ve en yakın tesis bulma gibi analizler yapma olanağı sağlayan servis türüdür.

Geoprocessing Servisleri: Yerel bilgisayarda oluşturulmuş olan fonksiyonları ve modelleri Web üzerinden paylaşma esnekliğini sağlar.

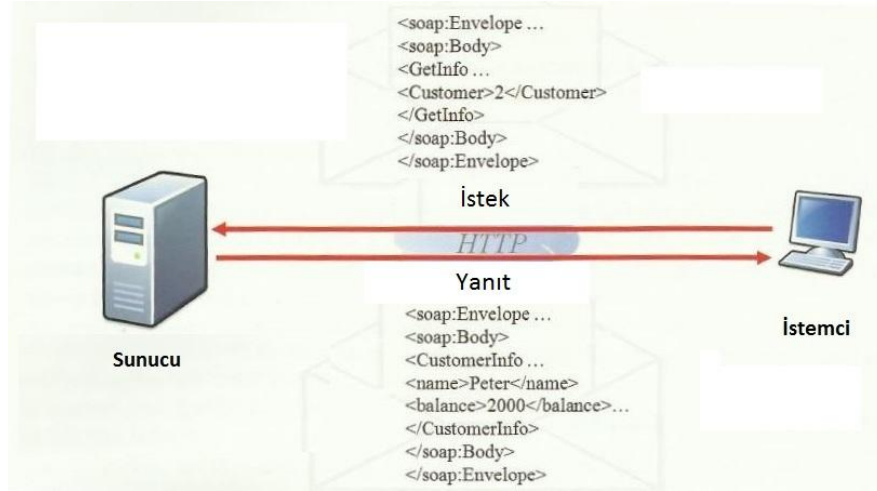
Metadata Katalog Servisleri: Metadata, CBS veri ve servislerini açıklamak için kullanılır. Metadata katalog servisi metadata yayımlamaya ve aramaya olanak sağlar.

3.2.4 Web Servisi Türleri

Web servisi ve istemci arasındaki bağlantı, istemcinin istek göndermesi ve Web servisin cevabından oluşur. Kullanılan bağlantının türüne bağlı olarak, Web servisleri iki ana başlık altında toplanabilir.

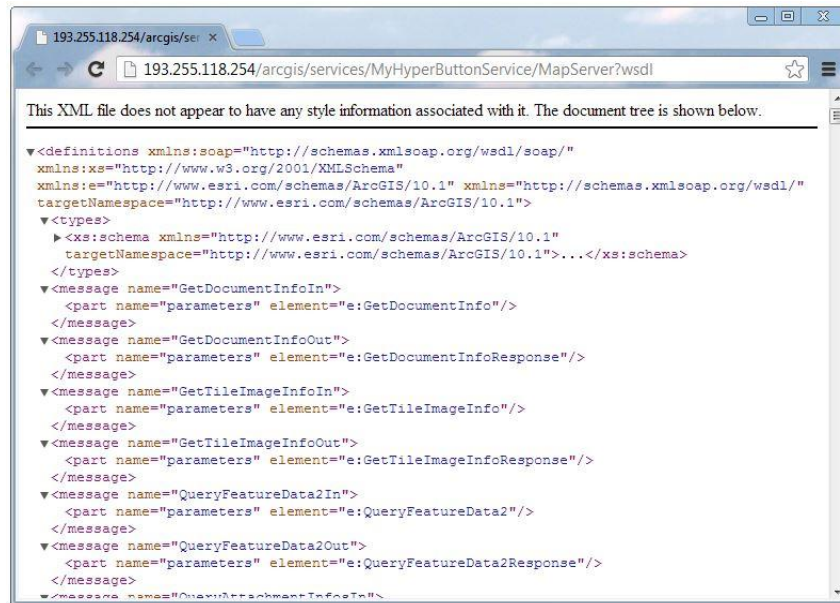
3.2.4.1 SOAP Tabanlı Web Servisleri

SOAP (Simple Object Access Protocol), XML (Extensible Markup Language) formatında veri alışverişi için bir protokol tanıımıdır. SOAP tabanlı Web servisleri, istek göndermek için HTTP (Hypertext Transfer Protocol) gönderileri kullanır. İstekler ve cevaplar SOAP içeren XML formatındadır. (Şekil 2.5)



Şekil 2. 5 SOAP tabanlı web servislerinin mimarisi [17]

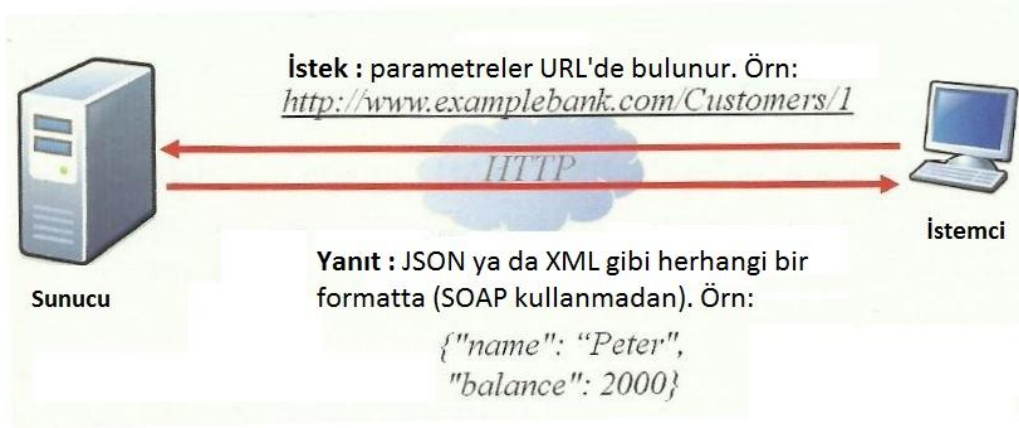
SOAP genellikle WSDL (Web Service Description Language) ile birlikte kullanılır. WSDL, Web servislerini ve onlara nasıl erişileceğini açıklayan XML tabanlı bir dildir. Örneğin; ArcGIS Server üzerindeki bir Web servisinden gelen SOAP yanıtı Şekil 2.6'da gösterilmiştir.



Şekil 2. 6 SOAP tabanlı bir web servisinin yanıtı

3.2.4.2 REST Tabanlı Web Servisleri

Representational State Transfer (REST) mimarisi, Hyper Text Transfer Protocol (HTTP)'nin tüm avantajlarından yararlanmak için tasarlanmıştır [5]. REST tabanlı Web servisleri, SOAP katmanı kullanmadan HTTP üzerinden veri aktaran Web servisleridir. Bu Web servisinde kullanıcı tüm parametreleri istek Uniform Resource Locator (URL)'si ile gönderir (Şekil 2.7).



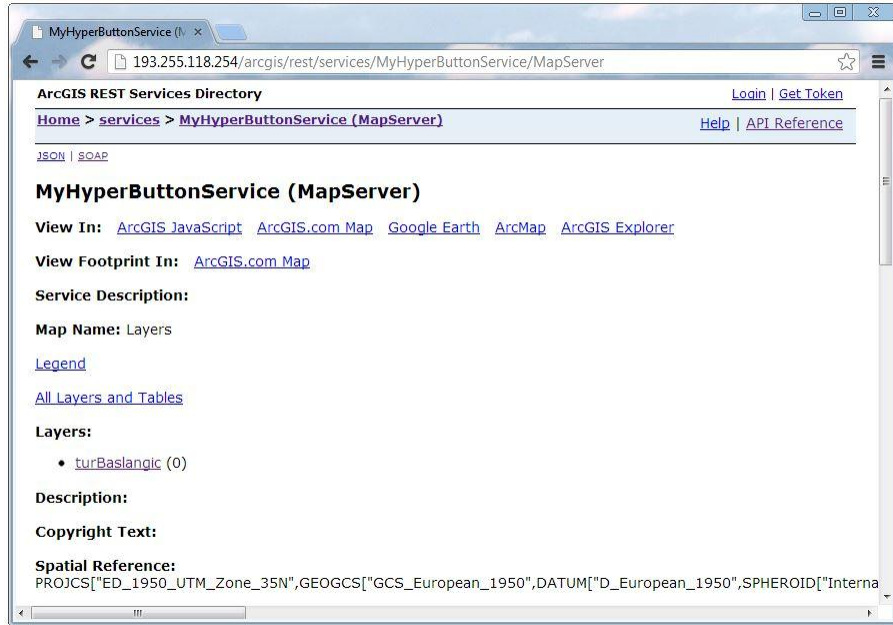
Şekil 2. 7 REST tabanlı web servislerinin mimarisi [17]

ArcGIS Server üzerindeki Web servislerinin listesine erişmek için, sunucuya "http://Server Adresi/arcgis/rest/services" isteği gönderilebilir. Sonuç ekranı Şekil 2.8'de gösterilmiştir.



Şekil 2. 8 ArcGIS Server yazılımının REST arayüzü

ArcGIS Server üzerindeki bir Web servisi üzerindeki bir Web servisi için istek URL'si "http://Server Adresi/arcgis/rest/services/Web Servis Adı/Servis Türü" şeklinde olmalıdır. Web yanıtı ise Şekil 2.9'da görüldüğü gibi olacaktır.



Şekil 2. 9 REST tabanlı web servisinin yanıtı

Bir Web servisinde bulunan katmanlara, 0'dan başlayarak kendilerine özgü kimlik numaraları atanır. Katmanlara ulaşmak için yapılması gereken istek "http://Server Adresi/arcgis/rest/services/Web Servisi Adı/Servis Türü/Katman No" şeklinde olmalıdır. Bu isteğe karşılık gelen yanıt Şekil 2.10'da gösterildiği gibidir.



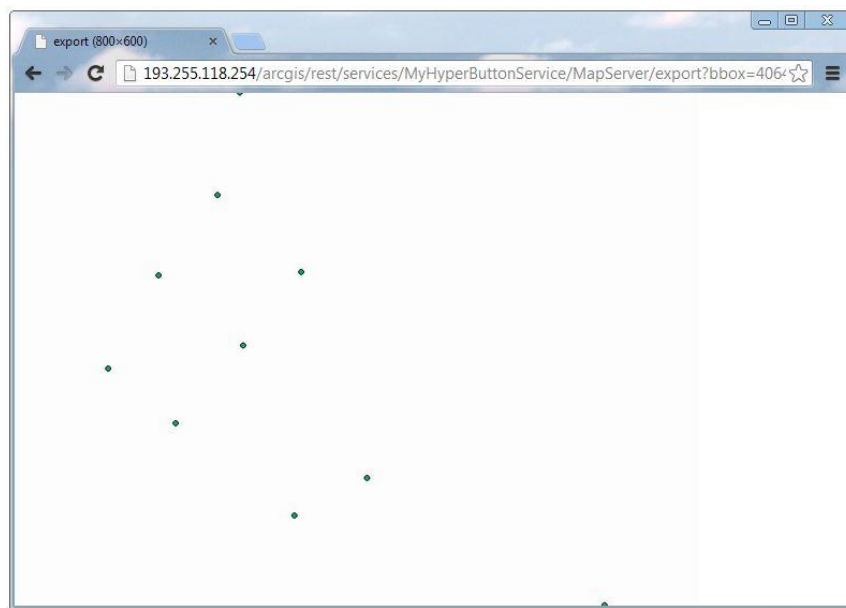
Şekil 2. 10 REST tabanlı web servisinin katman isteği yanıtı

ArcGIS Server REST API kullanılarak, Web servisi üzerinde sorgu veya dışa aktarma (export) gibi işlemler yapılabilir. Örneğin; bir katmanı PNG formatında görüntülemek için Çizelge 2.1'deki istek gerçekleştirilebilir. Bu görüntüleme işlemi içerisinde bir nokta katmanı bulunan bir Web servisi üzerinde yapılmıştır.

Çizelge 2. 1 ArcGIS Server REST API ile dışa aktarma sorgusu

Adres	Açıklama
http://193.255.118.254/arcgis/rest/services/MyHyperButtonService/MapServer/export?	Web Servisi Adresi
bbox=406460.2328, 4543646.0029, 407440.9117, 4544658.7823	Görüntülenmek İstenen Alanın Boyutları
&size=800, 600	Sonuç Görüntünün Boyutları
&format=png	Görüntü Formatı
&dpi=96	DPI Değeri
&f=image	Yanıt Türü

Bu işlem sonucunda elde edilen görüntü Şekil 2.11'de gösterilmiştir.



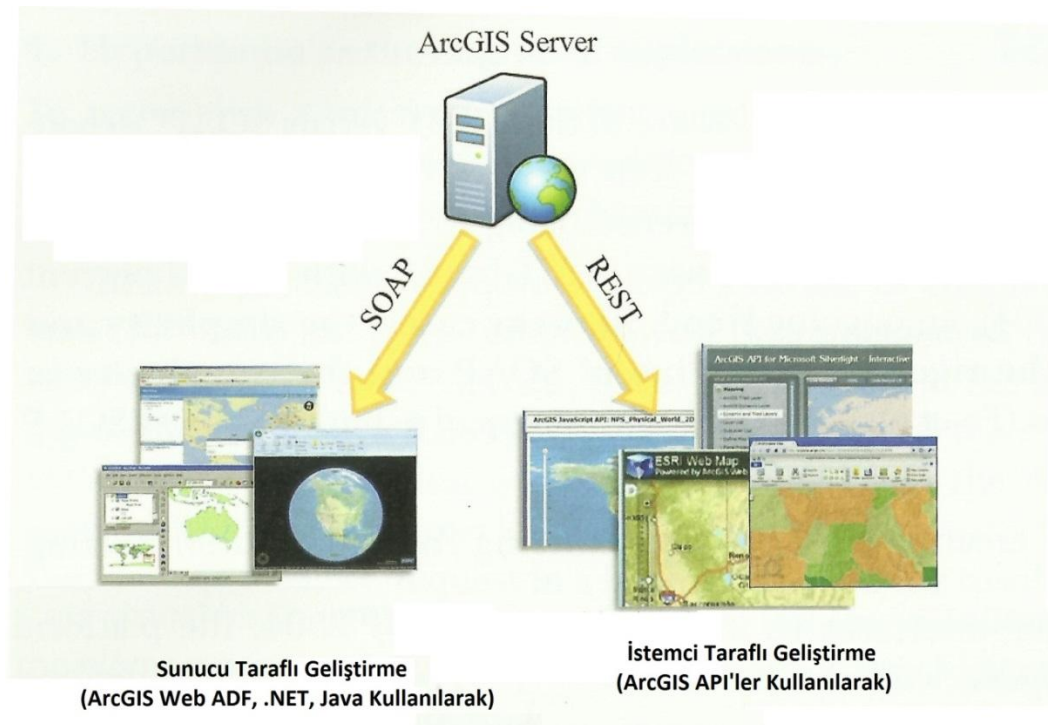
Şekil 2. 11 ArcGIS Server REST API ile gerçekleştirilmiş sorgu yanıtı

3.2.4.3 SOAP ve REST Tabanlı Servislerin Karşılaştırılması

SOAP protokolün sıkı disiplini ve SOAP tabanlı Web servislerinin karmaşık olması sebebiyle REST servislerinin kullanımı daha yoğundur [19]. REST servislerinin kullanılması, üreticiler için Web servisinin üretilmesi ve yayınlamasının maliyetinin düşürmesi, kullanıcıların daha çabuk öğrenmesi ve CBS sistemleri üretme hızını arttırması, yöneticilere ise esnek, güvenilir ve genişletilebilir bir mimari sunması gibi avantajlar sağlar [17].

SOAP Web servisleri kapsamlı ve güçlüdür bunun yanında karmaşıktır. REST Web Servisleri ise, basit ve verimlidir fakat SOAP protokolünün tüm yeteneklerine sahip değildir [20].

ArcGIS Server yazılımın hem SOAP tabanlı hem de REST tabanlı Web servisleri üretir (Şekil 2.12). Şekil 2.12’de görüldüğü gibi, SOAP tabanlı Web servisleri .NET, Java veya ArcGIS Server Web ADF (Application Development Framework) kullanılarak, server tarafından geliştirilen web uygulamalarında kullanılmaktadır. REST tabanlı Web Servislerinden, ArcGIS APIs (Application Programming Interface) için Javascript, Flex ve Silverlight kullanılan, istemci taraflı geliştirilen uygulamalarda yararlanır.



Şekil 2. 12 ArcGIS Server yazılımında SOAP ve REST tabanlı web servisleri [17]

Bu çalışmada, REST tabanlı Web Servisleriyle birlikte ArcGIS API for Silverlight kullanılmıştır. Bu nedenle, bir sonraki bölümde daha çok ArcGIS API for Silverlight üzerinde durulacaktır.

3.3 Web Servislerini Kullanma (API)

İstemci taraflı API'ler, kullanıcıların Web servislerini arka planda neler olduğunu bilmeden kullanmalarına imkan verdikleri için diğer yöntemlere göre daha kolay kullanılır. API'ler Web servislerinin görünümlerini ve kullanıcı ilişkilerini gibi bileşenlerin kurulumunu rahatlıkla sağlar. [17]

3.3.1 ArcGIS API for JavaScript

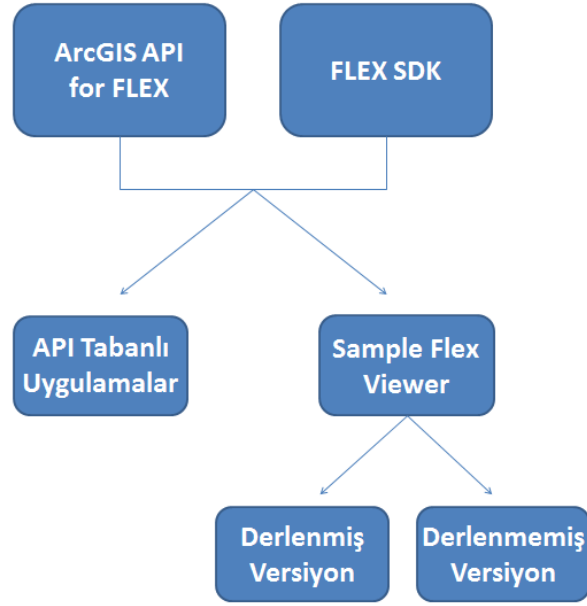
ArcGIS API for JavaScript, web uygulamalarında Web servislerini ve araçlarını kullanmak için basit bir yoldur. Geliştirilen bir uygulama çalıştığı zaman, kod sunucu üzerinde değil, tarayıcı üzerinde çalışır [21], [22].

ArcGIS API for JavaScript, Dojo Javascript aracının üzerine yazılmıştır [23]. Böylece birçok hazır kütüphaneler sunmaktadır.

3.3.2 ArcGIS API for Flex

Flex, Adobe tarafından sunulan açık kaynaklı bir frameworktür. Flex uygulamaları HTML sayfaları içerisinde çalışan Flash (SWF) dosyalarıdır.

ArcGIS API for Flex, kullanıcıların ArcGIS Server üzerinde zengin internet uygulamaları (Rich Internet Applications: RIA) geliştirmelerine imkan sağlar. Bu API sayesinde, ArcGIS Server veri kaynakları ve Flex bileşenleri birarada kullanılarak, interaktif ve canlı web uygulamaları geliştirilebilir [24],[25] (Şekil 2.13).



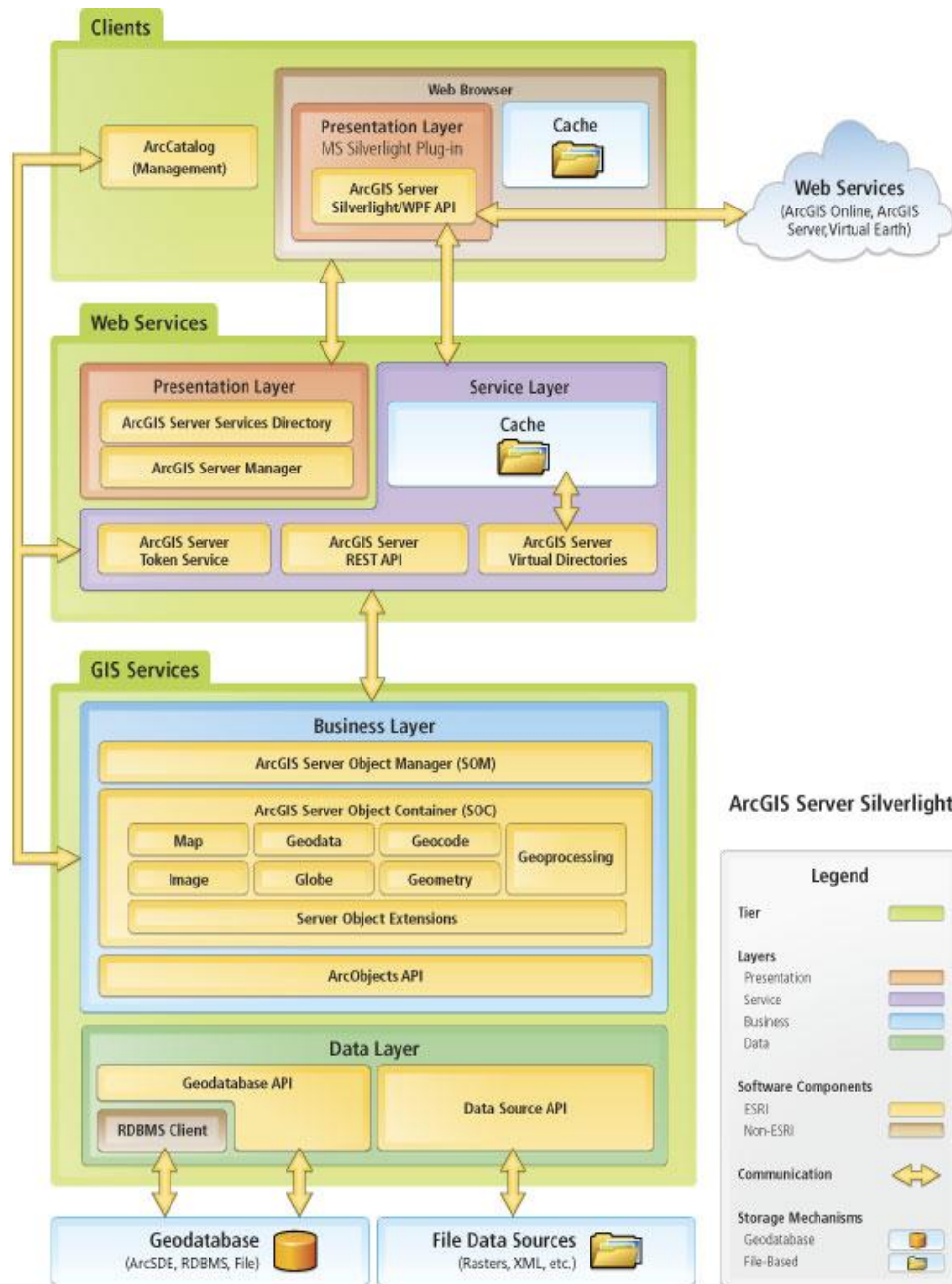
Şekil 2. 13 ArcGIS API for Flex mimarisi [25]

3.3.3 ArcGIS API for Silverlight

Microsoft Silverlight, grafikler, animasyonlar ve vidyolar barındıran zengin internet uygulamaları yaratmak ve görüntülemek için .NET Framework içerisinde çalışan güçlü bir araçtır. Bu araçla birlikte programlama için .NET dilleri, kullanıcı arayüzü için Extensible Application Markup Language (XAML)kullanılır. [26],[27]

ArcGIS API for Silverlight, kullanıcıların ArcGIS Server ve Bing Haritaları servislerini ve araçlarını bir Silverlight uygulamasında entegre edebilmesine imkan sağlar (Şekil 2.14).

