

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HARİTA MÜHENDİSLİĞİ ANABİLİM DALI

**ÜÇ BOYUTLU KENT MODELLERİ İÇİN SİLÜET ANALİZ MODÜLÜ TASARIMI VE
GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Sebahat TEMUÇİN KILIÇER

**HAZİRAN - 2017
TRABZON**



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce

Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : / /

Tezin Savunma Tarihi : / /

Tez Danışmanı :

Trabzon

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

başlıklı bu çalışma, Enstitü Yönetim Kurulunun / / gün ve sayılı
kararıyla oluşturulan jüri tarafından yapılan sınavda
YÜKSEK LİSANS TEZİ
olarak kabul edilmiştir.

Jüri Üyeleri

Başkan :

Üye :

Üye :

Prof. Dr. Sadettin KORKMAZ

Enstitü Müdürü

ÖNSÖZ

“Üç Boyutlu Kent Modelleri için Silüet Analiz Modülü Tasarımı ve Geliştirilmesi” başlıklı bu çalışma, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Harita Mühendisliği Anabilim Dalında, yüksek lisans tezi olarak gerçekleştirilmiştir.

Eğitim hayatımın her evresinde bana destek olan tüm öğretmenlerime, Karadeniz Teknik Üniversitesin’deki lisansüstü öğrenimim boyunca benden, bilgilerimi, tecrübelerini, zamanını esirgemeyen ve beni bu günlere getiren saygıdeğer tez danışmanım Prof. Dr. Çetin CÖMERT’e teşekkürlerimi borç bilirim. Ayrıca bilimsel bilgilerimi paylaşarak çoğaltmayı ve büyütmeyi temel edinmiş, Artvin Çoruh Üniversitesi, Harita Mühendisliği Bölümü öğretim üyelerinden Doç. Dr. Halil AKINCI’ya ve Yrd. Doç. Dr. Ayşe YAVUZ ÖZALP’a yardımlarından dolayı çok teşekkür ederim.

Tüm hayatım boyunca maddi ve manevi desteklerini esirgemeyen aileme, zor zamanlarda her zaman yanımda olan sevgi ve saygı dolu eşime, her an büyük içtenlikle yardımına koşan dostlarıma sonsuz teşekkürlerimi ve şükranlarımı sunarım.

Bu çalışmayı kızım Beren’e atfetmek istiyorum.

Sebahat TEMUÇİN KILIÇER

Trabzon 2017

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Üç Boyutlu Kent Modelleri için Silüet Analiz Modülü Tasarımı ve Geliştirilmesi” başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Çetin CÖMERT’in sorumluluğunda tamamladığımı, verileri ve örnekleri kendim topladığımı, analizleri ilgili programlarda yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 23/06/2017

Sebahat TEMUÇİN KILIÇER

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
SUMMARY	VIII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	XII
KISALTMALAR DİZİNİ	XIII
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Problemin Tanımı.....	3
1.3. Çalışmanın Amacı	4
1.4. Metodoloji	5
1.5. 3B Konumsal Veri Modelleri	6
1.5.1. Voksel Temsili	8
1.5.2. Sınır Temsili.....	8
1.5.3. 3B Temel Blokların Kombinasyonu.....	10
1.5.4. Birleşik Modeller.....	11
1.5.5. Multipatch Geometri Tipi.....	12
1.5.5.1. Üçgen Şerit.....	14
1.5.5.2. Üçgen Yelpaze	15
1.5.5.3. Üçgen.....	16
1.5.5.4. Ring	16
1.6. 3B Kent Modelleme	17
1.6.1. CityGML	18
1.6.2. Diğer Standartlar	19
1.6.3. CityEngine.....	20
1.7. CBS Yazılımları Tarafından Sunulan 3B Analiz Fonksiyonları ...	23
1.7.1. Görüş Hatları Çizme Analizi.....	24
1.7.2. Görüş Analizi	25

	<u>Sayfa No</u>
1.7.3. Görüş Hattı Analizi	25
1.7.4. Gözlemci Noktaları Analizi.....	27
1.7.5. Gölge Analizi	28
1.7.6. Görüş Alanı Analizi.....	28
1.7.7. Görünürlük Analizi.....	29
1.7.8. Skyline Analizi	30
1.7.9. Skyline Bariyer Analizi	31
1.7.10. Skyline Grafik Analizi.....	32
2. YAPILAN ÇALIŞMALAR	36
2.1. 3B Bina Modelli Üretimi.....	32
2.2. Silüet Analiz Modülünün Geliştirilmesi.....	42
2.3. Silüet Analiz Modülü İçin Araç Çubuğunun Tasarımı	46
2.4. Uygulama	52
3. BULGULAR VE TARTIŞMA	63
4. SONUÇLAR VE ÖNERİLER	69
5. KAYNAKLAR.....	70
6. EKLER	75
ÖZGEÇMİŞ	