Bu API'ın kullanıcılara sağladığı en büyük avantaj, Visual Studio ortamında uygulama geliştirilebilmesidir. Aynı zamanda API ile gelen sayısal kütüphane ile birlikte, kullanıcılara web harita uygulamalarında kullanılabilecekleri birçok hazır araç ve bileşenler sunmaktadır (Çizelge 2.2)



Şekil 2. 14 ArcGIS API for Silverlight mimarisi [28]

Çizelge 2. 2 ArcGIS API for Silverlight Kütüphanesi [29]

Kütüphane	Açıklama
ESRI.ArcGIS.Client.dll	Haritalar, ArcGIS Katmanları, Grafikler, Semboller, Geometri
ESRI.ArcGIS.Client.Bing.dll	Bing katmanları ve servislerini kullanma desteği
ESRI.ArcGIS.Client.Webmap.dll	Web Haritaları içeriğini ve metadatasının okuyan ve işleyen bileşenler
ESRI.ArcGIS.Client.Behaviors.dll	Haritanın interaktif olmasını sağlar
ESRI.ArcGIS.Client.Toolkit.dll	Harita kullanımı için araç çubuğu, kaydırma tuşları gibi kontroller
ESRI.ArcGIS.Client.Toolkit.DataSources.dll	WMS, OpenStreetMap gibi veri kaynakları desteği

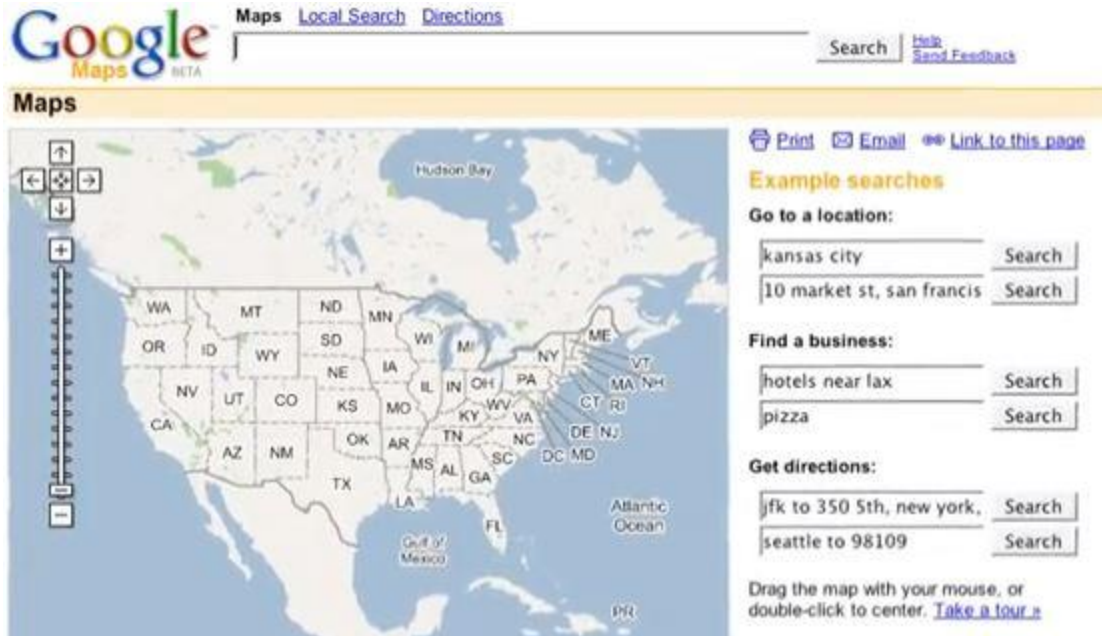
BÖLÜM 3

SOKAK GÖRÜNÜMÜ

Sokak görünümü, dünyada bir çok mekanda, sokak seviyesinde Google Earth yazılımı ve Google Maps aracılığıyla panoramik görüntüler sunan bir teknolojidir.

3.1 Sokak Görünümünün Oluşumu

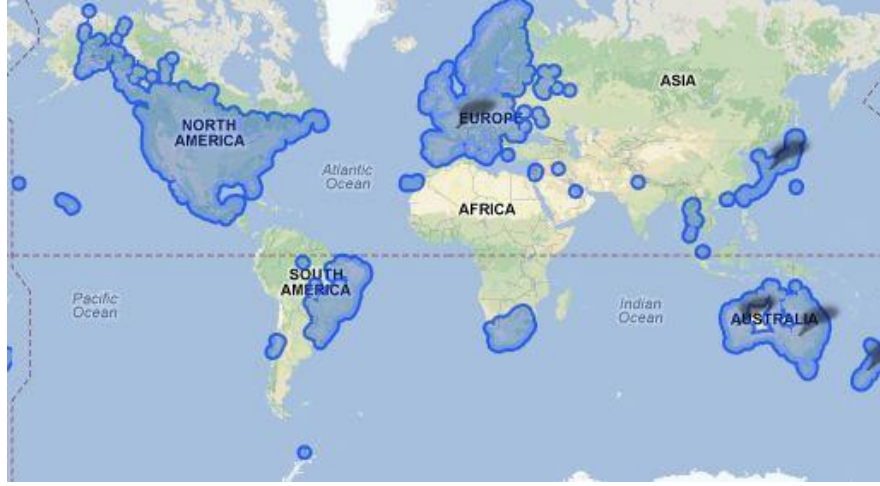
Google, 1998 yılında özel bir şirket olarak kurulduğunda sadece bir arama motorundan ibaretti. Şirket, bilgiyi artık coğrafi boyutta da sunmak amacıyla 2005 yılında Google Haritalar'ı kullanıma sunmuştur (Şekil 3.1).



Şekil 3. 1 Google haritaların ilk kullanım arayüzü [30]

Aynı yıl içerisinde Google Earth yazılımı ücretsiz olarak kullanıcılara sunulmuştur. Kullanıcılarla iletişimi arttırmak isteyen şirket, Google Earth ile birlikte, üçüncü parti kişilerin Google haritalarını kullanabilmelerini sağlayan ilk Google Maps API servisini hayata geçirmiştir.

2007 yılında Google, 5 Amerikan kenti için Sokak Görümü'nü yayınlamıştır. İlerleyen yıllarda Sokak Görünümü'nün kapsama alanı gittikçe artmıştır (Şekil 3.2).



Şekil 3. 2 Sokak görünümü kapsama alanı [31]

Google Maps Javascript API v3 ile Google kullanıcılara Sokak Görünümü servisi sağlamaya başlamıştır. Böylece, üçüncü parti kullanıcılar kendi oluşturdukları sokak görüntüleri Google arayüzü ile web sitelerinde sunabilme imkanı da bulmuşlardır (Şekil 3.3).



Şekil 3. 3 Kişisel panorama ile oluşturulmuş sokak görünümü

3.2 Sokak Görünümü Nedir? Nasıl Çalışır?

Sokak Görünümü, kullanıcılara belirli yollar üzerinde 360 derece panoramik görüş sağlar. Sokak görünümündeki görüş, tek bir noktadan 360 derece çekilen fotoğraflardan oluşur (Şekil 3.4).



Şekil 3. 4 Küresel projeksiyona izdüşürülmüş 360 derece panoramik görüntü

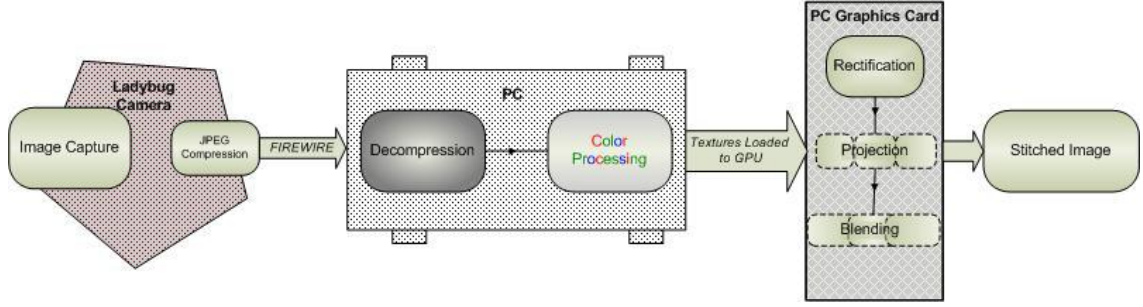
3.2.1 Panoramaların Oluşumu

Google, Sokak Görünümü projesinde Point Grey Research tarafından geliştirilen Ladybug kameralarını kullanmaktadır. Bu donanımda 6 adet kamera bulunmaktadır (Şekil 3.5).



Şekil 3. 5 Ladybug 3 ve Ladybug 2 kameraları [32]

Panoramik görüntü bu 6 kameradan elde edilen ham görüntülerin birleştirilmesiyle oluşturulmaktadır. Görüntü birleştirme işlemine genel bakış Şekil 3.6'da görülebilir.



Şekil 3. 6 Görüntü birleştirme işlemi şeması [32]

6 adet fotoğraf senkronize olarak çekilir. Bilgisayara aktarma işlemini kısaltmak amacıyla görüntüler JPEG olarak sıkıştırılabilir. Bu durumda gerekirse görüntüler bilgisayarda tekrar ham(raw) hale dönüştürülmektedir. Belirlenen bir enterpolasyon yöntemi ile ham(raw) görüntüler RGB'ye dönüştürülür (Şekil 3.7).



Şekil 3. 7 RGB görüntüler (sırasıyla : kamera 0, kamera 1, kamera 2, kamera 3, kamera 4, kamera 5)



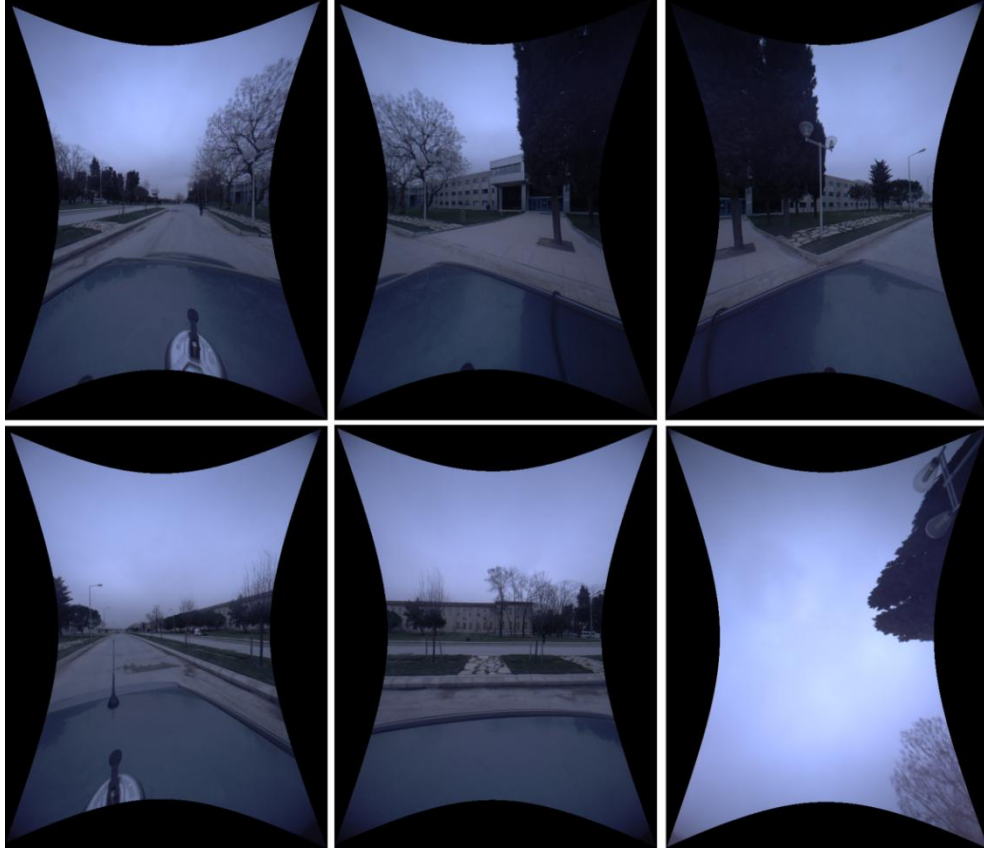
Şekil 3. 8 Renk işleme yöntemine göre sonuçlar, Nearest Neighbor(solda), Edge Sensing(ortada), High Quality Linear(sağda)

Belirlenen interpolasyon yöntemine göre elde edilen sonuçların renk kaliteleri değişmektedir (Şekil 3.8). Buna paralel olarak renk kalitesi arttıkça, renk işleme aşamasının süresi artmaktadır. Ladybug kameralarında rigorous interpolasyon yöntemi en iyi renk kalitesini verir fakat işleme süresi çok uzundur (Şekil 3.9).

Processor Type	Processor Speed	Operating System	Architecture	Nearest Neighbor Fast (ms)	Nearest Neighbor (ms)	Edge-Sensing (ms)	High-Quality Linear (ms)	Rigorous (ms)
Intel Xeon	3 GHz	Windows XP Prof. SP2	64-bit	3.4	4.2	10.0	22.4	863
Intel Pentium 4	2.93 GHz	Windows XP Prof. SP3	32-bit	4.3	7.8	16.8	29.1	1036
Intel Core2 Duo E6750	2.66 GHz	Windows XP Prof. SP2	32-bit	3.2	4.6	11.4	22.9	544
Intel Core2 Duo T9300	2.66 GHz	Windows Vista SP1	32-bit	3.2	4.7	12.1	24.3	537
Intel Core2 6600	2.4 GHz	Windows Vista SP1	64-bit	2.7	4.2	11.5	23.7	611
Intel Core2 Quad Q6600	2.4 GHz	Windows XP Prof. SP2	32-bit	3.6	5.3	12.8	25.5	603
Intel Core2 Quad	2.4 GHz	Windows XP Prof. SP2	32-bit	3.7	5.2	15.2	25.3	601
Intel Xeon	2.4 GHz	Windows XP Prof. SP2	32-bit	7.9	11.0	20.1	48.9	1254
AMD Athlon 64 3800+	2.4 GHz	Windows XP Prof. SP3	32-bit	6.4	9.4	24.9	36.3	853
Intel Core2 6300	1.86 GHz	Windows XP Prof. SP2	32-bit	5.4	7.5	17.2	34.0	796
Intel Core2 4300	1.8 GHz	Windows XP Prof. SP2	32-bit	4.8	7.2	17.1	34.1	806
AMD Athlon 64 3000+	1.8 GHz	Windows XP Prof. SP3	32-bit	5.9	9.4	19.5	38.8	907
Intel Atom N270	1.6 GHz	Windows XP Home SP3	32-bit	9.7	14.7	31.8	79.3	1789

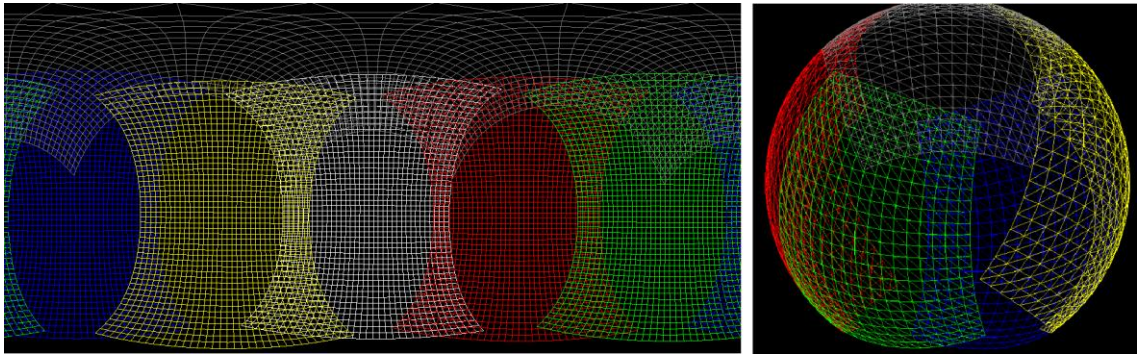
Şekil 3. 9 Renk işleme yöntemlerinin süreye ve donanıma göre karşılaştırılması (görüntü çözünürlüğü: 1280x960, bit: 8 bits/px, işlemden sonraki bit: 24 bits/px) [32]

Elde edilen 6 adet görüntü rektifiye edilerek, düzleme ya da küresel yüzeye giydirilir (Şekil 3.10) ve meshler oluşturularak, görüntü birleştirme işlemi yapılır. Böylece, poligonların köşe noktalarının koordinatları 2 boyutlu ya da 3 boyutlu düzlemde elde edilebilir.



Şekil 3. 10 Rektifiye edilmiş görüntüler

Eğer görüntü, bilgisayarda düzlem projeksiyonda gösterilmesi isteniyorsa, poligonlar 2 boyutlu mesh'ler olarak dizilir. Küre (Spherical) projeksiyonda gösterilmesi isteniyorsa, poligonlar 3 boyutlu mesh'ler olarak dizilir (Şekil 3.11).



Şekil 3. 11 2 boyutlu mesh (solda), 3 boyutlu mesh (Sağda) [32]

3.2.1.1 Hata Kaynakları

Birleştirme işlemi her zaman uygun sonuç vermeyebilir. Bazı hatalara sebep olan etkenler vardır. Öncelikle, 6 kameradan oluşan bu sistemde, kameralar arasında fiziksel olarak mesafe olduğu için paralaks meydana gelir. İkinci olarak, görüntüleri mükemmel bir şekilde birleştirmek için kameradan objeye olan uzaklık bilinmelidir. Fakat, sistemde uzaklık ölçülmemektedir. Bu bilinmeyi gidermek için, her objenin kameraya olan uzaklığı eşit olduğu kabul edilerek, 20 metre çapında bir küre fiziksel düzlem olarak kullanılır. 10 metre ve daha uzak objelerde bu varsayım, en fazla 1 piksel paralaks hatasına sebep olur. Fakat, 10 metreden daha yakın objelerde, çift görüntü hatası meydana gelir (Şekil 3.12).



Şekil 3. 12 Çift görüntü hatası

Bu hatadan dolayı kaynaklanan birleştirme hatası miktarını hesaplamak için şu formül kullanılır:

$$\text{Birleştirme Hatası (piksel)} = \left| \frac{1}{Z} - \frac{1}{Z_c} \right| K \quad (3.1)$$

Formülde Z kameradan objeye olan mesafe, Z_c varsayılan mesafe, K ise kamera sabitidir.

Birleştirme hatası meydana geldiğinde, kaynaşma genişliği değeri önem kazanır. Hata varsa bu değerin yüksek olması çift görüntü etkisine sebep olur (Şekil 3.14). Fakat bu değerin düşük olması da görüntü ofseti oluşmasına neden olur (Şekil 3.13). Bu nedenle, yakın objelerin görüntülenmesinde birleşme hatası ve kaynaşma genişliği arasında ters orantı vardır.



Şekil 3. 13 Kaynaşma genişliği = 0 durumu (görüntü ofseti oluşmuştur)



Şekil 3. 14 Kaynaşma genişliği = 100 durumu (çift görüntü hatası oluşmuştur)

Son olarak, her görüntünün sınırları, orta kısmına göre daha karanlık gözüktür. Bu durum birleştirilmiş görüntüde izlerin ortaya çıkmasına sebep olur (Şekil 3.15). Bu izler, parlaklık ayarlarıyla giderilebilir, fakat bu da birleştirilmiş görüntünün işleme süresinin uzamasına neden olur (3.16).



Şekil 3. 15 Parlaklık düzeltmesi kapalı



Şekil 3. 16 Parlaklık düzeltmesi açık

3.2.2 Projeksiyonlar

Birleştirilen görüntünün yansıtılması için farklı projeksiyonlar vardır. Her projeksiyon sisteminin avantajları ve dezavantajları vardır.

3.2.2.1 Küresel (Equirectangular – Plate Carrée) Projeksiyon

Google, sokak görünümü sistemindeki görüntüler küresel (equirectangular) projeksiyon sistemine yansıtılmıştır. Bu projeksiyonda görüntüler 2:1 en-boy oranını korurlar.

Bu projeksiyonda görüntü yatay ve düşey olmak üzere iki koordinata sahiptir. Görüntüyü küresel koordinat sisteminden, 2 boyutlu düzleme indirgemek için eşitlikler şu şekildedir:

$$R = \sqrt{X^2 + Y^2 + Z^2} \quad (3.2)$$

$$\theta = \arctan^2(Y, X) \quad (3.3)$$

$$\varphi = \arccos\left(\frac{Z}{R}\right) \quad (3.4)$$

X, Y, Z : Küresel Koordinat Sistemindeki Koordinatlar

θ, φ : Düzleme İndirgenmiş Koordinatlar



Şekil 3. 17 Küresel projeksiyon geometrisi

Bu projeksiyon sisteminin avantajı, küresel sistemdeki bütün noktalar tek bir görüntüye dönüştürülebilir. Dezavantajı ise, kürenin kutuplarına yaklaşıldıkça, pikseller genişlemeye başlamaktadır (Şekil 3.18).



Şekil 3. 18 Küresel projeksiyona indirgenmiş panoramik görüntü

3.2.2.2 Silindirik Projeksiyon

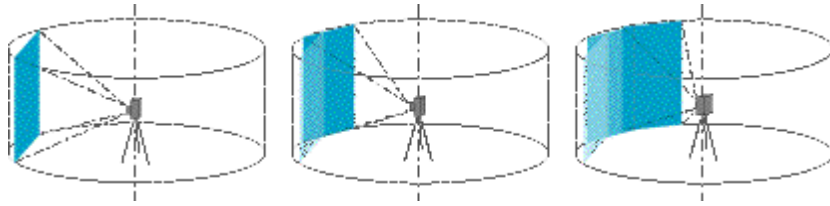
Silindirik projeksiyona indirgenmiş bir görüntü, kürenin kutupları ve etrafı hakkında bilgi içermez (Şekil 3.19, Şekil 3.20).

Projeksiyon eşitlikleri şu şekildedir:

$$R = \sqrt{X^2 + Y^2 + Z^2} \quad (3.5)$$

$$\theta = \arctan^2(Y, X) \quad (3.6)$$

$$\varphi = -\frac{ZR}{R^2 - Z^2} \quad (3.7)$$



Şekil 3.19 Silindirik projeksiyon geometrisi



Şekil 3. 20 Silindirik projeksiyona indirgenmiş panoramik görüntü

3.2.2.3 Kubbe Projeksiyon

Bu projeksiyonda küre, kubbe benzeri şekile yansıtılır. Bu projeksiyonda kürenin kuzey kutbu kubbenin merkezinde, güney kutbu ise kubbenin dış kısmına çizilir (Şekil 3.21, Şekil 3.22).

Projeksiyonun eşitlikleri şu şekildedir:

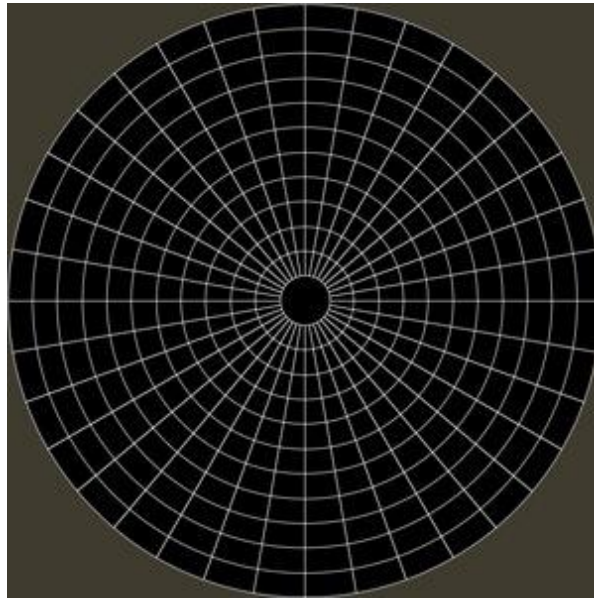
$$R = \sqrt{X^2 + Y^2 + Z^2} \quad (3.8)$$

$$\varphi = \arccos\left(\frac{Z}{R}\right) \quad (3.9)$$

$$U = \varphi \frac{X}{R} \quad (3.10)$$

$$V = \varphi \frac{Y}{R} \quad (3.11)$$

U, V : Kubbe Projeksiyonunda Görüntü Koordinatları



Şekil 3. 21 Kubbe projeksiyon geometrisi



Şekil 3. 22 Kubbe projeksiyona indirgenmiş panoramik görüntü

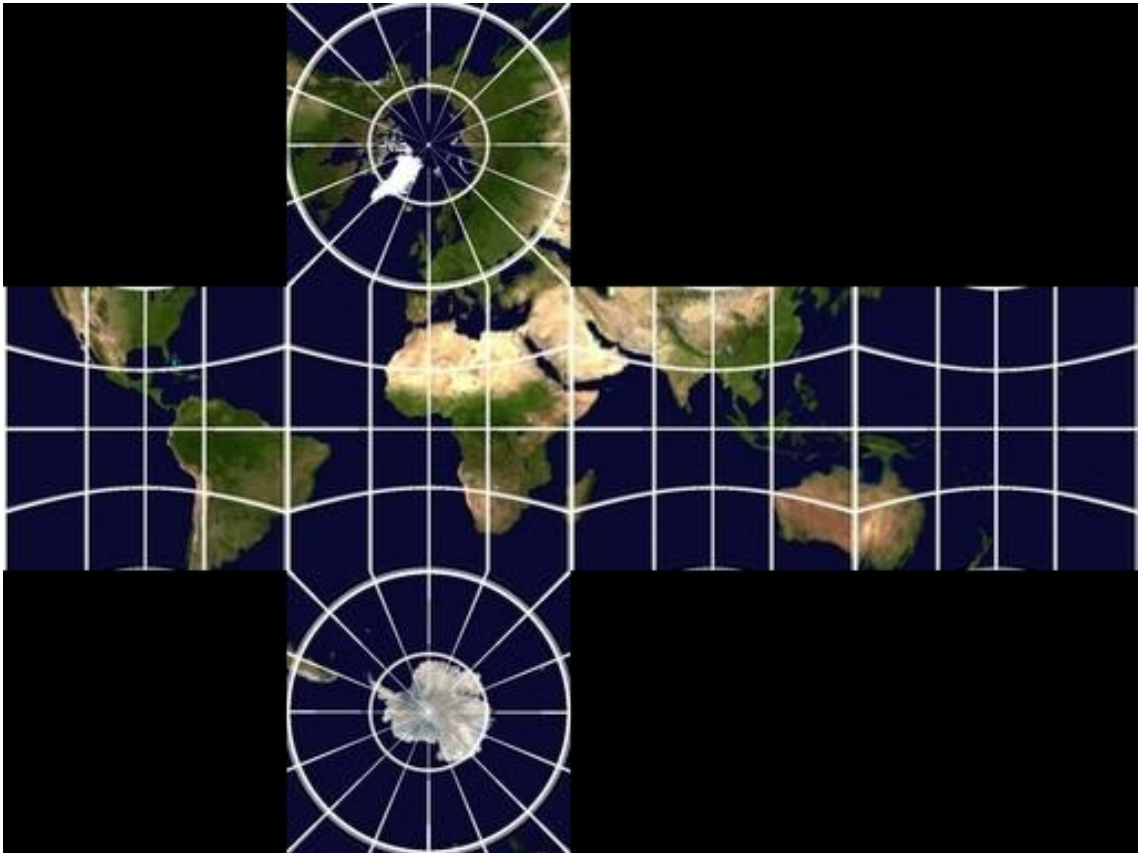
3.2.2.4 Kübik Projeksiyon

Bu projeksiyonda her görüntünün kübün bir yüzü olduğu 6 görüntü gerekir. Bütün küreyi 6 görüntüye bölerek, her görüntüde distorsiyon miktarı azaltılır (Şekil 3.23, Şekil 3.24). Bu projeksiyon tipi daha çok video oyunları uygulamalarında kullanılmaktadır.

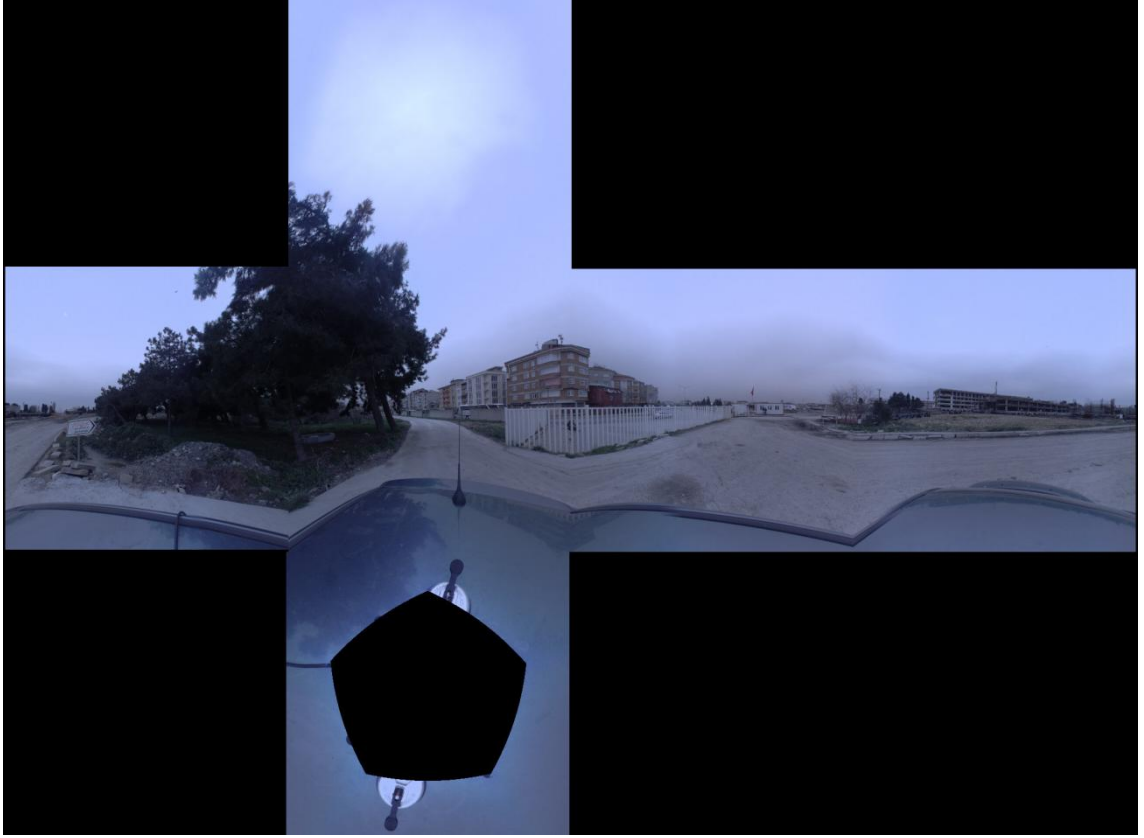
Projeksiyon eşitlikleri şu şekildedir:

$$U = \frac{x}{z} \quad (3.12)$$

$$V = \frac{y}{z} \quad (3.13)$$



Şekil 3. 23 Kübik projeksiyon geometrisi



Şekil 3. 24 Kübik projeksiyona indirgenmiş panoramik görüntü

3.2.3 Panoramik Döşemeler (Tile)

Google Sokak Görünümü, her yakınlaştırma seviyesi için farklı görüntü detayı sunar. Sokak görünümünde genelde 5 yakınlaştırma seviyesi vardır. Her yakınlaştırma seviyesi farklı görüş alanına sahiptir.

Görüntü, yakınlaştırma seviyesine göre parçalara(tile'lara) ayrılır (Şekil 3.25). Bu parçalar $4^{\text{YakınlıkSeviyesi}}$ formülüne göre belirlenir. Yani, 0 yakınlık seviyesinde görüntünün tümü ($4^0 = 1$), 2 yakınlık seviyesinde ise görüntü $4^2 = 16$ parçaya ayrılır. Sokak görünümü uygulamasında sadece o yakınlık seviyesine ve o görüş alanına uygun görüntü parçaları çağırılır.

Çizelge 3. 1 Yakınlık seviyesine göre görüş alanları [33]

Yakınlaştırma Seviyesi	Görüş Alanı (Derece)
0	180
1 (Varsayılan)	90
2	45
3	22.5
4	11.25



Şekil 3. 25 Görüntü parçaları

3.2.4 Yasal Boyut

Google Sokak görünümü, sadece halka açık yollar ve alanlarda mevcuttur. Özel parklar, eğlence alanları, üniversite kampüsleri Sokak Görünümü'ne dahil değildir. Bu tip kurum ve işletmeler Google Participate programına dahil olarak, kapsama alanına girebilirler.

Çekilen fotoğrafların birçoğunda insanlar, arabalar, evler, özel mülkler bulunmaktadır. Google kişilerin mahremiyet haklarına saygı duymak amacıyla, bulanıklaştırma teknolojisini geliştirmiştir. Bu teknolojiyle, insanların yüzleri, araba plakaları gibi kişiye

özel bilgiler bulanıklaştırılmaktadır. Kullanıcılar, yaşadıkları bölge için daha fazla bulanıklaştırma talebinde bulunabilir (Şekil 3.26).



Şekil 3. 26 Bulanıklaştırma işleminin uygulandığı bölgeler

3.3 Google Maps Javascript API v3 ve Street View Servisi

Google Maps Javascript API kullanıcıların kendi internet sitelerinde Google haritalarını kullanmalarını sağlar. API'nın bu sürümü daha hızlı ve masaüstü bilgisayar internet tarayıcıları kadar mobil uygulamalarda da uygulanabilir olması için tasarlanmıştır. Bu API'nın en önemli özelliklerinden birisi de Sokak Görünümü servisi sunmasıdır. Bu servis sayesinde artık kullanıcılar Google haritalarını kullanabildiği gibi, Sokak Görünümünü kullanabilir, kendileri fotoğraflar çekerek Sokak Görünümleri oluşturabilir ve oluşturdukları turları Google'ın mevcut Sokak Görünümlerine bağlayabilirler.

Sokak Görünümü servisi şu sınıflar(class) ve nesneleri (object) içerir:

- StreetViewPanorama
- StreetViewPanoramaOptions
- StreetViewAddressControlOptions
- StreetViewLink

- StreetViewPov
- StreetViewPanoramaData
- StreetViewLocation
- StreetViewTileData
- StreetViewService
- StreetViewStatus

HAREKET SENSÖRÜ VE DERİNLİK HARİTALARINDAN HAREKETİN ALGILANMASI

İnsan hareketi tanımlama, video ve resim dizilerinden insan hareketlerinin şablonunu anlamayı hedefler [34]. Birden fazla kameralar kullanılsa bile, insan hareketinin kompleks olmasından ve görüntülerdeki insanların dış görünüşlerinin değişken olmasından dolayı tanımlama işlemi zorlayıcı bir hal alır. Yeni teknolojik gelişmelerle birlikte, anlık olarak ortamı 3 boyutlu tarayabilen kameralar ortaya çıkmıştır. Derinlik haritaları ortam ışığından ve dış görünüşten neredeyse bağımsızdır [35].

Microsoft Kinect, 4 Kasım 2010 tarihinde piyasaya sürülmüştür. Kinect, anlık olarak derinlik haritası üretebilmektedir. Bu donanımın en büyük avantajı ise 150\$ gibi ucuz bir market fiyatına sahip olmasıdır.

4.1 Microsoft Kinect

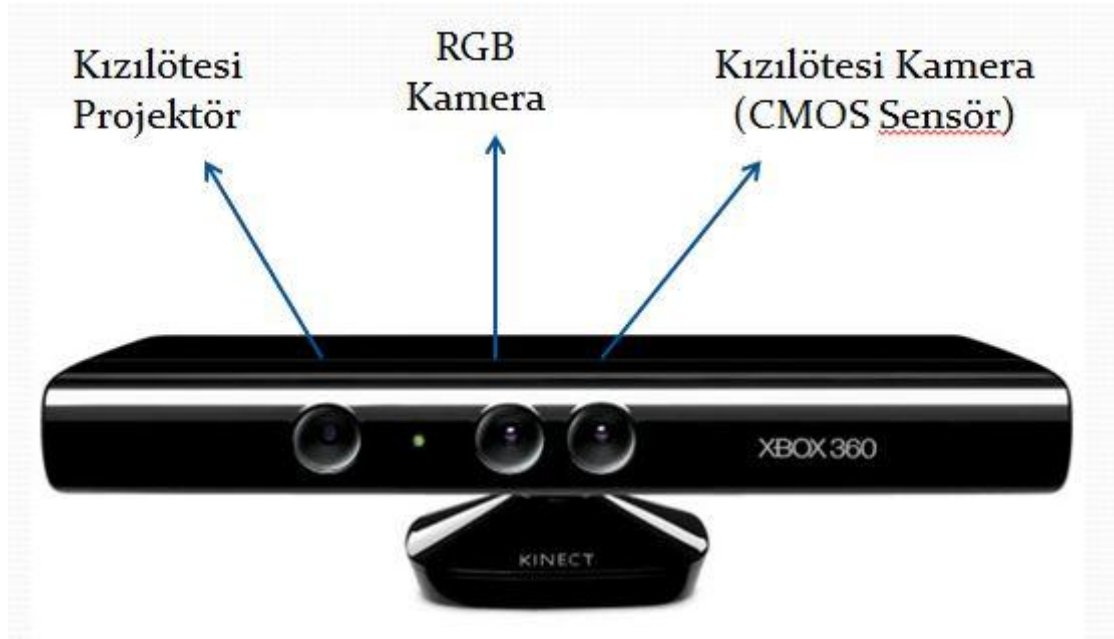
Kinect, oyuncuların herhangi bir kontrol kolu olmadan, interaktif bir biçimde oyun oynayabilmelerini amaçlayan bir donanımdır (Şekil 4.1).

Kontrol kolu olmadan oynanış biçimi, yeni bir çağ açmıştır. İnsanlar, Kinect'i sadece oyunlarda değil, gündelik hayatlarına uyarlamışlardır.

Kinect, genel olarak kızılötesi projektör, kızılötesi kamera ve rgb kameradan oluşur (Şekil 4.2).



Şekil 4. 1 Kinect Sports oyununun oynanışı

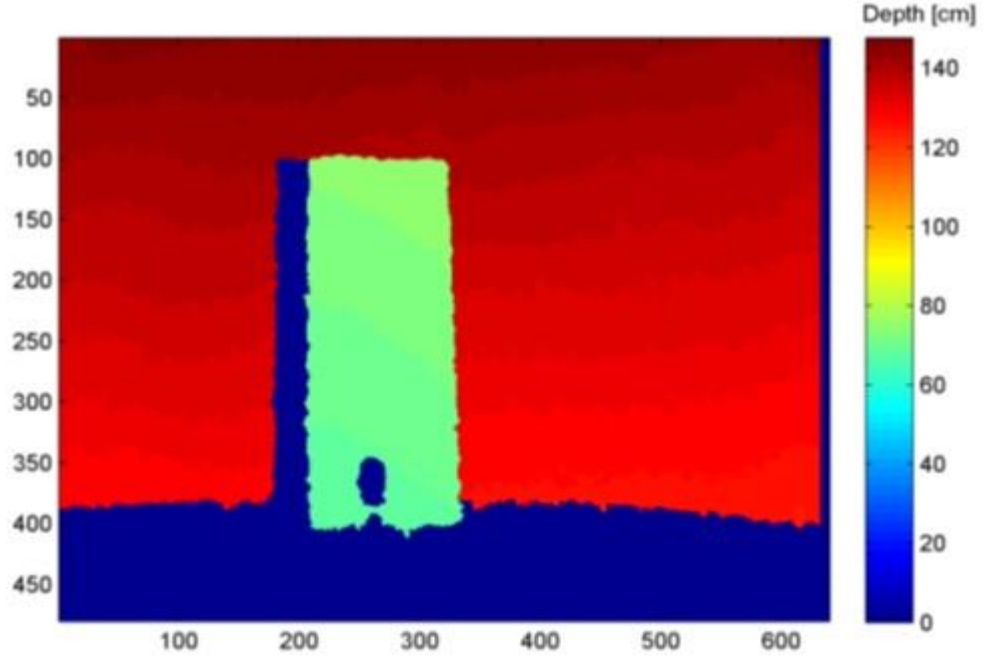


Şekil 4. 2 Microsoft Kinect

Kinect, aynı zamanda ses tanımlama ve konum belirleme amacıyla, bünyesinde 4 dizi mikrofona sahiptir.

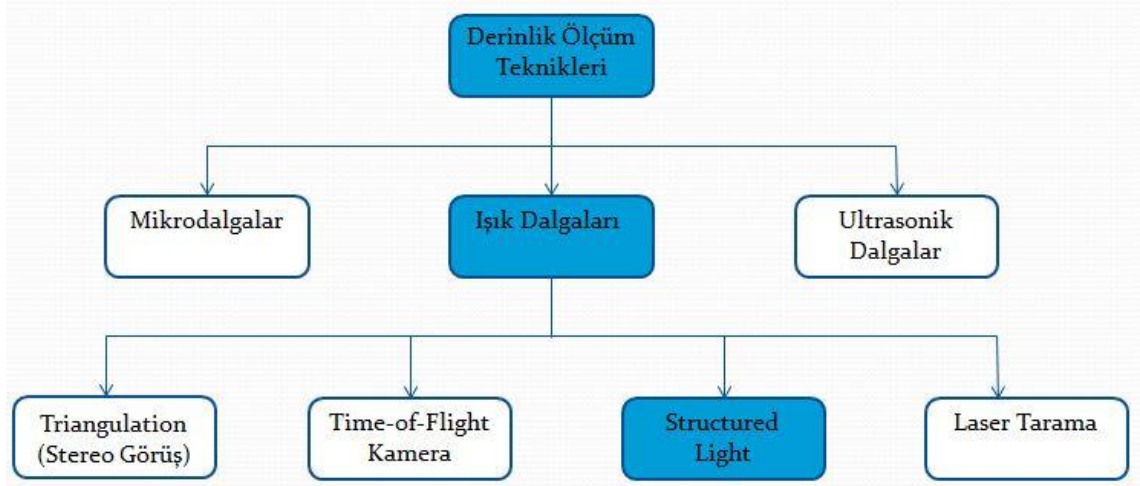
4.2 Derinlik Haritaları

Derinlik haritası, her bir piksel için, kamera sensöründen en yakın obje yüzeyine olan uzaklık bilgisini içerir [35] (Şekil 4.3).



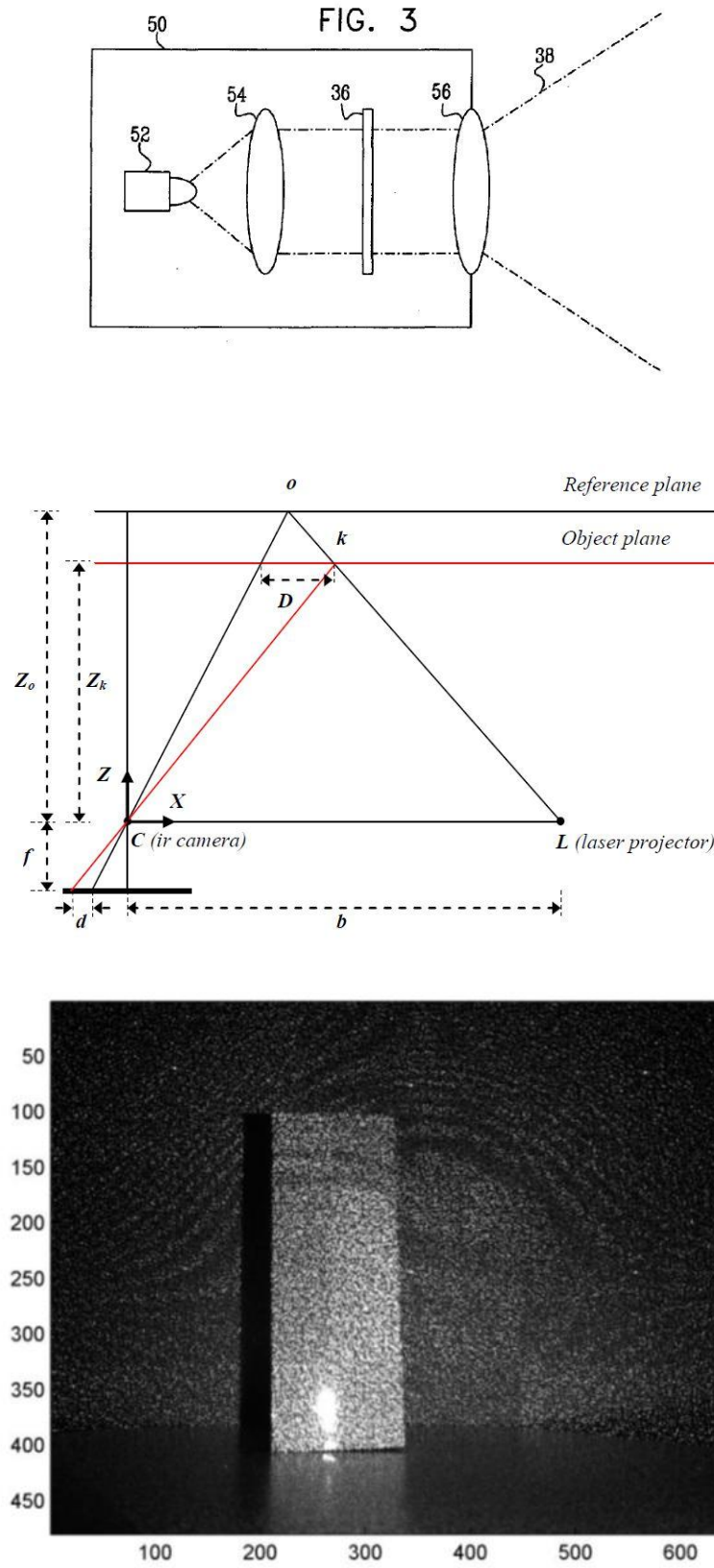
Şekil 4. 3 Derinlik haritası [36]

Kinect, diğer derinlik ölçme yöntemlerine göre hem ucuzdur hem de bu veriyi anlık olarak 30 FPS hızla sunar. Kinect, derinlik haritasını oluşturmak için Structured Light yöntemini kullanır (Şekil 4.4).



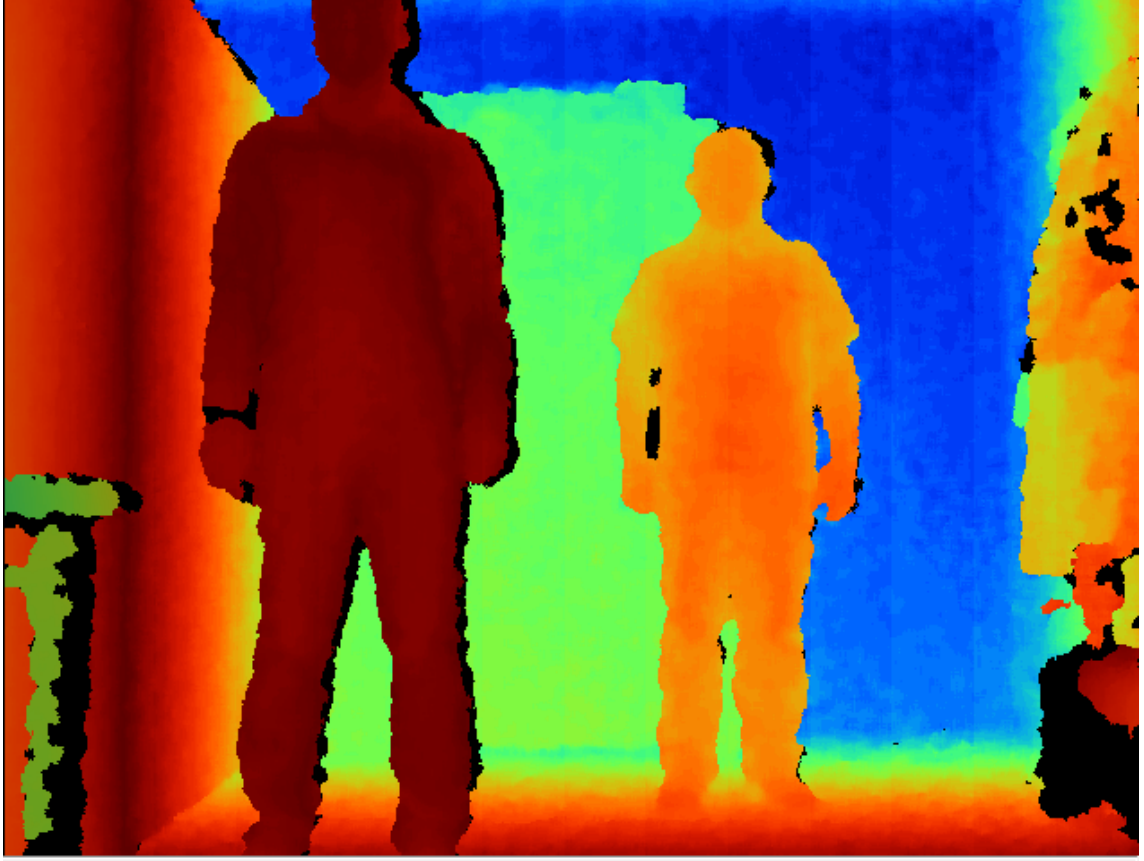
Şekil 4. 4 Derinlik ölçüm teknikleri

Kızılötesi projektör ile ortama noktalardan oluşan referans bir yüzey yansıtılır (Şekil 4.5a). Yansıtılan bu yüzey kızılötesi kamera tarafından algılanır. Fakat, bu algılama sırasında, projektör ve kamera arasındaki mesafeden dolayı paralaks oluşur (Şekil 4.5b). Bu paralaks üç boyutlu benzerlik dönüşümüyle giderilerek, kızılötesi görüntü elde edilir (Şekil 4.5c).