Yüksek Lisans

ÖZET

ÜÇ BOYUTLU KENT MODELLERİ İÇİN SİLÜET ANALİZ MODÜLÜ TASARIMI VE
GELİŞTİRİLMESİ

Sebahat TEMUÇİN KILIÇER

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Harita Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Çetin CÖMERT
2017, 74 Sayfa, 21 Ek Sayfa

Üç boyutlu kent modelleri; arazi yüzeyleri, yerleşim bölgeleri, binalar, bitki örtüsü, altyapı ve peyzaj öğeleri gibi kentsel alanlara ait nesnelere temsil edildiği dijital temsilleridir. Üç boyutlu kent modelleri, bilgisayar oyunları ve eğitim amaçlı kullanıldığı gibi kentsel planlama, afet yönetimi, tesis yönetimi, lojistik, güvenlik, telekomünikasyon, konumsal servisler, gayrimenkul değerlendirmeleri gibi birçok farklı uygulamada sunum, üretim, analiz ve yönetim görevlerinde de kullanılmaktadır. 3B konumsal analizlerden biri olan silüet analizi, kentsel alan düzenlemesi ve peyzaj planlaması için önemli analizlerdendir. Silüet analizleri, özellikle kentsel alanlarda kent simgesi haline gelmiş önemli binaların silüetinin korunması için kullanılmaktadır. En sık kullanılan Coğrafi Bilgi Sistemleri yazılımlarından olan ArcGIS, sahip olduğu 3D Analyst modülünün sunduğu fonksiyonlar ile görüş hatları çizme, görüş, görüş hattı, gözlemci noktaları, gölge, görüş alanı, görünürlük, skyline, skyline bariyer ve skyline grafik gibi analizleri gerçekleştirebilmektedir. Ancak, 3D Analyst modülünde binaların silüet görünümünü oluşturacak bir fonksiyon bulunmamaktadır. Bu çalışmada, ArcGIS yazılımı için Python programlama dili kullanılarak bir silüet analiz modülü geliştirilmiş ve bu modülün kullanılacağı bir araç çubuğu tasarlanmıştır. Silüet analiz modülü, gözlem yapılan noktadan belirli bir görüş alanında kalan binaların silüet görüntüsünü üretebilmekte ve aynı alanda yapılması düşünülen yeni binaların silüete etkilerini tespit edebilmektedir. Ayrıca silüet analiz modülünün kullanılması ile yeni yapılan binanın kent silüetini bozup bozmadığı belirlenebilmekte ve yeni binaların maksimum yüksekliği ve kat adedi hesaplanabilmektedir. Böylelikle kentsel planlama sürecinde ve imar planlarının uygulanmasında karşılaşılan kentsel silüetin bozulması problemlerinin çözümüne önemli katkı sağlanmıştır.

Anahtar Kelimeler: CBS, 3B Kent Modelleri, Kentsel Planlama, Silüet Analizi, ArcGIS, Python

Master Thesis

SUMMARY

DESIGN AND DEVELOPMENT OF SILHOUETTE ANALYSIS MODULE FOR THREE
DIMENSIONAL CITY MODELS

Sebahat TEMUÇİN KILIÇER

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Geomatics Engineering Graduate Program
Supervisor: Prof. Dr. Çetin CÖMERT
2017, 74 Pages, 21 Appendix Pages

3D city models are digital representations of objects such as terrain surfaces, sites, buildings, vegetation, infrastructure and landscape elements belonging to urban areas. 3D city models are used in presentation, exploration, analysis, and management tasks for a large number of different applications including urban planning, disaster management, facility management, logistics, security, telecommunication, location-based services, real estate portals as well as being used for computer games and educational purposes. Silhouette analysis, one of 3D spatial analyses, is important for urban area designing and landscape planning. Silhouette analysis is used to protect the silhouette of important buildings that have become landmark especially in urban areas. ArcGIS is one of the most popular software in Geographical Information Systems (GIS) market. It has “3D Analyst Tools” providing functions enabling many analyses including construct sight lines, intervisibility, line of sight, observer points, sun shadow volume, viewshed, visibility, skyline, skyline barrier and skyline graph. However, the 3D Analyst Tools does not have a function to create a silhouette view of buildings. In this study, a silhouette analysis module was developed using the Python programming language for the ArcGIS software and a toolbar was designed to use this module. Silhouette analysis module can produce a silhouette view of buildings in a certain field of view from the observation point and can track changes on silhouette when new buildings planned to be built in the same area. In addition, the use of the new silhouette analysis module can determine whether a newly built building has distorted the urban silhouette and calculates the maximum height and floor number of new buildings. Thus, important contribution was provided in solving the distortion problems of urban silhouette encountered in the urban planning process and in the implementation of the development plans.