Şekil 4. 5 (a) Kızılötesi projektör kesiti – üstte, (b) Paralaks oluşum – ortada, (c) Kızılötesi görüntü – altta [36]

Görüntüdeki bir nokta, yansıtılan yüzeydeki nokta ile normalize edilmiş çapraz korelasyon yöntemi kullanılarak eşleştirme yapılır ve 3 boyutlu triangulasyon uygulanır. Bu işlem görüntüdeki her nokta için uygulanarak derinlik haritası oluşturulur (Şekil 4.6).

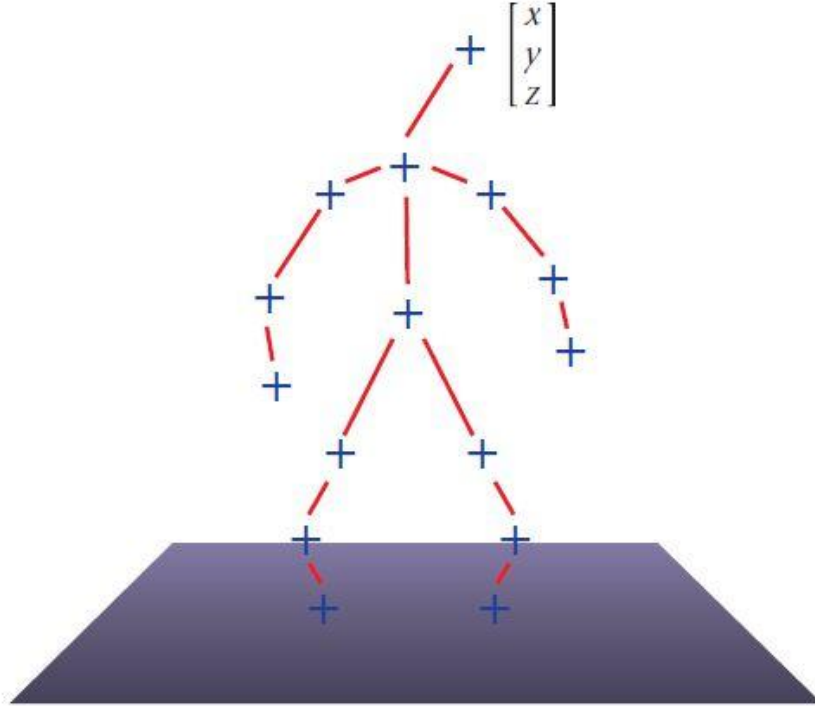


Şekil 4. 6 Derinlik haritası (kırmızı: yakın, mavi: uzak)

Derinlik haritalarındaki hata kaynakları üçe ayrılabilir. Bunlardan birincisi sensörden kaynaklı hatalardır. Sensör belli zaman aralıklarında kalibre edilmelidir. İkinci olarak, sensörün objeye olan uzaklığı da bir hata etkenidir. Kinect'in tavsiye edilen kullanım mesafesi 1,5 metre ile 5 metre arasındadır. Tavsiye edilen maksimum kullanım mesafesi olan 5 metrede 8 cm hata ortaya çıktığı çalışmalarda tespit edilmiştir [30]. Üçüncü hata kaynağı ise obje yüzeyidir. Pürüzsüz ve parlak yüzeyler, kızılötesi görüntüde aşırı aydınlatılmış (overexposed) görünerek, nokta bulutunda bir boşluğa sebep olurlar. Şekil 3'te üçüncü hata tipine rastlanabilir. Şekil 5c'de görüldüğü gibi kızılötesi görüntüde parlak olan yüzey, Şekil 3'teki derinlik haritasında boşluk oluşmasına sebep olmuştur.

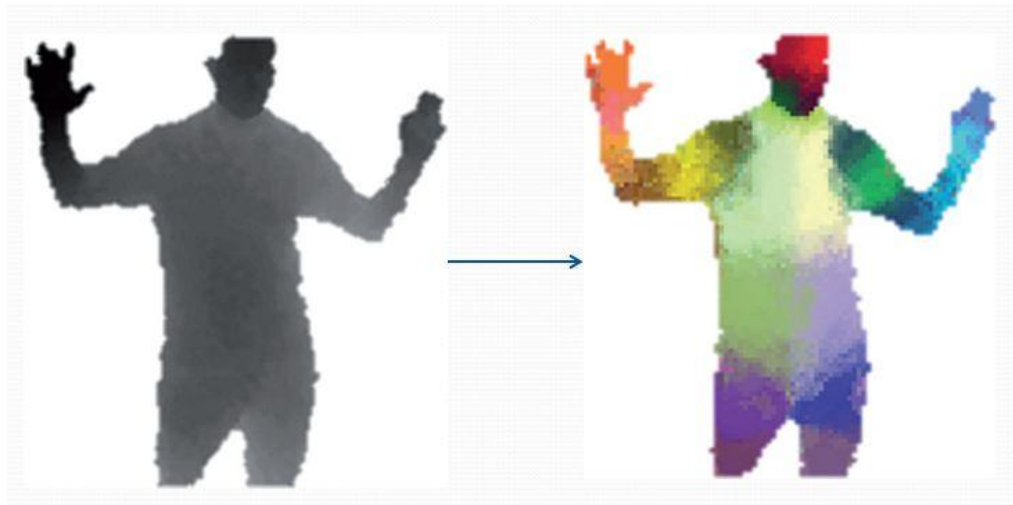
4.3 İskelet Çıkarımı

Kinect'in çalışma prensibinde insan vücudu, vücut bağlantı noktaları ve bu noktaları birbirine bağlayan çizgilerden oluşur (Şekil 4.7).



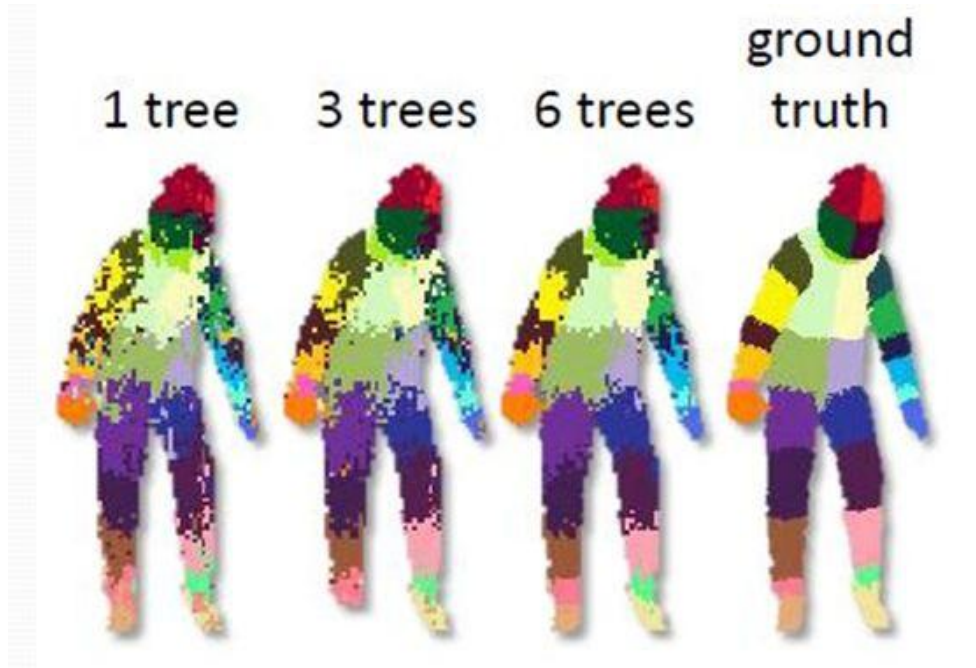
Şekil 4. 7 Kinect tarafından algılanan iskelet [37]

İskeleti, yani vücut bağlantı noktalarını elde etmek için, arka planın çıkarıldığı derinlik görüntüsünde, karar ağaçları yöntemiyle piksel piksel sınıflandırma yapılarak, vücut segmentlere ayrılır (Şekil 4.8).



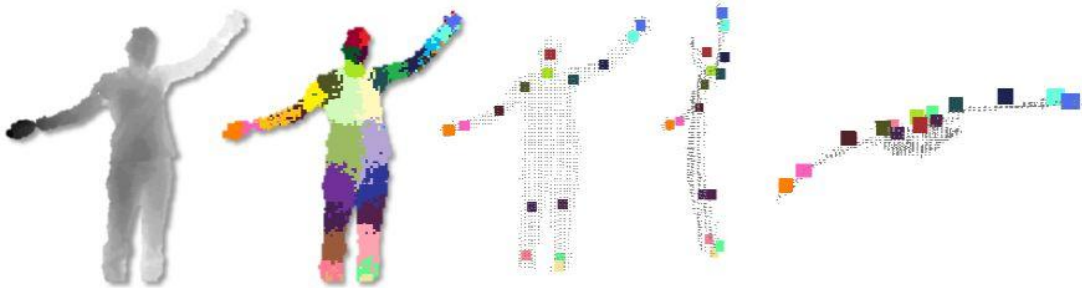
Şekil 4. 8 Sınıflandırma ile vücudun segmentlere ayrılması [37]

Sınıflandırma aşamasında, kullanılan karar ağacı sayısı, elde edilen segmentasyonun doğruluğu direkt olarak etki etmektedir (Şekil 4.9).



Şekil 4. 9 Kullanılan karar ağacı sayısının vücut segmentasyonuna etkisi [38]

Segmentasyondaki her piksel için bir yoğunluk hesabı yapılır. Hesaplanan yoğunluk ile ortalama kayma algoritmasına dayanan ağırlıklı Gauss kernel kullanılarak, segmentlerin 3 boyutlu merkezleri bulunur (Şekil 4.10).

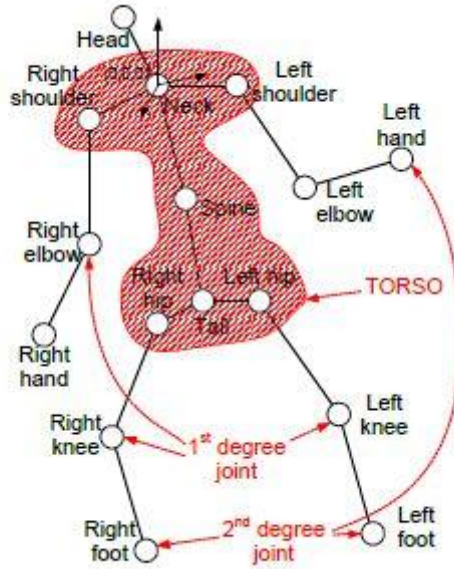


Şekil 4. 10 Üç boyutlu vücut birleşim noktaları [38]

4.4 Hareket Tanımlama

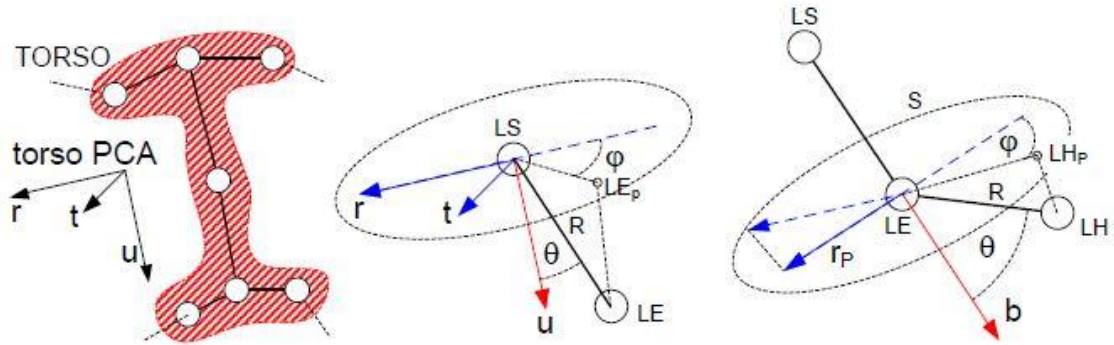
Vücut birleşim noktalarının her biri derecelendirme işlemine tabi tutulur. Boyun, sağ omuz, sol omuz, omurga, kuyruk sokumu, sağ kalça ve sol kalça noktalarından oluşan kısım gövde olarak adlandırılır. Direkt olarak gövdeye bağlı olan sağ dirsek, sol dirsek, sağ diz ve sol diz noktaları, 1. derece bağlantı noktaları olarak adlandırılır. 1. Derece

bağlantı noktalarına bağlı olan, sol el, sağ el, sol ayak ve sağ ayak noktaları 2. derece bağlantı noktaları olarak adlandırılır (Şekil 4.11).



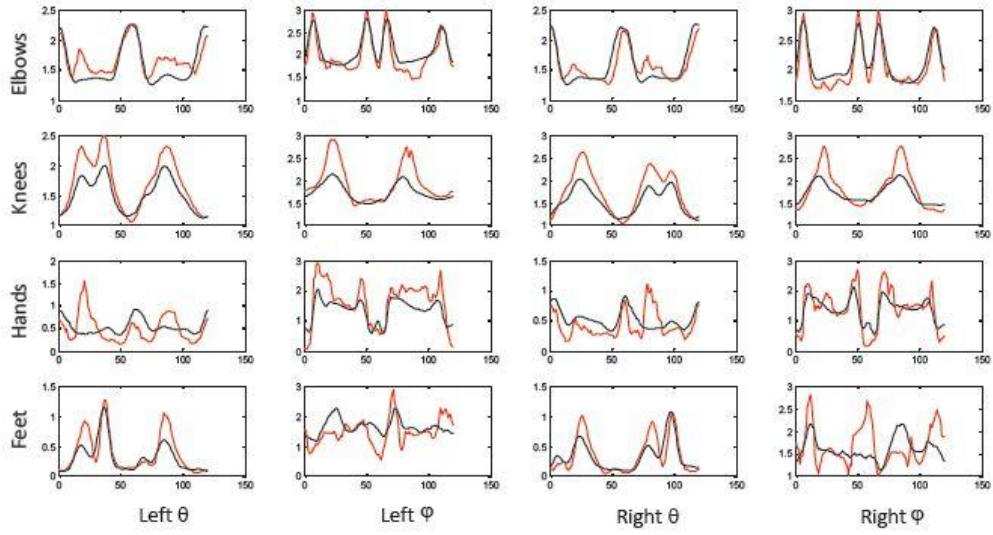
Şekil 4. 11 Gövde, 1. ve 2. derece bağlantı noktaları [39]

1. ve 2. Derece bağlantı noktaları, gövde tabanlı bir koordinat sisteminde tanımlanırlar. İki tip bağlantı noktasının da R , θ ve ϕ olmak üzere 3 parametresi vardır (Şekil 4.12).



Şekil 4. 12 Gövde, 1. ve 2. derece bağlantı noktalarının koordinat sisteminde tanımlanması [39]

Her bir bağlantı noktasının bir koordinat sisteminde tanımlanmasıyla birlikte, bir hareket, alt hareketlerin bütünleşmesiyle tanımlanır. Bir dans hareketi örneğinde, çok sayıda deneklere hareket yaptırılarak, bu hareketin grafiği çıkartılır (Şekil 4. 13).



Şekil 4. 13 Hareket grafiği (siyah), kullanıcı grafiği (kırmızı) [39]

Hareket tanımlama aşamasında, kullanıcının anlık grafiği çıkartılarak, yapılmaya çalışılan hareket en çok benzerlik yöntemi ile karşılaştırılarak, bir korelasyon değeri elde edilir. Buna göre, kullanıcının hareketi ne kadar iyi yaptığı bilgisi elde edilir.

Hareket tanımlamada bu yöntem ile elde edilen ortalama sınıflandırma doğruluğu %96, hızlanma ölçülerini girdi olarak alan yöntem ise %63.32' dir.

BÖLÜM 5

UYGULAMA

Bu çalışmada, Yıldız Teknik Üniversitesi Davutpaşa Kampüsünün tanıtılması amaçlanmıştır (Şekil 5.1). Bu tanıtımın kampüs alanında bulunan yollar bazında gerçekleştirilmesi için uygun yöntem araştırılması sonucunda Google Sokak Görünümü uygulamasına benzer bir çalışmanın gerçekleştirilmesi hedeflenmiştir. Google'ın Sokak görünümü uygulaması ülkemizde aktif değildir. Uygulamanın internet tabanlı olması üniversitemizin tanıtımın yapılmasında büyük fayda sağlayacaktır.

5.1 Uygulama Alanı

Bu çalışmada, Yıldız Teknik Üniversitesi Davutpaşa Kampüsünde Sokak Görünümü oluşturulması amaçlanmıştır (Şekil 5.1). Google Sokak Görünümü servisi henüz ülkemizde hizmet vermediği için, bu servisin bir kampüs için gerekliliği bu çalışmanın önemini arttırmaktadır.



Şekil 5. 1 Yıldız Teknik Üniversitesi Davutpaşa Kampüsü

5.2 Veri Toplama

Kampüs alanının sokak görünümü için yollar boyunca fotoğraflara gereksinim duyulmaktadır. Bu fotoğrafların alım işlemi Point Grey Research markasına ait Ladybug 2 model 360 derece görüş kamerasıyla gerçekleştirilmiştir. Bu kamera, 6 adet balık gözü lense sahiptir. Bu lensler Sony marka ICX204 isimli sensör bulunmaktadır. ICX204, köşegen olarak 6 mm CCD kare pikseli görüntü sensörüdür. İlerlemeli tarama özelliğiyle, bütün piksel sinyallerinin çıkış verileri bağımsız olarak elde edilebilir. Yüksek çözünürlük ve yüksek renk üretkenliğini Kırmızı Yeşil Mavi (RGB) ana renk mozaik filtrları kullanarak sağlar. Lens ve kamera ile ilgili diğer bilgiler Çizelge 5.1’de bulunabilir.

Çizelge 5. 1 Ladybug 2 kamera özellikleri

	Ladybug 2
Sensor	Sony ICX204 CCD
Sensör Çip Boyutu	5.80mm × 4.92 mm
Odak Uzaklığı	2.5 mm
Piksel Büyüklüğü	4.65 µm
Toplam Piksel Sayısı	1077 × 788 (850.000)
Etkili Piksel Sayısı	1034 × 779 (800.000)
Aktif Piksel Sayısı	1024 × 768 (790.000)
FPS	15 FPS (Uncompressed) 30 FPS (Compressed)
Görüş Açısı	Kürenin %75 ‘inden fazlası

Çekim anında balık gözü lenslerin distorsiyon düzeltmelerinden dolayı odak uzaklığı değişiklik gösterebilmektedir. Lenslerin çekim anındaki odak uzaklık değerleri ise Çizelge 5.2 de görölmektedir.

Çizelge 5. 2 Lenslerin çekim anındaki odak uzaklıkları

	Odak Uzaklığı _{min} (mm)	Odak Uzaklığı _{max} (mm)
Kamera 0	2.48677	3.26186
Kamera 1	2.48931	3.26520
Kamera 2	2.51163	3.29448
Kamera 3	2.50038	3.27971
Kamera 4	2.52859	3.31672
Kamera 5	2.53391	3.32371

Fotoğrafların alımı, Yıldız Teknik Üniversitesi Davutpaşa Kampüsü'nde 3 Şubat 2009 tarihinde gerçekleştirilmiştir (Şekil 5.2).



Şekil 5. 2 Fotoğraf alım yolları

Ladybug görüntüleri, .pgr uzantılı dosyalar olarak kaydedilir. Her dosyanın veri kapasitesi 2 GB'dır. Bu dosya Çizelge 5.3 verilen bilgileri içerir.

Çizelge 5. 3 PGR dosyası içeriği

İsim	Açıklama
İmza	Ladybug Dosyası İmzası
Dosya Başlığı	Dosya Hakkında Bilgi
Kalibrasyon Bilgisi	Kamera Kalibrasyon Bilgileri
Görüntü İlk	İlk Ladybug Görüntüsü
Görüntü Son	Son Ladybug Görüntüsü
Özet GPS Verisi	Dosyadaki Tüm Görüntüler için GPS Verisi

5.3 Görüntülerin İşlenmesi

Elde edilen .pgr uzantılı dosyalar, kamera üreticisine ait LadybugCapPro yazılımında işlenmiştir.

Google Sokak Görünümü servisinde 2:1 en boy oranında küresel (equirectangular) projeksiyona yansıtılmış görüntüler kullanılabilmektedir. Bu nedenle, öncelikle projeksiyon tipi küresel olarak belirlenmiştir.

Yazılım 2:1 en boy oranında, 8000 x 4000, 5700 x 2700, 2048 x 1024 gibi farklı çözünürlük seçenekleri sunmaktadır. Aynı şekilde, JPG, BMP, TIFF gibi farklı dosya türü seçenekleri ve farklı renk işleme algoritmaları sunmaktadır. Tüm bu etkenler doğrultusunda, görüntülerin boyutu ve işleme süresi değişiklik göstermektedir. Oluşturulacak olan sokak görünümü sistemi, internet üzerinden yayınlanması planlanan bir sistem olduğu için, sistemdeki görüntülerin görsellik kalitesi kadar, dosya boyutu da önemlidir. Görüntünün dosya boyutu büyüdükçe, internet sayfasında panoramanın görüntülenme hızı düşücektir. Bu da sistemin akıcılığı açısından olumsuz

bir etkindir. Bu nedenle, en iyi görüntü ve en ufak dosya boyunu elde etmek amacıyla farklı çözünürlük değerlerinde, farklı dosya türlerinin, çeşitli renk işleme algoritmalarıyla testler gerçekleştirilmiştir. Bu testler, Intel Core i5 CPU 2.67 GHz işlemcili, Windows 7 64-bit işletim sistemli bir dizüstü bilgisayarda gerçekleştirilmiştir (Çizelge 5.4).

Renk işleme algoritmasının dosya boyutuna herhangi bir etkisi olmadığı, sadece işlem süresini uzattığı tespit edilmiştir. Bu nedenle, en iyi sonuç veren fakat en uzun süren Rigorous renk işleme yönteminin kullanılmasının uygun olduğu kararına varılmıştır. BMP, TIFF ve PNG uzantılı dosyalar, herhangi bir sıkıştırma işlemine tabi tutulmadığı için, diskte daha fazla yer kaplamaktadırlar. Bu nedenle sıkıştırılmış bir görüntü dosyası uzantısı olan JPG kullanılması uygun görülmüştür. Testte 8000×4000, 5400×2700 ve 2048×1024 çözünürlükleri irdelenmiştir. 8000×4000 ve 5400×2700 çözünürlük değerleri yüksek kalite görüntü sağlamalarına rağmen dosya boyutları büyüktür. 2048×1024 çözünürlük değeri ise düşük boyut sağlamasına rağmen, düşük kalitede görüntü sunmaktadır. Bu nedenle 5400×2700 çözünürlüğünden daha kötü fakat 2048×1024 çözünürlüğünden daha iyi kalite sunan 4096×2048 çözünürlük değeri kullanılması uygun görülmüştür. Bu çözünürlük değeri iyi görüntü kalitesini ufak dosya boyutunda sunabilmektedir.

Çekimler dış ortamda gerçekleştirildiği için, sanal küre boyutu 20 metre olarak kullanılmıştır. Görüntülerde, kameraya yakın objeler çok az sayıda bulunduğu için, karışma genişliğinin 100 olarak alınması herhangi bir soruna sebep olmamıştır. 6 adet görüntünün kenarlarında oluşan, orta kısımlarına göre daha karanlık olma sorunu, parlaklık düzeltme ile giderilmiştir.

Bu görüntü ve renk ayarlarıyla birlikte, kampüs genelinde 447 adet 360 derece küresel görüntü elde edilmiştir. Bu görüntüler EXIF dosyalarında enlem, boylam ve yükseklik olmak üzere GPS bilgileri içermektedir.

Çizelge 5. 4 Çözünürlük, renk işleme ve dosya türü karşılaştırılması

Çözünürlük	Renk İşleme	Dosya Türü	Süre (s)	Boyut (mb)
8000 x 4000	High Quality Linear	JPG	16.253	2.64
8000 x 4000	Rigorous	BMP	17.301	91.5
5400 x 2700	Directional Filter	TIFF	8.736	83.4
5400 x 2700	High Quality Linear	PNG	15.927	11.2
5400 x 2700	High Quality Linear	JPG	7.503	1.32
2048 x 1024	Nearest Neighbor	JPG	0.280	0.22
2048 x 1024	High Quality Linear	JPG	0.359	0.22
2048 x 1024	Directional Filter	JPG	0.592	0.21
2048 x 1024	Rigorous	JPG	1.185	0.21
2048 x 1024	Directional	PNG	1.700	2.04
2048 x 1024	Rigorous	PNG	2.356	2.06
2048 x 1024	Directional Filter	BMP	0.592	6.00
2048 x 1024	Rigorous	BMP	1.154	6.00

5.4 Sokak Görünümünün Oluşturulması

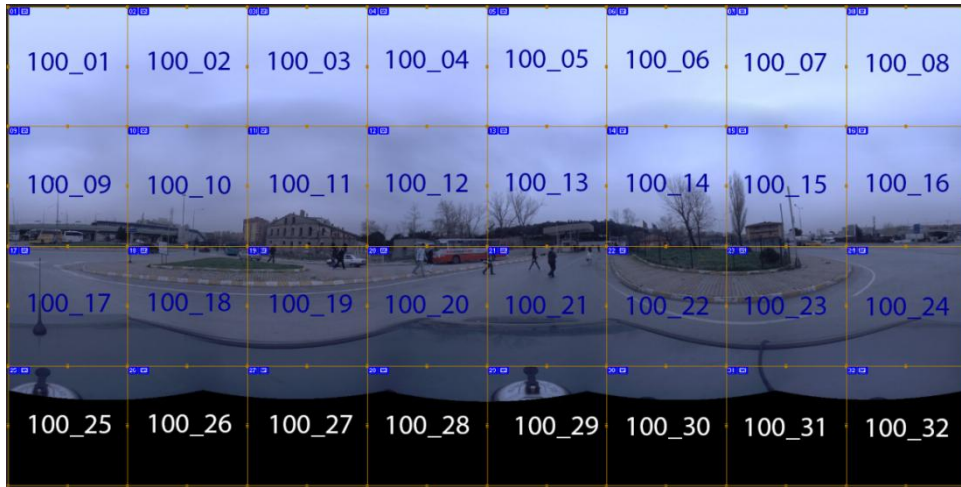
Görüntülerin her birine kampüste bulunduğu konuma göre kimlik numaraları (ID No) atanmıştır. Numaralandırma işlemi Çizelge 5.5'te görüldüğü gibi yapılmıştır.

Kullanılacak olan görüntülerin döşeme parçalarının oluşturulması için Adobe Photoshop CS6 yazılımı kullanılmıştır. Bu yazılımda 2048 x 1024 piksel boyutunda olan her bir görüntü, 256x256 piksel boyutundaki 32 adet parçaya dönüştürülerek, toplam 8160 adet görüntü parçası elde edilmiştir (Şekil 5.3).

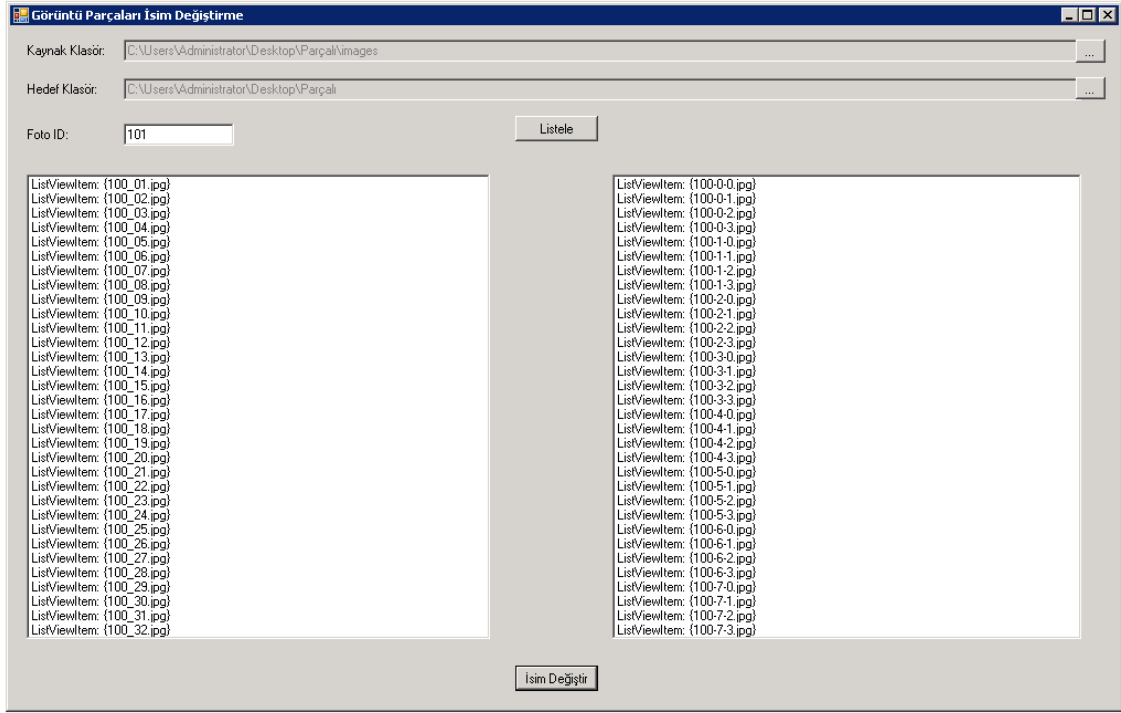
Çizelge 5. 5 Görüntülerin numaralandırılması

Yol	Numara
Ana Giriş – Fen Edebiyat Fak.	100 - 149
Yabancı Diller MYO - Kavşak	200 - 212
İnşaat Fak. – Yabancı Diller MYO	300 - 342
Kavşak – Kimya Met. Fak.	400 - 417
Kimya Met. Fak. – Elektrik Fak.	500 - 529
Kimya Met. Fak. – İnşaat Fak.	600 - 639
Elektrik Fak. – İnşaat Fakültesi	700 - 732
Arka Giriş – Elektrik Fak.	800 - 827

Elde edilen görüntü parçalarını programlanabilir ve koordinatlı bir biçimde isimlendirilmesi gerekmektedir. Fakat, Photoshop CS6 yazılımı her parçayı otomatik olarak isimlendirmeye olanak sağlamamaktadır (Şekil 5.3). Bu nedenle, 8160 adet görüntü parçasını isimlendirmek için, özgün bir program geliştirilmiştir. (Şekil 5.4). Bu program, kaynak klasördeki “100_01, 100_02, 100_03,...” düzeninde isimlendirilmiş olan görüntü parçalarını, “100-0-0, 100-0-1, 100-0-2,...” düzeninde isimlendirerek, hedef klasöre kaydedilmesini sağlar. (Şekil 5. 5).

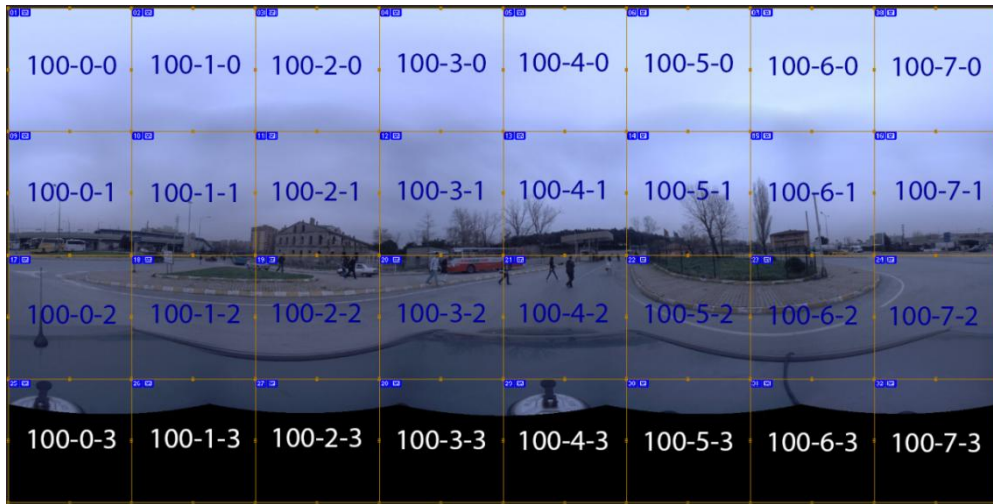


Şekil 5. 3 100 Kimlik numaralı görüntü parçaları



Şekil 5. 4 İsim değiştirme programı arayüzü

Google Maps Javascpit API ile yazılan sokak görünümleri, güvenlik sebepleri sebebiyle yerel ağlarda çalışmamaktadırlar. Bu nedenle, öncelikle “www.tolgabakirman.com” şeklinde bir alan adı elde edilerek, görüntüler parçalı ve bütün olarak, ayrı klasörler halinde internet ortamına aktarılmıştır.



Şekil 5. 5 100 Kimlik numaralı görüntü parçaları son hali

Sokak görünümü, metin belgesi üzerine yazılan javascript kodlarının, html formatında kaydedilmesiyle oluşturulmuştur. Öncelikle html dosyasının dili, dil karakter seti ve stil seti belirlenmiştir (Şekil 5.6).

```

1 <html>
2
3 <head>
4 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5 <meta http-equiv="content-language" content="en" />
6 <meta http-equiv="Content-Style-Type" content="text/css" />
7

```

Şekil 5. 6 Html dosyasının dil, karakter ve stil seti belirlenmesi

Google Maps Javascript API'a referans ataması yapılmıştır. HTML dosyasının düzeni ve kaynak görünümü açısından, sokak görünümü penceresi ve görünüm stili ayrı dosyalar olarak kayıt edilmiştir (Şekil 5.7).

```

8 <title>Davutpasa Kampusu Sokak Görünümü</title>
9 <script src="http://maps.googleapis.com/maps/api/js?sensor=false" type="text/javascript"></script>
10 <script src="js/map_code.js" type="text/javascript"></script>
11 <link href="Style/main.css" rel="stylesheet" type="text/css" />
12 </head>
13
14 <body>

```

Şekil 5. 7 Referanslar

Stil görünümü ile arka plan rengi ve arka plan resmi gibi özellikleri belirlenmiştir. Arka plan rengi özelliği, her tarayıcı tarafından farklı algılandığı için, bu kod her tarayıcı için ayrı ayrı tanımlanmıştır (Şekil 5.8).

```

1 body
2 {
3     margin:15px 0px 0px 0px ;
4     /*background-color:#6E6E6E;*/
5
6     /* IE10 Consumer Preview */
7     background:#6E6E6E -ms-linear-gradient(top,#DBDBDB 0%, #6E6E6E 100%) no-repeat;
8
9     /* Mozilla Firefox */
10    background:#6E6E6E -moz-linear-gradient(top, #DBDBDB 0%, #6E6E6E 100%) no-repeat;
11
12    /* Opera */
13    background:#6E6E6E -o-linear-gradient(top, #DBDBDB 0%, #6E6E6E 100%) no-repeat;
14
15    /* Webkit (Safari/Chrome 10) */
16    background:#6E6E6E -webkit-gradient(linear, left top, left bottom, color-stop(0, #DBDBDB), color-stop(100%, #6E6E6E)) no-repeat;
17
18    /* Webkit (Chrome 11+) */
19    background:#6E6E6E -webkit-linear-gradient(top, #DBDBDB 0%, #6E6E6E 100%) no-repeat;
20
21    /* W3C Markup, IE10 Release Preview */
22    background:#6E6E6E linear-gradient(to bottom, #DBDBDB 0%, #6E6E6E 100%) no-repeat;
23 }
24
25 #Harita
26 {
27     background-image:url('bg.png');
28     background-position:center;
29     background-position:top;
30     background-repeat:no-repeat;
31 }

```

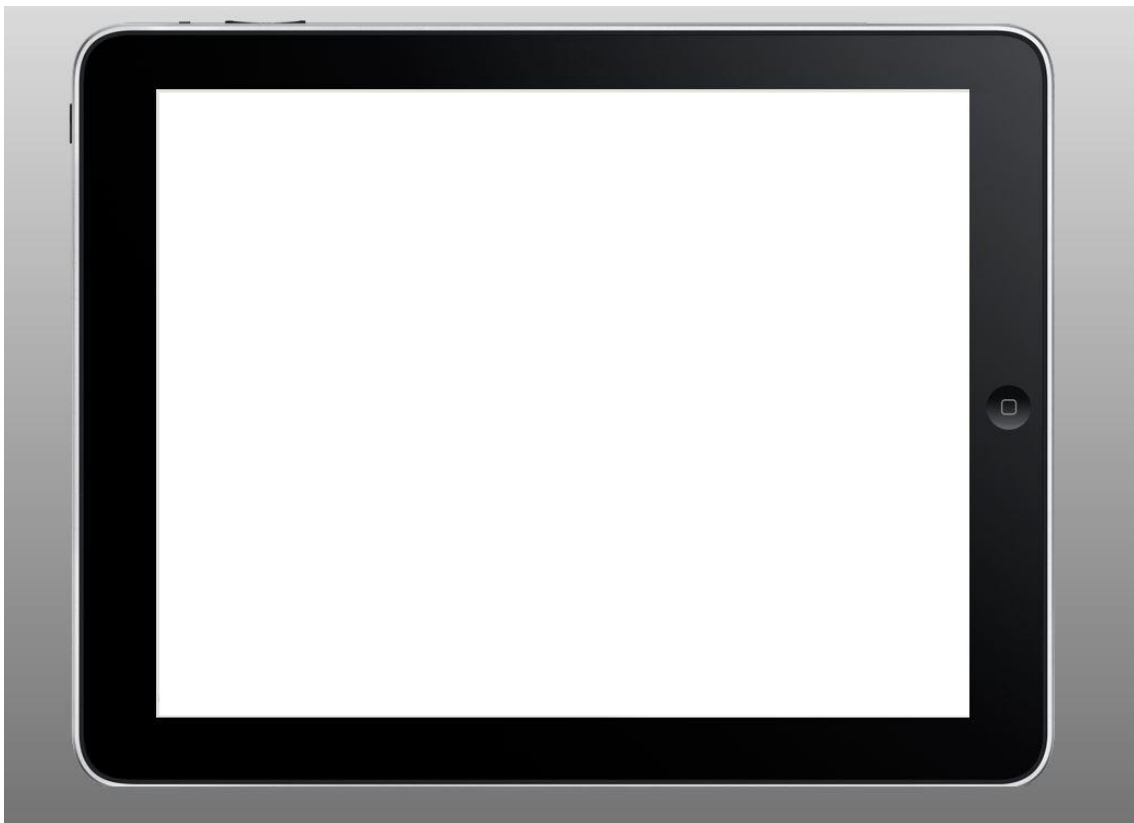
Şekil 5. 8 Main.css dosyası

HTML dosyasında referansların tanımlanması işlemi tamamlandıktan sonra, arka plan resminin tasarımı ve büyüklüğü gibi değişkenler belirlenmiştir (Şekil 5.9).

```
16 <table width="100%" cellpadding="0" cellspacing="0"><tr height="700px"><td id="Harita" align="center">
17
18     <div id="HaritaGorunumu" style="width:743px; height:574px; font-size:15px;" ></div>
19
20 </td></tr></table>
21 </body>
22 </html>
```

Şekil 5. 9 Sayfa tasarımı

Bu çalışmada kullanılan Google Maps API for Javascript uygulamanın mobil aygıtlarda da çalışmasına olanak vermektedir. Bu nedenle, sistemin bilgisayar, tablet ve telefon gibi tüm aygıtlarda rahatça kullanılabilmesi için sade bir görünüm tercih edilmiştir. Sayfa tasarımının tamamlanmış hali Şekil 5.10' da görülebilmektedir.



Şekil 5. 10 Sayfa görünümü

Map_code.js dosyası görüntüler parçalı ve bütün olarak kullanılarak, iki kopya oluşturulmuştur. Öncelikle sokak görünümü değişkeni atanarak, sokak görünümü sağlayıcısı ayarları belirlenmiştir (Şekil 5.11). Bu ayarlarda, yakınlaştırma seviyesi 0,

sokak görünümü sağlayıcısı kişisel, başlangıç görüntü kimlik numarası 100 olarak atanmıştır. Fare tekerleği ve yakınlaştırma kontrolü özelliği kapalı tutulmuştur.

```
1    var streetView;
2
3    function initialize() {
4        // Sokak Görünümü Sağlayıcısı Ayarları
5        var streetViewOptions = {
6            zoom: 0,
7            panoProvider: getCustomPanorama,
8            pano: "100",
9            clickTogo: true,
10           scrollwheel: false,
11           zoomControl: false
12       };
13
```

Şekil 5. 11 Sokak görünümü sağlayıcısı ayarları

Sokak görünümü objesi oluşturularak, html dosyasındaki çerçeveye atanmıştır ve ayarlarını yukarıda belirlenen sokak görünümü sağlayıcısı ayarlarından alması sağlanmıştır (Şekil 5. 12).

```
// Sokak Görünümü Objesinin Oluşturulması
var streetViewDiv = document.getElementById('HaritaGorunumu');
streetView = new google.maps.StreetViewPanorama(streetViewDiv, streetViewOptions);
```

Şekil 5. 12 Sokak görünümü objesinin oluşturulması

Görüntüler arasında geçiş için bağlantı objesi tanımlanmıştır (Şekil 5. 13).

```
// Bağlantı Objesinin Oluşturulması
google.maps.event.addListener(streetView, "links_changed", createCustomLink);
```

Şekil 5. 13 Bağlantı objesinin oluşturulması

Sonraki aşamada, görüntülerin bulunduğu dosya yolunu getiren ve görüntü dosyasını alan iki fonksiyon tanımlanmıştır. Görüntülerin parçalı ve bütün olarak kullanıldığı iki ayrı sistemdeki, kodlardaki tek fark, bu iki fonksiyonun tanımlanış biçimidir. Görüntülerin bütün olarak kullanıldığı sistemde, görüntü dosya yolunu getiren fonksiyonda görüntülerin bulunduğu "images4096" klasörü gösterilmiştir. Görüntü dosyasının alan fonksiyonda, görüntü parça boyutu, görüntü boyutuyla eşit tutulmuştur (Şekil 5.14).

```

function getCustomPanoramaTileUrl(panoID, zoom, tileX, tileY) {
    // Görüntü Yolu
    return "../images4096/" + panoID + '.jpg';
}

function getCustomPanorama(panoID) {
    var streetViewPanoramaData = {
        links: [],
        copyright: 'Imagery (c) Tolga Bakirman',
        tiles: {
            tileSize: new google.maps.Size(4096, 2048),
            worldSize: new google.maps.Size(4096, 2048),
            centerHeading: 0,
            getTileUrl: getCustomPanoramaTileUrl
        }
    };
};

```

Şekil 5. 14 Bütün görüntü fonksiyonları

Görüntü parçalarının kullanıldığı sistemde, dosya yolu için öncelikle parçaların bulunduğu “images4096tile512” klasörü belirtilmiş ve görüntü parçalarının, isimlendirildiği formata uygun olarak, “parça-X” ve “parça-Y” koordinatlandırılması, fonksiyon içinde tanımlanmıştır. İkinci fonksiyonda ise, görüntü parçalarının çözünürlüğü, görüntünün 32 eş parçaya bölünmüş hali olan, 512 x 512 olarak tanımlanmıştır (Şekil 5.15).

```

function getCustomPanoramaTileUrl(panoID, zoom, tileX, tileY) {
    // Görüntü Parçaları Yolu
    return "../images4096tile512/" + panoID + '-' + tileX + '-' + tileY + '.jpg';
}

function getCustomPanorama(panoID) {
    var streetViewPanoramaData = {
        links: [],
        copyright: 'Imagery (c) Tolga Bakirman',
        tiles: {
            tileSize: new google.maps.Size(512, 512),
            worldSize: new google.maps.Size(4096, 2048),
            centerHeading: 0,
            getTileUrl: getCustomPanoramaTileUrl
        }
    };
};

```

Şekil 5. 15 Parçalı görüntü fonksiyonları

Fonksiyonların tanımlanma işleminden sonra, her görüntüyü kimlik numarasına göre ayrı bir durum olarak değerlendirmek için bir “switch – case” fonksiyonu

oluşturulmuştur. Bu fonksiyonda her bir görüntünün kimlik numarası için durumlar (case) tanımlanmıştır (Şekil 5.16).

Görüntülerin her biri tanımlandıktan sonra, görüntüler arasında birbirine geçişi sağlamak için bağlantılar eklenmiştir. Bu bağlantılar, Google Maps Javascript API v3 tarafından ok şeklindeki bir buton olarak tasarlanmıştır. Her bir görüntü için, buton yönü, geçişin hangi görüntüye olacağı ve buton açıklamaları tanımlanmıştır (Şekil 5.17). Buton yönünü belirleyen “heading” özelliği, 0 – 360 derece arasında bir değerdir. 0 derece fotoğrafın çekim yönüdür (Şekil 5. 18).

Bağlantıların eklenmesiyle birlikte tamamlanan sokak görünümü penceresi kodu, web sunucusuna yerleştirilmiştir.