Key Words: GIS, 3D City Models, Urban Planning, Silhouette Analysis, ArcGIS, Python

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Voksel temsil ile gösterim	8
Şekil 2. BREP temsil ile gösterim	10
Şekil 3. Boolean operasyonları	11
Şekil 4. CSG örneği	11
Şekil 5. Multipatch objesi için varlık ilişki veri modeli	13
Şekil 6. Üçgen şerit oluşturulması	13
Şekil 7. Üçgen şerit objesi	15
Şekil 8. Üçgen yelpaze objesi	15
Şekil 9. Üçgen objesi	16
Şekil 10. Ring objeleri	16
Şekil 11. CityGML tarafından tanımlanan LoD gösterimi	19
Şekil 12. VDI 3805, IFC ve CityGML arasındaki ilişki	20
Şekil 13. CityEngine için içe ve dışa aktarılabilen formatlar	21
Şekil 14. Görüş hatları için durulan nokta ve bakılan hedefler	24
Şekil 15. Görüş hatları çizme analizinin gösterimi	24
Şekil 16. Görüş analizinin gösterimi	25
Şekil 17. Görüş hattı analizi gösterimi	26
Şekil 18. Görüş hattı analizi uygulaması	26
Şekil 19. ArcMap görüş hattı analizinin uygulanması	27
Şekil 20. Gözlemci noktaları analizi gösterimi	27
Şekil 21. Gölge analizi gösterimi	28
Şekil 22. Görüş alanı analizi gösterimi	29
Şekil 23. ArcGIS Azimuth açısı temsili	30
Şekil 24. Skyline analizi gösterimi	31
Şekil 25. Skyline bariyer analizi gösterimi	32
Şekil 26. ArcGIS kutupsal açı temsili	32
Şekil 27. Skyline grafik analizi tablo gösterimi	33
Şekil 28. Skyline grafik analizi grafik gösterimi	34
Şekil 29. Yeni binanın silüet görüntüsüne etkisi	35
Şekil 30. Terrain penceresi	37

Şekil 31. File Geodatabase penceresi	38
Şekil 32. Project All kullanılarak üretilmiş 3B bina modelleri	39
Şekil 33. Translate to Average kullanılarak üretilmiş 3B bina modelleri ...	40
Şekil 34. Kurallar kullanılarak 3B bina modelleri üretimi	40
Şekil 35. Kurallar kullanılarak oluşturulan 3B bina modelleri.....	41
Şekil 36. Esri FileGDB penceresi.....	42
Şekil 37. IDE Options penceresi.....	44
Şekil 38. Project Settings penceresi.....	46
Şekil 39. Add-In Contents penceresi	48
Şekil 40. Kod klasörü görünümü.....	48
Şekil 41. Esri ArcGIS Add-In Installation Utility penceresi	49
Şekil 42. Add-In Manager penceresi	49
Şekil 43. Silhouette araç çubuğu	50
Şekil 44. Silhouette araç çubuğunda butonların aktif ve pasif görünümü...	50
Şekil 45. Protected Building açılır kutusunda multipatch binaların gösterimi	51
Şekil 46. Durulan noktanın değiştirilmesine dair uyarı mesajı	51
Şekil 47. Silüet görüntüsünün üretilen tarih ve saatinin gösterimi.....	52
Şekil 48. Silüet görüntüsü penceresi	52
Şekil 49. 3B bina modelleri	53
Şekil 50. Yüzey modelinin seçtirilmesi.....	54
Şekil 51. Durulan noktanın belirlenmesi	54
Şekil 52. Silüet hattının belirlenmesi.....	55
Şekil 53. Binalar katmanının seçtirilmesi.....	56
Şekil 54. Silüet üretiminin başlatılması.....	57
Şekil 55. Binalar katmanına ait silüet görüntüsü.....	57
Şekil 56. Yeni bina katmanının seçtirilmesi.....	58
Şekil 57. Yeni silüet üretiminin başlatılması.....	59
Şekil 58. Yeni ve mevcut binalar katmanına ait silüet görüntüsü	59
Şekil 59. BuildingID 10 için korunan bina katmanının sorgulanması	60
Şekil 60. BuildingID 10 için korunan bina katmanının seçtirilmesi	60
Şekil 61. Silüet analizinin başlatılması.....	61

Şekil 62. Silüet analiz raporu.....	61
Şekil 63. BuildingID 7 için korunan bina katmanının sorgulanması	62
Şekil 64. BuildingID 7 için silüet analiz raporu	62
Şekil 65. Skyline analizi üzerinde açı ve mesafelerin gösterimi	63
Şekil 66. Skyline grafik analizi üzerinde açı ve mesafelerin gösterimi.....	63
Şekil 67. Skyline grafikleri üzerinden silüet durumunun yorumlanması	65
Şekil 68. Görsel skyline analizi gösterimi; 1999 yılı, doğu-batı yönü (a), 2008 yılı ve sonrası, doğu-batı yönü (b), 1999 yılı, kuzey-güney yönü (c), 2008 yılı, kuzey-güney yönü (d)	66
Şekil 69. Zamansal skyline analizi gösterimi; Boğaziçi Köprüsü (a), Fatih Sultan Mehmet Köprüsü (b), Harem (c), Çamlıca Tepesi (d)	67
Şekil 70. Kuala Lumpur'un silüet analizleri gösterimi; 1 nolu gözlem noktası, mevcut durum (a), 1 nolu gözlem noktası, yeni durum (b), 2 nolu gözlem noktası, mevcut durum (c), 2 nolu gözlem noktası, yeni durum (d), 3 nolu gözlem noktası, mevcut durum (e), 3 nolu gözlem noktası, yeni durum (f).....	68

TABLÖLAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1. 3B veri modellerinin karşılaştırılması	12



KISALTMALAR DİZİNİ

BREP	: Boundary Representations
BS	: Bina Servisleri
CAD	: Computer Aided Design
CBS	: Coğrafi Bilgi Sistemleri
CGA	: Computer Generated Architecture
CityGML	: City Geographic Markup Language
COLLADA	: Collaborative Design Activity
CSG	: Constructive Solid Geometry
DXF	: Data eXchange Format, Veri Değişim Formatı
E	: Kenarların Sayısı
ERDAS	: Earth Resources Data Analysis System
ESRI	: Environmental Systems Research Institute Incorporated Compony
F	: Yüzey Sayısı
GML	: Geographic Markup Language, Coğrafi Veri İşaretleme Dili
GUI	: Graphical User Interface, Grafiksel Kullanıcı Arayüzü
IDE	: Integrated Development Enviroment
IFC	: Industrial Foundation Class
LoD	: Level of Detail
OGC	: Open Geospatial Consortium
OO	: Object Oriented
OO 3D	: Object Oriented Three Dimention
OpenGL	: Open Graphics Library
TEN	: Tetrahedral Network
SOMAS	: Solid Object Managment System
SSM	: Simplified Spatial Model
SVG	: Scalable Vector Graphics, Ölçeklenebilir Vektör Grafikleri
SYM	: Sayısal Yükseklik Modeli
UDM	: Urban Data Model
V	: Köşe Sayısı

VBA	: Visual Basic for Applications
XML	: Extensible Markup Language, Geniřletilebilir İřaretleme Dili
z	: ykseklik
2B	: İki Boyutlu
3B	:  Boyutlu
3DFDS	: Three Dimention Formal Data Structure



1. GENEL BİLGİLER

1.1. Giriş

Genel olarak, “konumsal veri tabanı yönetimi için tasarlanmış yazılım ve donanım elemanlarının bütünü (Masry ve Lee, 1988)” şeklinde tanımlanan Coğrafi Bilgi Sistemleri (CBS), dünya üzerindeki karmaşık sosyal, ekonomik ve çevresel sorunların çözümüne yönelik konuma dayalı karar verme süreçlerinde yaygın olarak kullanılmaktadır. CBS, konumsal verilerin gösterimi ve sunumunun ötesinde, sahip olduğu iki boyutlu (2B) ve üç boyutlu (3B) veri modelleme, sorgulama ve analiz fonksiyonları sayesinde turizm, çevre, enerji, tarım, orman, ulaşım, afet ve acil durum yönetimi, araç takibi, kentsel planlama, şehircilik ve arazi kullanım uygulamaları gibi farklı birçok alanda karar vericilerin doğru kararlar almasına katkı sağlamaktadır.