Görüntülerin bütün olarak kullanıldığı sisteme, “www.tolgabakirman.com” adresinden, görüntü parçalarının kullanıldığı sisteme ise “www.tolgabakirman.com/tile.html” adresinden erişilebilir (Şekil 5.19)

```

switch (panoID) {
    case "100":
        // Görüntü Açıklamaları
        streetViewPanoramaData["location"] = {
            pano: '100',
            description: "Davutpaşa",
            latLng: new google.maps.LatLng(0, 0)
        };
        return streetViewPanoramaData;

    case "101":
        streetViewPanoramaData["location"] = {
            pano: '101',
            description: "Yıldız Teknik Üniversitesi Davutpaşa Kampusu",
            latLng: new google.maps.LatLng(0, 0)
        };
        return streetViewPanoramaData;

    ~
    .
    .

    case "826":
        streetViewPanoramaData["location"] = {
            pano: '826',
            description: "Arka Giriş Yolu",
            latLng: new google.maps.LatLng(0, 0)
        };
        return streetViewPanoramaData;

    case "827":
        streetViewPanoramaData["location"] = {
            pano: '827',
            description: "Arka Giriş Yolu",
            latLng: new google.maps.LatLng(0, 0)
        };
        return streetViewPanoramaData;
}

```

Şekil 5. 16 Switch – Case fonksiyonu ile durumların tanımlanması

```

function createCustomLink() {
    //Bağlantıları Ekleme
    var links = streetView.getLinks();
    var panoID = streetView.getPano();

    switch (panoID) {

        case "100":
            links.push({
                description: "Yildiz Teknik Universitesi Davutpasa Kampusu",
                pano: "101",
                heading: 0
            });
            break;

        case "101":
            links.push({
                description: "Davutpasa",
                pano: "100",
                heading: 180
            });
            links.push({
                description: "Ana Giriş",
                pano: "102",
                heading: 0
            });
            break;

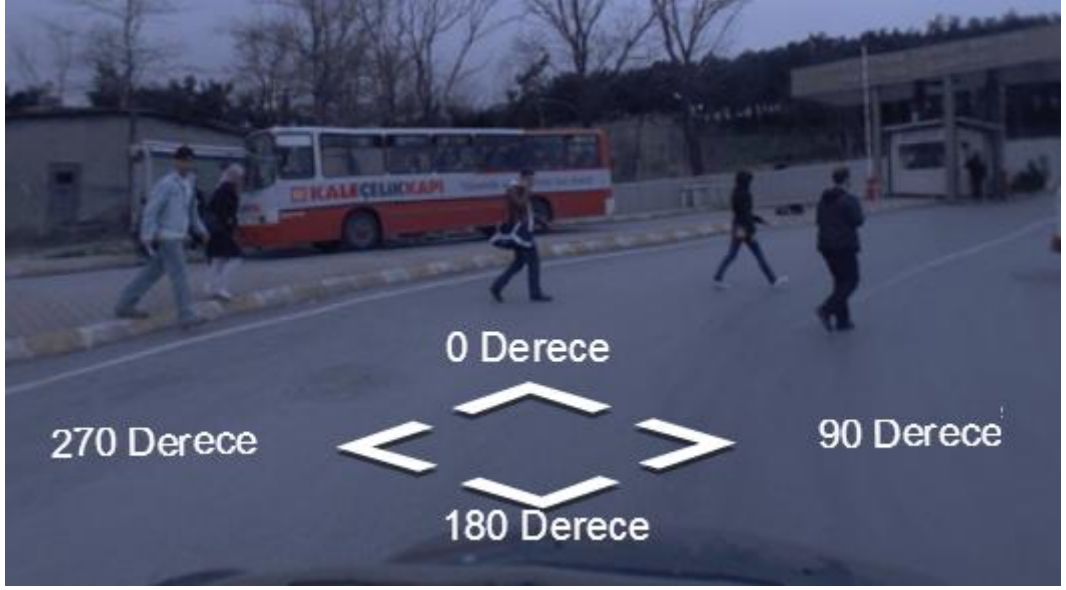
        ~
        .
        .

        case "826":
            links.push({
                description: "Arka Giriş Kapisi",
                pano: "825",
                heading: 180
            });
            links.push({
                description: "Arka Giriş Yolu",
                pano: "827",
                heading: 0
            });
            break;

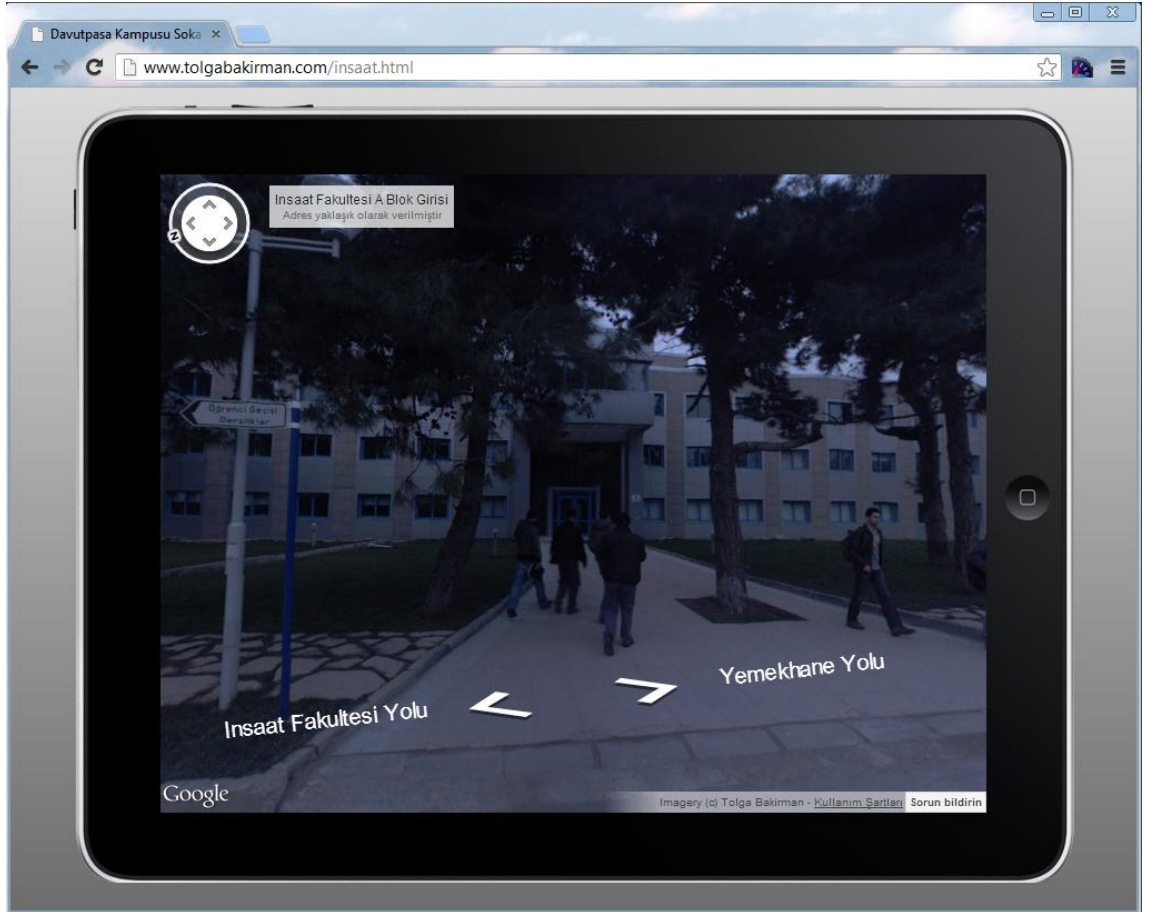
        case "827":
            links.push({
                description: "Arka Giriş Kapisi",
                pano: "826",
                heading: 180
            });
            links.push({
                description: "Elektrik Elektronik Fakültesi",
                pano: "529",
                heading: 0
            });
            break;
    }
}

```

Şekil 5. 17 Fotoğrafları arasındaki geçiş bağlantılarını ekleme



Şekil 5. 18 Heading özelliğinin değer aralığı



Şekil 5. 19 Sayfanın son durumu

5.5 Sokak Görünümü Analizi

Kullanıcıların kampüs içindeki diğer noktalardan da sokak görünümüne geçiş yapabilmesi için, farklı internet sayfaları oluşturulmuştur (Çizelge 5.6).

Çizelge 5. 6 Sokak görünümü internet adresleri

Yer	Web Adresi
Kampüs Ana Girişi	www.tolgabakirman.com
Kavşak	www.tolgabakirman.com/kavsak.html
Yabancı Diller MYO	www.tolgabakirman.com/ydmyo.html
İnşaat Fakültesi	www.tolgabakirman.com/insaat.html
Kimya-Metalurji Fakültesi	www.tolgabakirman.com/kimya.html
Fen-Edebiyat Fakültesi	www.tolgabakirman.com/fefak.html
Sanat Tasarım Fakültesi	www.tolgabakirman.com/sanat.html
Elektrik-Elektronik Fakültesi	www.tolgabakirman.com/elek.html
Yemekhane	www.tolgabakirman.com/ymkhn.html
Kampüs Arka Girişi	www.tolgabakirman.com/arka.html

Sokak görünümü çalışmanın bir önceki bölümünde belirtildiği gibi görüntülerin bütün ve parçalı olarak kullanıldığı iki yöntemle oluşturulmuştur.

Parçalı görüntülerin kullanıldığı sokak görünümü sisteminin web adresleri Çizelge 5.7’de görüldüğü gibidir. Parçalı görüntülerin kullanıldığı sistemde, kullanıcının bilgisayarına sadece görüntülediği bölgede görüntü dosyaları indirilir. Böylece, daha hızlı gezinti imkanı sağlanmış olur.

Çizelge 5. 7 Parçalı görüntü yöntemi internet adresleri

Yer	Web Adresi
Kampüs Ana Giriş	www.tolgabakirman.com/tile.html
Kavşak	www.tolgabakirman.com/tKavsak.html
Yabancı Diller MYO	www.tolgabakirman.com/tYdmyo.html
İnşaat Fakültesi	www.tolgabakirman.com/tInsaat.html
Kimya Metalurji Fakültesi	www.tolgabakirman.com/tKimya.html
Fen Edebiyat Fakültesi	www.tolgabakirman.com/tFefak.html
Sanat Tasarım Fakültesi	www.tolgabakirman.com/tSanat.html
Elektrik Elektronik Fakültesi	www.tolgabakirman.com/tElek.html
Yemekhane	www.tolgabakirman.com/tYmkhn.html
Kampüs Arka Giriş	www.tolgabakirman.com/tArka.html

Her bir yer için bütün ve parçalı sistemin sayfa yüklenme süreleri karşılaştırılmıştır. Sayfa yüklenme süreleri Pingdom adlı web aracı kullanılarak hesaplanmıştır [40]. Pingdom birçok web sitesinin kendi performanslarını ölçme ve geliştirmek amacıyla kullandığı ücretsiz bir web hizmetidir. Bu araç, belirtilen bir web sitesindeki objelerin boyutunu ve yüklenme sürelerini sayısal ve grafiksel olarak sunmaktadır.

Araç üç nokta üzerinden test yapma imkanı sunmaktadır. Bunların içerisinde ülkemize en yakın olan Hollanda üzerinden testler gerçekleştirilmiştir.

Kampüs ana girişi için yapılan testte, bütün görüntülü sistemde, görüntü boyutu 1.0 MB'tır ve yüklenme süresi 1.08 saniye olarak belirlenmiştir. Parçalı görüntü kullanılan sistemde yapılan testte, 28.2 KB, 29.9 KB, 30.8 KB ve 25.0 KB boyutlarında olmak üzere 4 görüntü bulunmaktadır ve bu görüntülerin toplam yüklenme süreleri 0.3 saniye olarak belirlenmiştir. Bu nokta için iki sistemdeki görüntülerin yüklenme süreleri arasında 0.8 saniye fark vardır. Diğer noktalarda gözlemlenen yüklenme süre farkları Çizelge 5.8'de belirtilmiştir.

Çizelge 5. 8 Yükleme sürelerinin karşılaştırılması

Yer	Bütün Sistem Süre	Görüntü Boyutu	Parçalı Sistem Süre	Görüntü Boyutu	Süre Farkı
Ana Giriş	1.08 s	1.0 MB	291 ms	113.9 KB	789 ms
Kavşak	1.22 s	1.1 MB	288 ms	133.7 KB	932 ms
Yabancı Diller	1.15 s	1.1 MB	319 ms	136.0 KB	831 ms
İnşaat Fak.	1.35 s	1.1 MB	322 ms	115.6 KB	1028 ms
Kimya-Met.	1.82 s	1.6 MB	344 ms	164.9 KB	1476 ms
Fen-Edebiyat	1.28 s	1.1 MB	277 ms	104.8 KB	1003 ms
Sanat Tas.	867 ms	1.0 MB	295 ms	97.2 kb	647 ms
Elektrik	1.12 s	1.1 MB	330 ms	103.1 KB	790 ms
Yemekhane	1.36 s	1.1 MB	358 ms	130.0 KB	1002 ms
Arka Giriş	1.13 s	1.1 MB	284 ms	102.5 KB	846 ms

Yapılan analizler sonucunda bütün ve parçalı görüntülerin yüklenme süreleri arasında ortalama 0.9 saniye fark olduğu belirlenmiştir.

5.6 İnternet Tabanlı Coğrafi Bilgi Sisteminin Oluşturulması

İnternet tabanlı CBS kurulumu için, ArcGIS Desktop 10.1 ve ArcGIS for Server 10.1 yazılımları ile ArcGIS API for Silverlight programlama arayüzü kullanılmıştır. Bu sistem ile kullanıcıya belli başlangıç noktaları sunularak, kullanıcının istediği noktadan sokak görünümüne geçiş yapabileceği şekilde tasarlanmıştır.

6.6.1 Verinin Hazırlanması

Öncelikle UTM projeksiyonunda ED50 datumunda bir coğrafi veri tabanı oluşturulmuştur. Bu veri tabanı içerisinde başlangıç noktalarını temsil eden, nokta türünde vektör veri oluşturularak, gerekli öznitelikler belirlenmiştir. 2010 yılına ait ortofoto görüntüsünden yararlanılarak, kampüs içerisindeki önemli noktalar

sayısallaştırılmıştır (Şekil 5.20). Sayısallaştırılan noktalara yer adı ve sokak görünümü öznitelikleri girilmiştir(Şekil 5. 21).



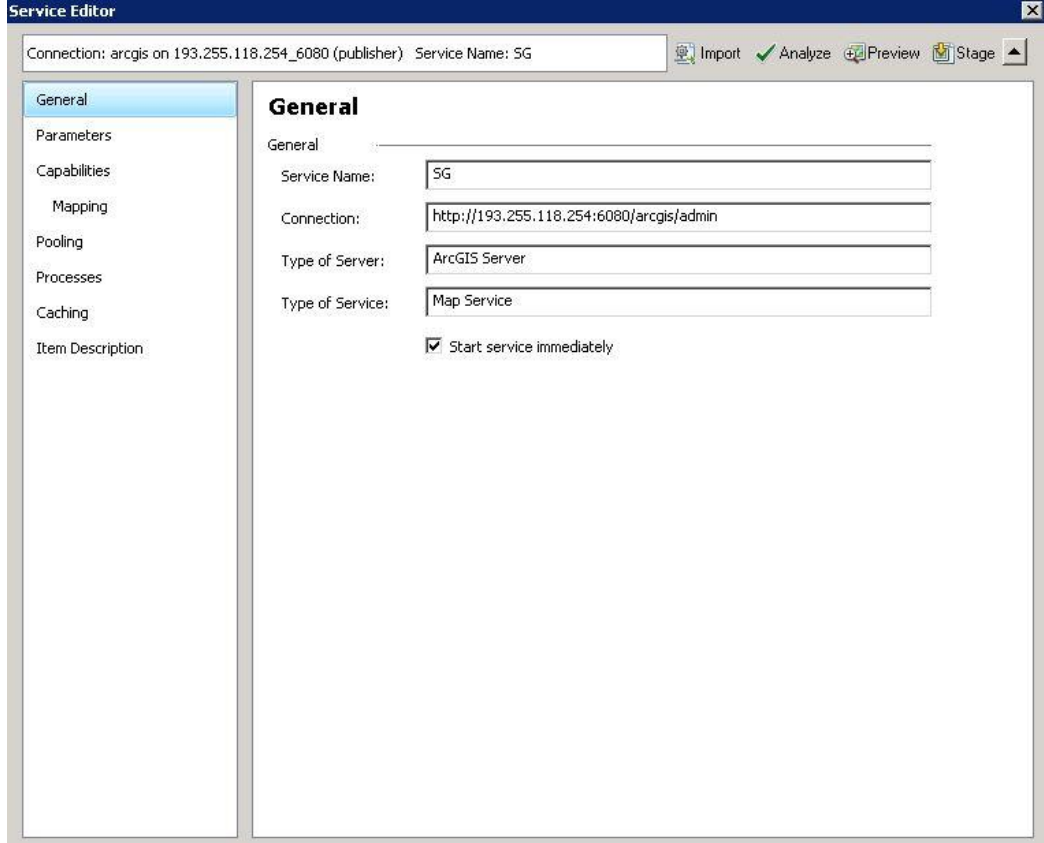
Şekil 5. 20 Sayısallaştırılan noktalar

Table				
Sokak Görünümü				
FID	Shape *	Id	Yer	Sokak Görünümü
0	Point	0	Kavşak	http://www.tolgabakirman.com/kavsak.html
1	Point	0	YTÜ Ana Giriş Kapısı	http://www.tolgabakirman.com
2	Point	0	İnşaat Fakültesi	http://www.tolgabakirman.com/insaat.html
3	Point	0	Yabancı Diller MYO	http://www.tolgabakirman.com/ydmyo.html
4	Point	0	Fen Edebiyat Fakültesi	http://www.tolgabakirman.com/fevak.html
5	Point	0	Kimya Metalurji Fakültesi	http://www.tolgabakirman.com/kimya.html
6	Point	0	Yemekhane Binası	http://www.tolgabakirman.com/ymkhn.html
7	Point	0	Sanat Tasarım Fakültesi	http://www.tolgabakirman.com/sanat.html
8	Point	0	Elektrik Elektronik Fakültesi	http://www.tolgabakirman.com/elek.html
9	Point	0	YTÜ Arka Giriş Kapısı	http://www.tolgabakirman.com/arka.html

Şekil 5. 21 Sokak Görünümü başlangıç noktaları öznitelik tablosu

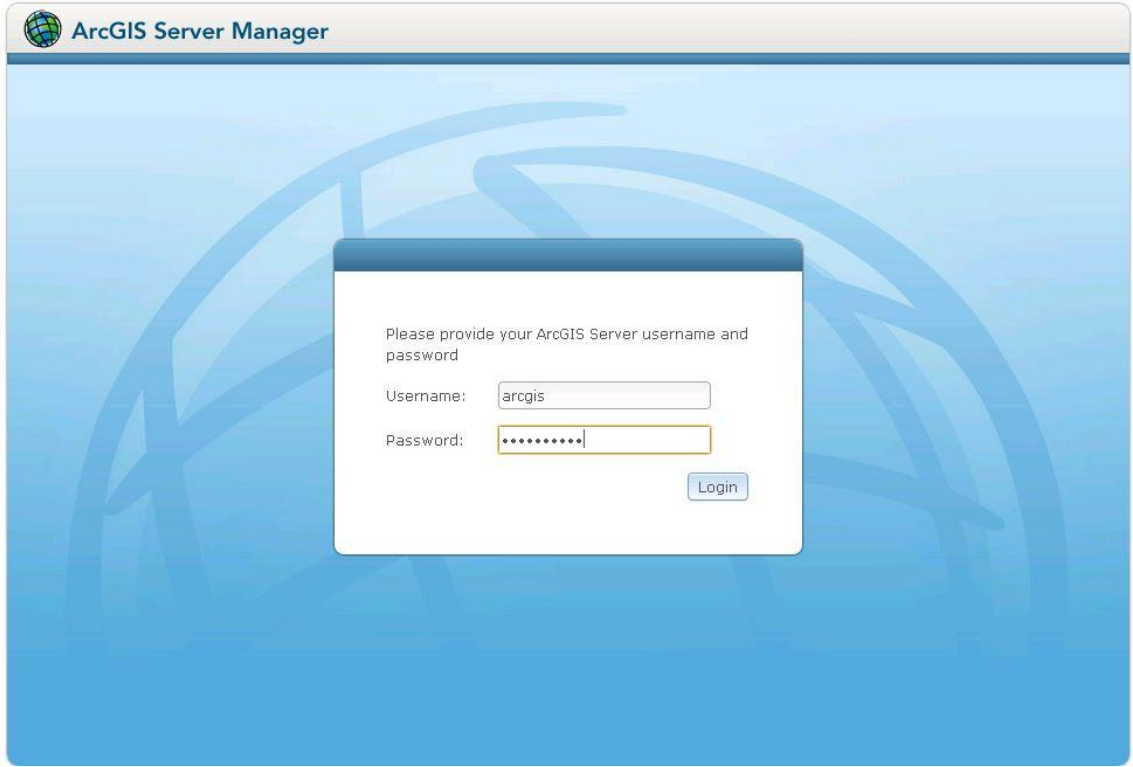
6.6.2 Verinin Servis Edilmesi

ArcMap arayüzü kullanılarak, oluşturulan projenin service definition (.sd) dosyası elde edilmiştir. Bu dosya oluşturulurken, ArcMap bir tür hata ayıklama yaparak, hata kaynaklarını ve çözümleri işaret eder. Bu sayede, serviste oluşabilecek sorunların önüne geçilmiş olur. Oluşturulan bu dosyada, servisin adı, yetenekleri, görünümü, gibi bilgiler bulunmaktadır. (Şekil 5.22)



Şekil 5. 22 Service Definition (.sd) dosyası oluşturma arayüzü

ArcGIS Server Manager, <http://localhost/arcgis/manager> adresinde yer almaktadır. Bu adresteki “localhost” kısmına, sunucunun IP adresi yazılarak, ArcGIS Server Manager’a dışardan erişilebilir. (Şekil 5.23)



Şekil 5. 23 ArcGIS Server Manager giriş sayfası

ArcGIS Server Manager arayüzü ile .sd dosyası ile istenilen veri sadece MapServer yeteneği ile servis edilmiştir. Servisle birlikte, servisin REST ve SOAP adresleri oluşmuştur (Şekil 5.24).

Mapping Configuration

URLs

REST URL: <http://localhost:6080/arcgis/rest/services/MyMapService/MapServer>

SOAP URL: <http://localhost:6080/arcgis/services/MyMapService/MapServer>

Operations Allowed

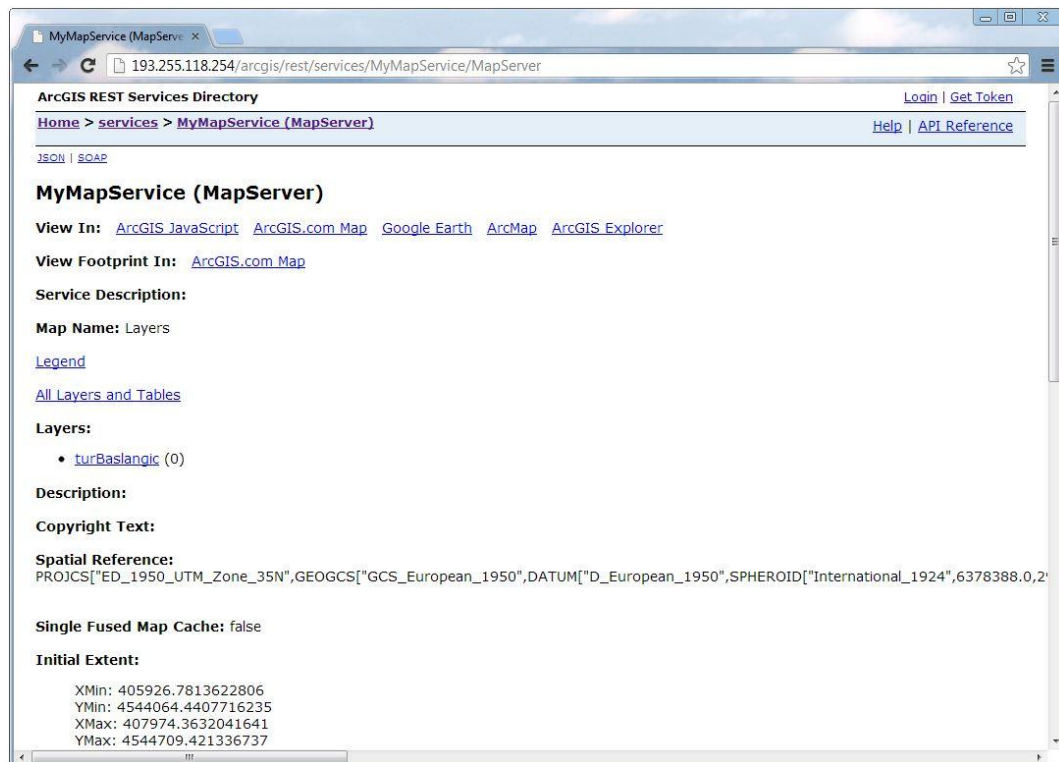
☒ Map

☒ Query

☒ Data

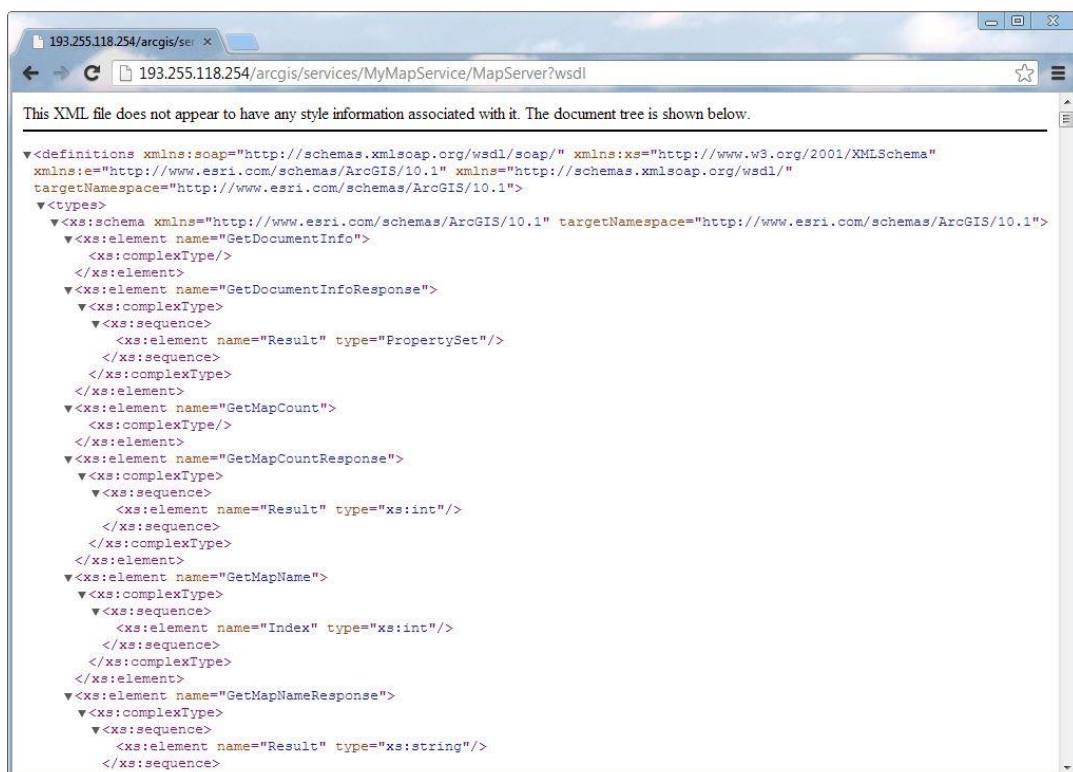
Şekil 5. 24 Oluşturulan servisin REST ve SOAP adresleri

Servisin REST yanıtı Şekil 5.25 'deki gibidir.



Şekil 5. 25 Oluşturulan servisin REST yanıtı

SOAP yanıtı ise Şekil 5.26'daki gibidir.

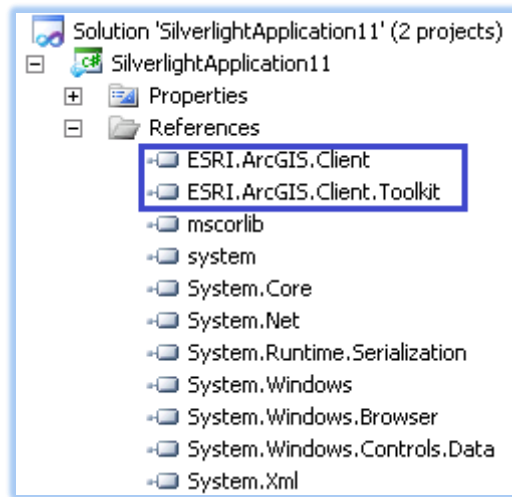


Şekil 5. 26 Oluşturulan servisin SOAP yanıtı

6.6.3 Web Uygulamasının Oluşturulması

Web uygulaması Visual Studio IDE (Integrated Development Enviroment) üzerinde, ArcGIS API for Silverlight programlama arayüzü kullanılarak oluşturulmuştur. Silverlight web uygulamasının oluşumu iki kısma ayrılabilir. Bunların birincisi tasarım bilgilerini tutan “MainPage.xaml” ve işlemlerin gerçekleştirildiği “MainPage.xaml.cs” dosyalarıdır. İkincisi ise uygulama kaynaklarının tutulduğu “App.xaml” ve “App.xmal.cs” dosyalarıdır. Uygulamanın görünüm kısmı XAML (Extensible Application Markup Language) dili kullanılarak tasarlanmıştır. Programlama kısmı ise C# dili kullanılarak geliştirilmiştir.

Öncelikle bilgisayara yüklenmiş olan ArcGIS API for Silverlight kütüphanelerine proje için gerekli referanslar atanmıştır (Şekil 5.27).



Şekil 5. 27 Referanslar

Tasarım dosyası için referans atamaları XAML dosyası üzerinde gerçekleştirilmiştir (Şekil 5.28)

```
<UserControl x:Class="SilverlightApplication11.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:esri="http://schemas.esri.com/arcgis/client/2009"
    xmlns:slData="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data"
    xmlns:i="http://schemas.microsoft.com/expression/2010/interactivity"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">
```

Şekil 5. 28 XAML referansları

Referanslar atandıktan sonra, bir grid oluşturulmuştur. Bu griddeki harita tepsisinde kullanılacak olan nokta vektör verisinin görünümü için kaynak oluşturulmuştur. Bu sayede haritadaki nokta verileri Şekil 5.29'daki gibi gözükcektir.

```
<Grid.Resources>
  <esri:SimpleMarkerSymbol x:Key="DefaultMarkerSymbol" Color="Coral" Size="10" Style="Circle" />
</Grid.Resources>
```



Şekil 5. 29 Grid kaynağı XAML (solda), görünüm (ağda)

Grid içerisinde bir harita tepsisi oluşturulmuştur. Bu tepsinin sınırları YTÜ Davutpaşa kampüsünü görüntüleyecek şekilde belirlenmiştir. Harita tepsisi içerisinde görüntülenecek olan katmanların belirlenmesi gerekmektedir (Şekil 5.30). Bu çalışmada altlık harita olarak İstanbul Büyükşehir Belediyesi (İBB) CBS sunucusunda bulunan 2010 yılına ait ortofoto görüntü servisi kullanılmıştır. Bu görüntü ED50 datumunda ve JPEG formatında, 11 seviye olarak cache edilmiş bir servistir. Bu görüntünün REST adresi “http://ibbmaps.ibb.gov.tr/maps/rest/services/ORTOFOTO_2010/MapServer” şeklindedir. İBB CBS sunucusundaki bu servisinin REST yanıtı ise Şekil 5.31'deki gibidir.

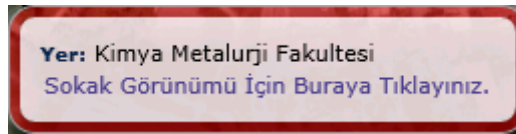
```
<esri:Map x:Name="MyMap" WrapAround="True" Extent="406460.23283851135,4543646.002966981,407440.9117279333,4544658.782366748">
  <esri:ArcGISDynamicMapServiceLayer ID="Ortofoto"
    Url="http://ibbmaps.ibb.gov.tr/maps/rest/services/ORTOFOTO_2010/MapServer" />
  <esri:GraphicsLayer ID="Noktalar"...>
</esri:Map>
```

Şekil 5. 30 Harita tepsisi



Şekil 5. 31 Servisin detayları (REST arayüzü)

Kullanıcıların noktalar üzerinde sorgu yapabilmesi için farenin imlecini noktalardan herhangi birinin üzerine gelmesiyle birlikte çalışan bir kod yazılmıştır. Kullanıcın fare imlecini nokta üzerine getirdiğinde, kullanıcıya o noktanın özniteliklerini ve o noktadan sokak görünümüne geçiş bağlantısını sunan bir pencere açılmaktadır (Şekil 5.32). Sorgulamanın gerçekleşebilmesi için yazılan kodlar ayrıntılı bir şekilde EK-B’de verilmiştir.



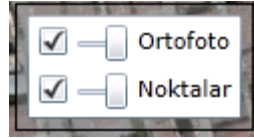
Şekil 5. 32 Sorgu sonucu

Kullanıcılar, katmanların görünümü açıp kapayabileceği, katmanların şeffaflık değerini değiştirebileceği bir katman listesi tasarlanmıştır (Şekil 5. 33).

```

<Border Background="#77919191" BorderThickness="1" CornerRadius="5"
    HorizontalAlignment="Left" VerticalAlignment="Top"
    Margin="20" Padding="5" BorderBrush="Black" >
    <ListBox x:Name="MyList" ItemsSource="{Binding ElementName=MyMap, Path=Layers}">
        <ListBox.ItemTemplate>
            <DataTemplate>
                <StackPanel Orientation="Horizontal">
                    <!--Katman Görünürlüğü Ayarlama-->
                    <CheckBox IsChecked="{Binding Visible, Mode=TwoWay}" />
                    <!--Transparan Oranı-->
                    <Slider Margin="-5,0,0,0" Minimum="0" Maximum="1" Width="30"
                        Value="{Binding Opacity, Mode=TwoWay}" Height="18" />
                    <!--Katman İsmi-->
                    <TextBlock Text="{Binding ID, Mode=OneWay}" Margin="5,0,0,0" >
                        <!-- Ekran Bilgisi-->
                        <ToolTipService.ToolTip>
                            <StackPanel MaxWidth="400">
                                <TextBlock FontWeight="Bold" Text="{Binding CopyrightText}" TextWrapping="Wrap" />
                                <TextBlock Text="{Binding Description}" TextWrapping="Wrap" />
                            </StackPanel>
                        </ToolTipService.ToolTip>
                    </TextBlock>
                </StackPanel>
            </DataTemplate>
        </ListBox.ItemTemplate>
    </ListBox>
</Border>

```



Şekil 5. 33 Katman listesi tasarımı XAML (üstte), katman listesi (altta)

Aynı şekilde kullacının haritayı kaydırabilmesi ve döndürebilmesi için API kütüphanesine hazır bulunan yönlendirme aracı eklenmiştir (Şekil 5. 34).

```

<esri:Navigation Margin="5" HorizontalAlignment="Left" VerticalAlignment="Bottom"
    Map="{Binding ElementName=MyMap}" >
</esri:Navigation>

```



Şekil 5. 34 Harita kontrolü XAML, harita kontrolü (altta)

Oluşturulan Web tabanlı coğrafi bilgi sisteminin son hali Şekil 5.35’de görüldüğü gibidir.



Şekil 5. 35 Oluşturulan Web tabanlı CBS’nin kullanıcı arayüzü

5.7 Sokak Görünümünün Vücut Hareketleriyle Kontrol Edilmesi

Vücut hareketlerinin algılanması için, bu çalışmada Microsoft tarafından satışa sunulan Kinect for Xbox 360 sensörü kullanılmıştır. Bu sensör aynı zamanda Windows işletim sistemi kurulu bilgisayarlarda da işlev görmektedir. Bu amaçla yayınlanan iki tip sürücü yazılımı vardır. Bunların birincisi Microsoft tarafından bir SDK ile birlikte kullanıma sunulmuştur. İkincisi ise Kinect sensöründeki temel teknolojinin üreticisi olan PrimeSense şirketinin de üyesi olduğu 3 boyutlu algılama organizasyonu OpenNI tarafından yayınlanmıştır. Bu çalışmada, Microsoft’un kullanıma sunduğu sürücü yazılımı ve Kinect for Windows SDK v1.5 kullanılmıştır.

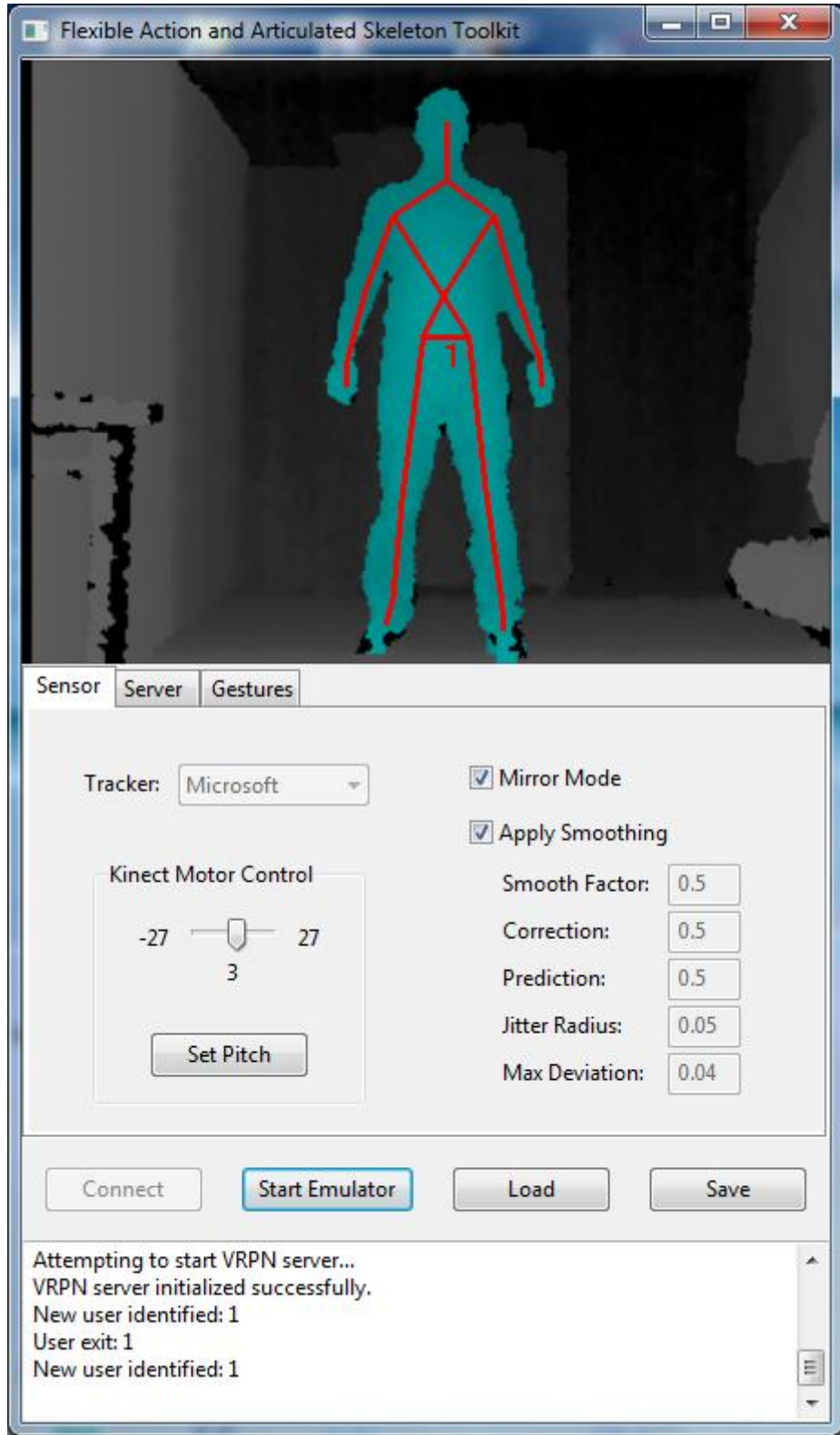
Çalışmada kullanılan diğer bir yazılım ise, Güney Kaliforniya Üniversitesi Yaratıcı Teknolojiler Enstitüsü tarafından üretilen Flexible Action and Articulated Toolkit (FAAST) özel yazılımdır (middleware). FAAST, Microsoft Kinect for Windows ya da

OpenNI sürücü yazılımlarını kullanarak, oyunlarla ve sanal gerçeklik uygulamalarıyla vücut kontrolünü kolaylaştıran bir yazılımdır. Bu yazılım sayesinde, vücut hareketlerinin herhangi bir klavye yada fare komutunu tetiklemesi sağlanabilir [41].

FAAST, 4 iskelete kadar destek vermektedir. Her iskelet için 24 eklem noktası algılamaktadır (Çizelge 5.9) (Şekil 5.36).

Çizelge 5. 9 Algı noktaları

Nokta	Vücut	Nokta	Vücut
0	Kafa	12	Sağ Dirsek
1	Boyun	13	Sağ Bilek
2	Gövde	14	Sağ El
3	Bel	15	Sağ Parmakucu
4	Sol Yaka	16	Sol Kalça
5	Sol Omuz	17	Sol Diz
6	Sol Dirsek	18	Sol Ayak Bileği
7	Sol Bilek	19	Sol Ayak
8	Sol El	20	Sağ Kalça
9	Sol Parmakucu	21	Sağ Diz
10	Sağ Yaka	22	Sağ Ayak Bileği
11	Sağ Omuz	23	Sağ Ayak



Şekil 5. 36 FAAST arayüzünde iskelet görünümü

Google Street View ve Google Maps API ile oluşturulan sokak görünümü uygulamalarında, kullanılabilecek klavye komutları Çizelge 5.10'daki gibidir.

Çizelge 5. 10 Sokak görünümü klavye komutları

Klavye Komutu	İşlevi
A – Sol Ok	Sola Döndürme
D – Sağ Ok	Sağa Döndürme
W – Page Up	Yukarı Döndürme
S – Page Down	Aşağı Döndürme
Yukarı Ok	Sonraki Görüntü
Aşağı Ok	Önceki Görüntü
+	Yakınlaştırma
-	Uzaklaştırma

Bu çalışmada sola, sağa, yukarı, aşağı döndürme ve görüntüler arasında geçiş komutları kullanılmıştır. Her yakınlaştırma değeri için ayrı görüntü parçaları oluşturulmadığından, yakınlaştırma ve uzaklaştırma komutları kullanılmamıştır (Çizelge 5.11).

FAAST arayüzü kullanılarak, Çizelge 5.11’de belirtilen vücut hareketlerine, karşılık gelen klavye komutları atanmıştır. Bu atanma işlemi, arayüz aracılığıyla XML dosyaları ile yapılır. Vücut hareketi “input” olarak, bu “input” sonucunda atanan komut “output” olarak tanımlanmaktadır. Şekil 5.37’de sağa dönme hareketinin tanımlandığı XML dosyası görülmektedir. Bu hareketin “input” olarak tanımlanmasında 5 adet tanımlayıcı (descriptor) mevcuttur. Birinci tanımlayıcı vücutun eğilmesi, dönmesi ya da zıplaması gibi ne tip bir hareket yapacağını belirler. Bu harekette vücut dönmektedir. İkinci tanımlayıcı vücutun hangi yöne doğru hareket edeceğini tanımlar. Bu harekette vücut sağa doğru dönmektedir. Üçüncü tanımlayıcı hareketin üst sınırını ya da alt sınırını belirler. Bu harekette “input” alt sınıra sahiptir. Dördüncü tanımlayıcı hareketin “input” olarak algılanması için dönme miktarını belirler. Beşinci tanımlayıcıyla birlikte dördüncü tanımlayıcının birimi belirlenmektedir. Bu “input” hareketi vücutun en az 25 derece sağa dönmesiyle gerçekleşir.

Çizelge 5. 11 Vücut hareketleri

Hareket	Komut
Vücutun Sağa Dönmesi	Sağ Ok
Vücutun Sola Dönmesi	Sol Ok
Vücutun Arkaya Yaslanması	Page Up
Vücutun Öne Eğilmesi	Page Down
Sağ Ayağın Öne Çıkması	Yukarı Ok
Sağ Ayağın Geriye Çıkması	Aşağı Ok

“input” sonucunda, gerçekleşecek olan “output” komutunu tanımlayan dört adet tanımlayıcı vardır. Birinci tanımlayıcı atanacak klavye komutuna bir kere basma ya da basılı tutma davranışını belirler. Bu “output” tanımında klavye komutu basılı kalacaktır. İkinci tanımlayıcı ise “output”un hangi klavye komutuna atanacağını belirler. Sağa dönmeye çağrışım yapması için bu harekete “output” olarak sağ ok tuşu atanmıştır. Üçüncü tanımlayıcı “output” komutunun ne zaman sonlandırılacağını belirler. Bu “output” tanımında, komut “input” hareketi devam ettiği sürece çalışacak şekilde belirlenmiştir. Dördüncü tanımlayıcı, üçüncü tanımlayıcının miktarını belirler. Sonuç olarak, sağa dönme “input” hareketinin gerçekleşmesi sonucunda, hareket tamamlanana kadar sağ ok tuşuna basma “output” komutu gerçekleşmektedir.

Sağa, sola, yukarı ve aşağı dönme hareketleri bu şekilde tanımlanmıştır. Bu hareketlerin tanımlayıcıları Çizelge 5.12’de görülmektedir. Bu hareketlere karşılık gelen komutlar ve tanımlayıcıları Çizelge 5.13’de görüldüğü gibidir.

```

<gesture name="sag" timeout="0">
  <input type="0">
    <descriptor>turn</descriptor>
    <descriptor>right</descriptor>
    <descriptor>at least</descriptor>
    <descriptor>25</descriptor>
    <descriptor>degrees</descriptor>
  </input>
  <output type="0">
    <descriptor>hold</descriptor>
    <descriptor>right_arrow</descriptor>
    <descriptor>until complete</descriptor>
    <descriptor>1</descriptor>
  </output>
</gesture>

```

Şekil 5. 37 Hareketlerin XML’de görünümü

Çizelge 5. 12 “input” tanımlayıcıları

	1. Tanımlayıcı	2. Tanımlayıcı	3. Tanımlayıcı	4. Tanımlayıcı	5. Tanımlayıcı
Sağa Dönme	Turn	Right	At Least	25	Degrees
Sola Dönme	Turn	Left	At Least	25	Degrees
Yukarı	Lean	Backwars	At Least	10	Degrees
Aşağı	Lean	Forward	At Least	10	Degrees

Çizelge 5. 13 “output” tanımlayıcıları

	1. Tanımlayıcı	2. Tanımlayıcı	3. Tanımlayıcı	4. Tanımlayıcı
Sağ Ok	Hold	Right Arrow	Until Complete	1
Sol Ok	Hold	Left Arrow	Until Complete	1
Page Up	Hold	Page Up	Until Complete	0
Page Down	Hold	Page Down	Until Complete	0

Sokak görünümünde sonraki ve önceki fotoğrafa geçiş için, ileri ve geri komutları oluşturulmuştur. Bu iki komutun tanımlayıcıları diğer komutlardan farklıdır (Şekil 5.38). Bu hareketler vücut parçalarının birbiriyle olan ilişkileriyle oluşturulmuştur. Bu

hareketlerin “input”unda altı adet tanımlayıcı vardır. Bu tanımlayıcıların birincisi vücutun ilk parçasını belirler. İleri hareketinde ilişkinin ilk parçası sağ ayak ile kurulmuştur. İkinci tanımlayıcı ise, belirlenen vücut parçasının önde, arkada, yukarıda olması gibi ne tip bir ilişkide olacağını belirler. Bu hareket, sağ ayağın ilişki kurulacak vücut parçasından önde olmasıyla gerçekleşir. Üçüncü tanımlayıcı ilk kısımda belirlenen vücut parçasının hangi vücut parçasıyla ilişkiye gireceğini tanımlanır. Bu senaryoda sağ ayağın gövdeden önde olması “input” olarak alınmaktadır. Diğer üç tanımlayıcı iki vücut parçası arasındaki ilişkinin miktar ve birimini belirlemektedir. Sonuç olarak ileri hareketi, sağ ayağın gövdenin 25 santimetre önüne geçmesiyle gerçekleşmektedir.

“output” komutları diğer hareketlerle aynı şekilde atanmıştır. İleri ve geri hareketlerinin “input” tanımlayıcıları Çizelge 5.14’te gösterilmiştir. Bu hareketlere karşılık gelen komutlar ve tanımlayıcıları Çizelge 5.15’te görüldüğü gibidir.

```
<gesture name="İleri" timeout="0">
  <input type="1">
    <descriptor>right foot</descriptor>
    <descriptor>in front of</descriptor>
    <descriptor>torso</descriptor>
    <descriptor>at least</descriptor>
    <descriptor>25</descriptor>
    <descriptor>centimeters</descriptor>
  </input>
  <output type="0">
    <descriptor>hold</descriptor>
    <descriptor>up_arrow</descriptor>
    <descriptor>until complete</descriptor>
    <descriptor>0</descriptor>
  </output>
</gesture>
```

Şekil 5. 38 İleri hareketi XML dosyası

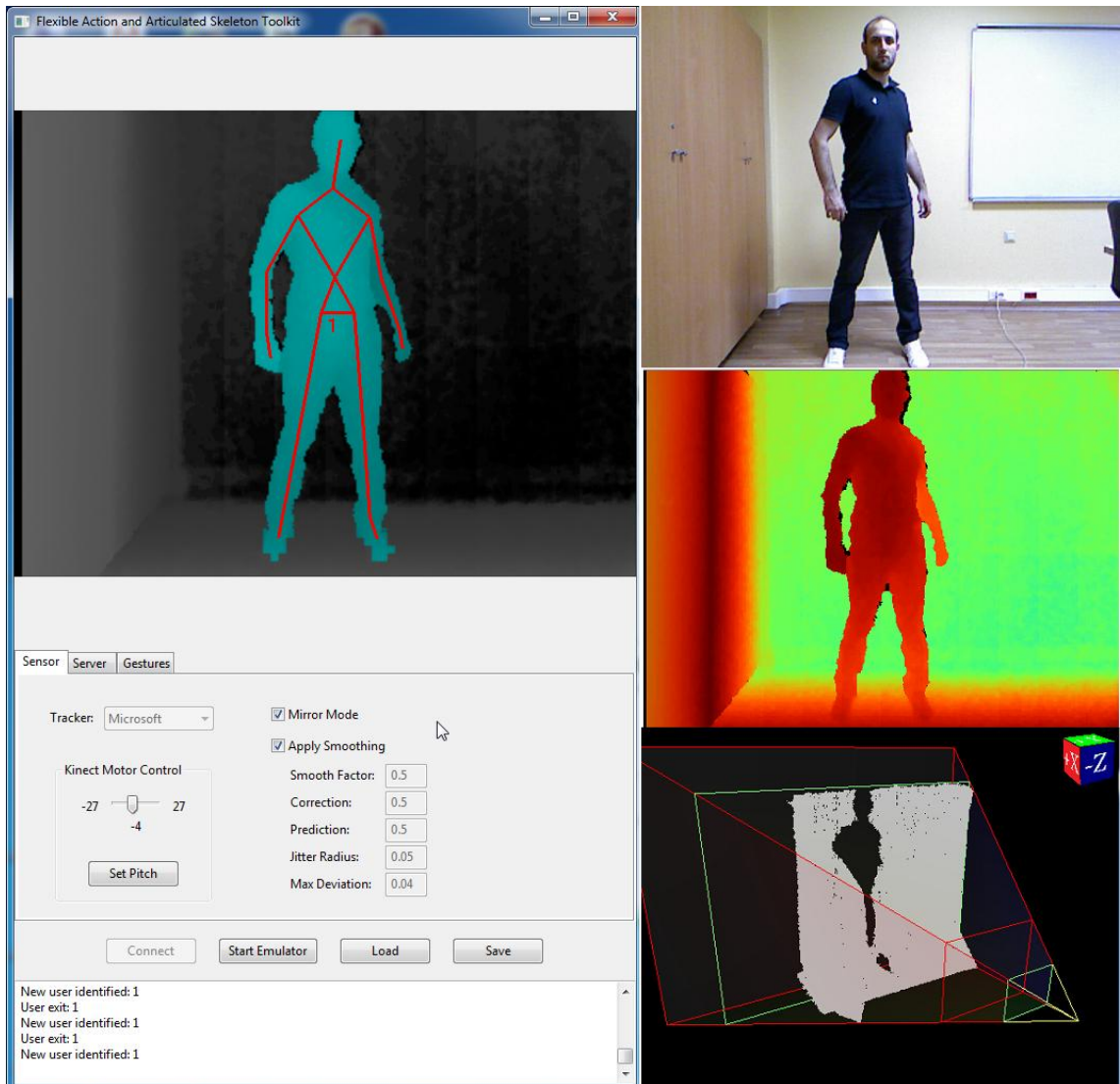
Çizelge 5. 14 İleri ve geri “input” tanımlayıcıları

	1. Tanım.	2. Tanım.	3. Tanım.	4. Tanım.	5. Tanım.	6. Tanım.
Sağ Ayak Önde	Right Foot	In Front Of	Torso	At Least	25	cm
Sağ Ayak Geride	Right Foot	Behing	Torso	At Least	25	cm

Çizelge 5. 15 İleri ve geri “output” tanımlayıcıları

	1. Tanımlayıcı	2. Tanımlayıcı	3. Tanımlayıcı	4. Tanımlayıcı
Yukarı Ok	Hold	Up Arrow	Until Complete	0
Aşağı Ok	Hold	Down Arrow	Until Complete	0

“Input” ve “Output” komutlarının atanma işlemi tamamlandıktan sonra, FFAST üzerindeki emulatör başlatılarak, atanan komutlarla sokak görünümü kontrolü sağlanmıştır. Örneğin, sokak görünümünü sağa doğru çevirmek için yapılması gereken vücut hareketi Şekil 5.39’da gösterildiği gibidir.

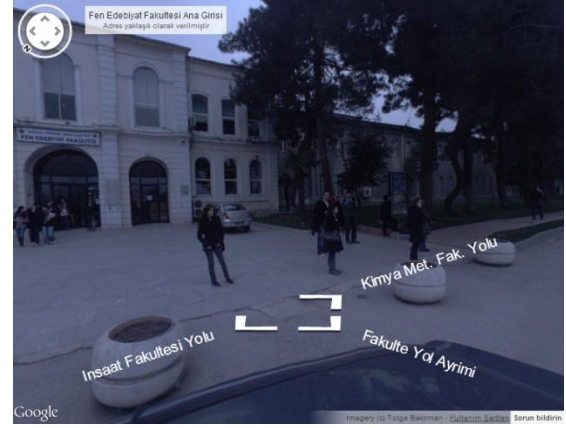


Şekil 5. 39 Sağa dönme hareketi (sol : FFAST, sağ – üst : RGB, sağ – orta : derinlik, sağ – alt : üç boyutlu görünüm)

Bu hareket sonucunda, sağa dönme hareketinin “input”u gerçekleşir. Hareketin “output”u olan sağ oka basma eylemi, hareket tamamlanana kadar devam eder. Böylece, sokak görünümünde heading değeri bir miktar sağa doğru kayar (Şekil 5.40).



(a)



(b)

Şekil 5. 40 Hareketten önce (a), hareketten sonra (b)

Diğer vücut hareketleri Şekil 5.41’ de görüldüğü gibidir. Bu hareketlerden sağa ve sola dönme sokak görünümünde heading değerini, öne ve arkaya eğilme düşey (pitch) değerini değerini değiştirir. Sağ ayağın gövdeye göre hareketleri ise bir sonraki ve bir önceki fotoğrafa geçişi sağlar. Böylece, kampüs için oluşturulan sokak görünümünün vücut hareketleriyle kontrolü gerçekleştirilmiştir.



(a)



(b)



(c)



(d)



(e)



(f)

Şekil 5. 41 (a) Sağa döndürme, (b) sola döndürme, (c) aşağı döndürme, (d) yukarı döndürme, (e) sonraki görüntü, (f) önceki görüntü

BÖLÜM 6

SONUÇ VE ÖNERİLER

Çalışma sonucunda, YTÜ Davutpaşa Kampüsü'ne ait sokak görünümü sistemi oluşturulmuştur. Oluşturulan sistem “www.tolgabakirman.com” adresinden kullanıma sunulmuştur. Bu adres kullanıcıyı sokak görünümünde kampüs ana girişine yönlendirmektedir.

Kampüs için oluşturulan CBS ile de fakülte girişlerine noktalar atılarak, bu noktalara sokak görünümüne geçiş adresleri öznitelik olarak girilmiştir. Böylece, birbiriyle bütünleşik bir sistem oluşturulmuştur. Oluşturulan sistem internet tabanlı olduğu için, birçok kullanıcının kolaylıkla sisteme erişmesi sonucunda Yıldız Teknik Üniversitesi, Davutpaşa kampüsün tanıtımına da katkı sağlanmıştır..

Bu çalışmada aynı zamanda sokak görünümü sisteminin performansı analiz edilmiştir. Sokak görünümü çalışmanın 5. Bölümünde belirtildiği gibi görüntülerin bütün ve parçalı olarak kullanıldığı iki farklı yöntemle oluşturulmuştur. Parçalı görüntülerin kullanıldığı yöntemle, bakılan bölgenin görüntüleri kullanıcı bilgisayarına aktarılır ve görüntülenir. Bunun yöntemin sonucunda sistemde daha hızlı gezinti yapma imkanı elde edilmiş olur. Diğer yöntem kullanıldığında görüntünün tamamı kullanıcı bilgisayarına aktarıldığından görüntülenme süresi artmaktadır. Yapılan analizler sonucunda bütün ve parçalı görüntülerin yüklenme süreleri arasında ortalama 0.9 saniye fark olduğu belirlenmiştir. Sonuç olarak, parçalı görüntülerin kullanıldığı sokak görünümü sistemi, daha hızlı çalışacağı için kullanıcıya daha akıcı ve kolay bir kullanım sunacaktır.

Çalışmanın son bölümünde, hareket sensörü kullanılarak sokak görünümü için vücut kontrol sistemi tanımlanmıştır. Bu sistem sayesinde, sokak görünümü vücut hareketleriyle kontrol edilebilmektedir. Böylece, bu çalışmada hem sokak görünümüne yeni bir kontrol mekanizması sunulmakta hem de bu iki farklı teknoloji bir arada kullanılmaktadır.

Web tabanlı CBS'nin yaygın kullanılmasıyla dünyanın her yerinden insanların konumsal ve sözel verilere internet aracılığı ile daha hızlı erişmesi mümkün olmaktadır. İnternet teknolojisinin gelişimine paralel olarak verilere erişim ve sunum teknikleri sürekli gelişmektedir.

Sokak görünümü sistemi hareket sensörüyle birlikte, iç mekanlarda kullanılarak, sanal müze ve planeteryum uygulamaları gerçekleştirilebilir. Ülkemizdeki turistik alanların tanıtılmasında böyle sistemlerin kullanılması büyük fayda sağlayacaktır.

Hareket sensörlerinde bulunan gerçek zamanlı iskelet ve hareket tanıma, insan ve bilgisayar etkileşimine yeni bir boyut kazandıracaktır. Bu teknoloji Augmented Reality ve Telepresence alanlarında da yeniliklere olanak sağlayacaktır.

KAYNAKLAR

- [1] Goodchild, M. F., (2004). "GIScience, Geography, Form, and Process", *Annals of the Association of American Geographers*, 4(94):709-714.
- [2] Xie, X. vd., (2011). "Application of China's National Forest Continuous Inventory Database", *Environmental Management*, 6(48):1095-1106.
- [3] Zhang, J., (2012). "A high-performance web-based information system for publishing large-scale species range maps in support of biodiversity studies", *Ecological Informatics*, 2(8):68-77.
- [4] Werts, J. D. vd., (2012). "An Integrated WebGIS Framework for Volunteered Geographic Information and Social Media in Soil and Water Conservation", *Environmental Management*, 4(49):816-832.
- [5] Glendhill, D. vd., (2003). "Panoramic Imaging – a review", *Computers & Graphics*, 3(27):435-445).
- [6] Chan, Y. vd., (1999). "A Panoramic-based Walkthrough System using Real Photos", *Seventh Pacific Conference on Computers Graphics and Applications*, 05-07 Ekim 1999, Seul.
- [7] Kelly, C. M. vd., (2013). "Using Google Street View to Audit the Built Environment: Inter-rater Reliability Results", *Annals of Behavioral Medicine*, 1(45):108-112.
- [8] Andrew, G. vd., (2011). "Using Google Street View to Audit Neighborhood Environments", *American Journal of Preventive Medicine*, 1(40):94-100.
- [9] Boulos M. N. vd., (2011). Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation", *International Journal of Health Geographics*, 10(45):1-14.
- [10] Sanna, A. vd., (2013). "A Kinect-based natural interface for quadrotor control", *Entertainment Computing*, 3(4):94-100.
- [11] Peng, Z., (1999). "An assessment framewrok for the development of Internet GIS", *Enviroment and Planning B: Planning and Design*, 1(26):117-132.

- [12] Linthicum, D., (2008), SOA in the Real World, <http://www.microsoft.com/en-us/download/details.aspx?id=16187>, 25 Mayıs 2013.
- [13] Marshall, J., (2000), Developing Internet-Based GIS Applications, http://proceedings.esri.com/library/userconf/proc00/professional/papers/PA_P405/p405.htm, 25 Mayıs 2013.
- [14] Şahin, K., (2006). Orman Yangınlarının İnternet Ortamında İnteraktif Olarak Sunumu, Yüksek Lisans Tezi, YTÜ, Fen Bilimleri Enstitüsü, İstanbul.
- [15] Şahin, K. ve Gümüşay, M. Ü., (2007). “İnternet Tabanlı Coğrafi Bilgi Sistemleri ve Orman Yangınlarında Kullanılması”, Harita Dergisi, (138):69-83.
- [16] Green, D. ve Bossomaier, T., (2002). Online GIS and Spatial Metadata, Taylor & Francis, London.
- [17] Fu, P. ve Sun, J., (2011). Web GIS : Principles and Applications, 1st Edition, ESRI Press, California.
- [18] Peng, Z. R. ve Tsou M. H., (2003). internet GIS : Distributed Geopgraphic Services for the Internet and Wireless Networks, John Wiley & Sons, New Jersey.
- [19] Fu, P. vd.,(2008). The best of REST : Using REST in ArcGIS Server 9.3, ESRI Press, California.
- [20] Cappelaere, P. vd. (2007). “RESTful OGC Services”, OGC Technical Committee Meeting, 17 Eylül 2007, Colorado.
- [21] ESRI, ArcGIS API for JavaScript, <http://developers.arcgis.com/en/javascript/jshelp/>, 25 Mayıs 2013.
- [22] ESRI, Creating web applications with the ArcGIS API for JavaScript, <http://resources.arcgis.com/en/help/main/10.1/index.html#//01540000040n000000>, 25 Mayıs 2013.
- [23] Dojo, Dojo Toolkit, <http://dojotoolkit.org/>, 25 Mayıs 2013.
- [24] ESRI, ArcGIS API for Flex, <http://resources.arcgis.com/en/help/flex-api/concepts/index.html#//017p00000004000000>, 25 Mayıs 2013 .
- [25] Peterson, M. P., (2012). Online Maps with APIs and WebServices, Springer, Nebraska.
- [26] Microsoft, About Silverlight, <http://www.microsoft.com/silverlight/what-is-silverlight/>, 25 Mayıs 2013.
- [27] ESRI, ArcGIS API for Silverlight Overview, http://resources.arcgis.com/en/help/silverlight-api/concepts/index.html#/ArcGIS_API_for_Silverlight_overview/016600000006000000/, 25 Mayıs 2013.
- [28] ESRI, ArcGIS API for Silverlight/WPF, http://resources.arcgis.com/content/enterprise/10.0/web_app_silverlight, 25 Mayıs 2013.

- [29] ESRI, ArcGIS API for Silverlight Library Reference, http://help.arcgis.com/en/webapi/silverlight/apiref/api_start.htm, 25 Mayıs 2013.
- [30] Google, A Brief History of Google Maps, <http://www.google.com/help/maps/helloworld/behind/history.html>, 25 Mayıs 2013.
- [31] Google, Where is Street View available?, <http://www.google.com/help/maps/streetview/learn/where-is-street-view.html>, 25 Mayıs 2013.
- [32] Point Grey Research, Ladybug®2 360° Video Camera, http://www.ptgrey.com/products/ladybug2/ladybug2_360_video_camera.asp, 25 Mayıs 2013.
- [33] Google, Creating Custom Panoramas, <https://developers.google.com/maps/documentation/javascript/streetview#CreatingPanoramas>, 25 Mayıs 2013.
- [34] Zhou, H. vd., (2009). "Human action recognition by feature-reduced Gaussian process classification", Pattern Recognition Letters, 12(30):1059-1066.
- [35] Schwarz, L. A. vd., (2012). "Human skeleton tracking from depth data using geodesic distances and optical flow", Image and Vision Computing, 3(30):217-226.
- [36] Khoshelham, K. ve Elberink, S. O., (2012). "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications", Sensors, 2(12):1437-1454.
- [37] Zeng, W., (2012). "Microsoft Kinect Sensor and Its Effect", Multimedia at Work, 2(19):4-10.
- [38] Shotton, J. vd., (2011). "Real-time human pose recognition in parts from single depth images", 2011 IEEE Conference on Computer Vision and Pattern Recognition, 20-25 Haziran 2011, Colorado Springs.
- [39] Raptis, M. vd., (2011). "Real-time classification of dance gestures from skeleton animation", 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 5-7 Ağustos 2011, Vancouver.
- [40] Pingdom, Pingdom Website Speed Test, <http://tools.pingdom.com/fpt/>, 25 Mayıs 2013.
- [41] Suma, E. vd., (2011). "FAAST: The Flexible Action and Articulated Skeleton Toolkit", IEEE Virtual Reality, 19-23 Mart 2011, Singapur.

İSİM DEĞİŞTİRME PROGRAMI C# KODLARI

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;


namespace WindowsFormsApplication5
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

private void button1_Click(object sender, EventArgs e)
{
    folderBrowserDialog1.ShowDialog();

    textBox1.Text = folderBrowserDialog1.SelectedPath.ToString();

    folderBrowserDialog2.SelectedPath = folderBrowserDialog1.SelectedPath;

    textBox3.Text = folderBrowserDialog2.SelectedPath.ToString();
}

private void button2_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    string[] dosyalar = Directory.GetFiles(textBox1.Text);

    foreach (string dosya in dosyalar)
    {
        FileInfo fileInfo = new FileInfo(dosya);

        string dosyaAdi = fileInfo.Name;

        ListViewItem item = new ListViewItem(dosyaAdi);

        listBox1.Items.Add(item);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    string fotoID = textBox2.Text;

    File.Move(textBox1.Text + "\\ " + fotoID + "_01.jpg", textBox3.Text + "\\ " + fotoID
+ "-0-0.jpg");
}

```

```

File.Move(textBox1.Text + "\\\" + fotoID + "_02.jpg", textBox3.Text+ "\\\" + fotoID
+ "-1-0.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_03.jpg", textBox3.Text+ "\\\" + fotoID
+ "-2-0.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_04.jpg", textBox3.Text+ "\\\" + fotoID
+ "-3-0.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_05.jpg", textBox3.Text+ "\\\" + fotoID
+ "-4-0.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_06.jpg", textBox3.Text+ "\\\" + fotoID
+ "-5-0.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_07.jpg", textBox3.Text+ "\\\" + fotoID
+ "-6-0.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_08.jpg", textBox3.Text+ "\\\" + fotoID
+ "-7-0.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_09.jpg", textBox3.Text+ "\\\" + fotoID
+ "-0-1.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_10.jpg", textBox3.Text+ "\\\" + fotoID
+ "-1-1.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_11.jpg", textBox3.Text+ "\\\" + fotoID
+ "-2-1.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_12.jpg", textBox3.Text+ "\\\" + fotoID
+ "-3-1.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_13.jpg", textBox3.Text+ "\\\" + fotoID
+ "-4-1.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_14.jpg", textBox3.Text+ "\\\" + fotoID
+ "-5-1.jpg");

File.Move(textBox1.Text + "\\\" + fotoID + "_15.jpg", textBox3.Text+ "\\\" + fotoID
+ "-6-1.jpg");

```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_16.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-7-1.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_17.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-0-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_18.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-1-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_19.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-2-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_20.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-3-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_21.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-4-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_22.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-5-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_23.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-6-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_24.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-7-2.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_25.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-0-3.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_26.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-1-3.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_27.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-2-3.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_28.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-3-3.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_29.jpg\", textBox3.Text+ \"\\\" + fotoID  
+ \"-4-3.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_30.jpg\", textBox3.Text + \"\\\" + fotoID  
+ \"-5-3.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_31.jpg\", textBox3.Text + \"\\\" + fotoID  
+ \"-6-3.jpg\");
```

```
File.Move(textBox1.Text + "\\\" + fotoID + \"_32.jpg\", textBox3.Text + \"\\\" + fotoID  
+ \"-7-3.jpg\");
```

```
listBox2.Items.Clear();
```

```
string[] dosyalar = Directory.GetFiles(textBox3.Text);
```

```
textBox2.Text = (Convert.ToInt32(fotoID) + 1).ToString();
```

```
foreach (string dosya in dosyalar)
```

```
{
```

```
FileInfo fileInfo = new FileInfo(dosya);
```

```
string dosyaAdi = fileInfo.Name;
```

```
long byteBoyut = fileInfo.Length;
```

```
DateTime olsTarihi = fileInfo.CreationTime;
```

```
ListViewItem item = new ListViewItem(dosyaAdi);
```

```
listBox2.Items.Add(item);
```

```
}
```

```
}
```

```
private void button4_Click(object sender, EventArgs e)
```

```
{
```

```
folderBrowserDialog2.ShowDialog();
```

```
textBox3.Text = folderBrowserDialog2.SelectedPath.ToString();
```

```
}
```

SORGULAMA KODLARI

B-1 MainPage.xaml Sorgulama Tasarım Kodları

```

<esri:GraphicsLayer ID="Noktalar">

    <esri:GraphicsLayer.MapTip>

        <Border                                esri:GraphicsLayer.MapTipHideDelay="00:00:01.5"
BorderBrush="DarkGray"                        CornerRadius="13"                        BorderThickness="1"
Margin="0,0,15,15">

            <Border.Effect>

                <DropShadowEffect                ShadowDepth="10"                        BlurRadius="14"
Direction="300" />

            </Border.Effect>

            <Border                                CornerRadius="10"                        Background="#DDFFEEEE"
BorderThickness="5" BorderBrush="#77FF0000">

                <StackPanel                Orientation="Vertical"                HorizontalAlignment="Center"
Margin="10">

                    <StackPanel Orientation="Horizontal">

                        <TextBlock                Text="Yer:                "                FontWeight="Bold"
Foreground="#FF0F274E" FontSize="10" VerticalAlignment="Center"/>

                        <TextBlock                Text="{Binding [Yer]}"                HorizontalAlignment="Left"
VerticalAlignment="Center" />

```

```

        </StackPanel>

        <HyperlinkButton x:Name="myHyperLink" NavigateUri="{Binding
[sokakGorun]}" TargetName="_blank" Content="Sokak Görünümü İçin Buraya
Tıklayınız." Foreground="DarkSlateBlue"/>

    </StackPanel>

</Border>

</Border>

</esri:GraphicsLayer.MapTip>

</esri:GraphicsLayer>

```

B-2 MainPage.xaml.cs Kodları

```

private void CitiesGraphicsLayerLoad()
{
    ESRI.ArcGIS.Client.Tasks.Query query = new ESRI.ArcGIS.Client.Tasks.Query()
    {
        Geometry = new ESRI.ArcGIS.Client.Geometry.Envelope(-180, 0, 0, 90) {
        SpatialReference = new ESRI.ArcGIS.Client.Geometry.SpatialReference(4326) },

        ReturnGeometry = true,

        OutSpatialReference = MyMap.SpatialReference
    };

    query.OutFields.Add("*");

    query.Where = "Id > -1";

    QueryTask queryTask = new
    QueryTask("http://127.0.0.1:6080/arcgis/rest/services/MyMapService/MapServer/0");

    queryTask.ExecuteCompleted +=
    CitiesGraphicsLayerQueryTask_ExecuteCompleted;

    queryTask.ExecuteAsync(query);
}

```

```

    }

    void CitiesGraphicsLayerQueryTask_ExecuteCompleted(object sender,
    QueryEventArgs queryArgs)
    {
        if (queryArgs.FeatureSet == null)
            return;

        FeatureSet resultFeatureSet = queryArgs.FeatureSet;

        ESRI.ArcGIS.Client.GraphicsLayer graphicsLayer =
        MyMap.Layers["Noktalar"] as ESRI.ArcGIS.Client.GraphicsLayer;

        if (resultFeatureSet != null && resultFeatureSet.Features.Count > 0)
        {
            foreach (ESRI.ArcGIS.Client.Graphic graphicFeature in
            resultFeatureSet.Features)
            {
                graphicFeature.Symbol = LayoutRoot.Resources["DefaultMarkerSymbol"] as
                ESRI.ArcGIS.Client.Symbols.Symbol;

                graphicsLayer.Graphics.Add(graphicFeature);
            }
        }
    }
}

```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Tolga BAKIRMAN
Doğum Tarihi ve Yeri : 06.07.1989 / Şişli
Yabancı Dili : İngilizce
E-posta : bakirman@yildiz.edu.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Harita Müh.	Yıldız Teknik Üniversitesi	-
Lisans	Harita Müh.	Yıldız Teknik Üniversitesi	2010
Lise	-	Ahmet Rasim Lisesi	2006

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2010	Yıldız Teknik Üniversitesi	Araştırma Görevlisi

YAYINLARI

Makale

1. Bakirman, T., ve Gümüřay, M. Ü. (2011). “Yıldız Teknik Üniversitesi Davutpařa Kampüsü ÖSYM Salon Bilgi Sistemi”, HKM Dergisi, 2(104 Özel S.1):146-149.

Bildiri

1. Visual Tour Based on Panoromic Images for Indoor Places in Campus, T. Bakirman. Proc. of The 22th Congress of the International Society for Photogrammetry and Remote Sensing, 25 Ağustos – 1 Eylül, 2012, Melbourne, Avustralya

Proje

1. İstanbul Çevre Düzeni Planı’nın Kıyı Alanlarına İliřkin Yaklaşımının Mekana Yansıtılmasına Yönelik Arařtırma, Deęerlendirme ve Model Geliřtirme İři, BİMTAř ve İBB, 2011 – (Devam ediyor)

ÖDÜLLERİ

1. En İyi Genç Sunum Ödülü, 13. THBTK , 18-22 Nisan, 2011, Ankara, Türkiye