Konumsal verilerin 2B düzlemde temsil edilmesiyle üretilen 2B haritalar, CBS'nin kullanıldığı birçok çalışmada altlık olarak kullanılmaktadır. Ancak, gürültü tahmin modelleri (Kluijver ve Stoter, 2003), hava kirliliği modelleri, taşkın modelleri, jeolojik modeller (Van Wees vd., 2002) ve emlak piyasası (Stoter ve Zlatanova, 2003; Stoter ve Ploeger, 2003) ile ilgili uygulamalarda 2B konumsal verilerin ve bu veriler üzerinde gerçekleştirilen analizlerin yetersiz kaldığı görülmektedir (Stoter ve Zlatanova, 2003). Donanım ve bilgisayar grafiklerindeki gelişmeye paralel olarak uygulamalarda 3B veriye olan talebin artması, konumsal veri modellemesinde üçüncü boyuta odaklanılmasını sağlamıştır (Zlatanova vd, 1998). Böylelikle yeryüzünde bulunan enerji nakil hatları, aydınlatma direkleri, yollar, ağaçlar ve binalar gibi objeler bilgisayar ortamında 3B temsil edilerek “3B Kent Modelleri” üretilmiştir. 3B kent modellerinin kullanıldığı uygulama alanlarının genişlemesi ile birlikte 3B konumsal modelleme ve analizler, günümüzde CBS alanında önemli araştırma konuları haline gelmiştir.

Literatürde 3B kent modelleri kullanılarak kentsel alanların planlanması ve yönetimine katkıda bulunan birçok çalışma bulunmaktadır. Sadek vd. (2002) tarafından yapılan çalışmada, şehirlerin 3B görselleştirilmesi için şehir plancılarının kullanabileceği bir 3B kent modeli oluşturulmuştur. Czerwinski vd. (2007) tarafından yapılan çalışmada, Almanya'nın nüfusa göre birinci şehri olan North Rhine-Westphalia'da 3B modellenen binaların gürültü emisyon hesapları ile gürültü analizleri yapılmıştır. Ban vd. (2011)

tarafından yapılan ViSuCity projesi ile sürdürülebilir kent ve çevre planlamasını desteklemek için web tabanlı interaktif bir görüntüleyici tasarlanmıştır. Lamberti vd. (2011), 3B kent modelini kullanarak cadde ve sokakların aydınlatılması üzerine çalışmışlardır (Mao, 2011). Schulte ve Coors (2008), 3B tasarlanmış binaların taşkın simülasyonunu yaparak afet yönetimi alanında uygulama geliştirmişlerdir. Lee ve Zlatanova (2008) ise yangın gibi insanların acil tahliye edilmesi gereken durumlarda uygun tahliye yollarını belirlemek için binaların 3B modellenmesi ve 3B topolojik analizler üzerinde çalışmışlardır.

Gürültü analizi (noise analysis), hava kirliliği analizi (air pollution analysis), ağ analizi (network analysis), gölge analizi (shadow analysis), görünürlük analizi (visibility analysis) ve silüet analizi (silhouette analysis) gibi konumsal analizler, 3B CBS uygulamalarında ihtiyaç duyulan analizlere örnek olarak gösterilebilirler. Görünürlük analizleri, 1970'li yıllardan beri CBS uygulamalarında kullanılan analizlerdir (Yang vd., 2007). Görüş hattı, gölge ve skyline gibi görünürlük analizleri günümüz CBS'nde kullanılan görünürlük analizlerine örnek olarak gösterilebilirler. Görünürlük analizleri; özellikle kentsel planlama, çevre düzenlenmesi, peyzaj planlamaları ile baz istasyonları, rüzgâr türbinleri ve güneş enerjisi sistemlerinin kurulacağı alanların belirlenmesi gibi uygulamalarda yaygın olarak kullanılmaktadır.

Görünürlük analizleri içinde yer alan silüet analizi, kentsel planlamada kent mimarisini korumak ve imar planlarının üretilmesinde doğru kararlar verebilmek açısından büyük öneme sahiptir. Güney vd. (2012) çalışmalarında iki ayrı silüet tanımına yer vermişlerdir. Birincisinde silüet, "yerin ve gökyüzün buluştuğu çizgi, ufuk; bunun resim veya başka bir sanattaki temsili" olarak tanımlanmaktadır. İkincisinde ise silüet, "bir veya bir dizi binanın veya gökyüzünde görülen diğer nesnelerin taslağı" olarak tanımlanmaktadır. Kentsel silüet veya şehir silüeti ise kentsel alanlardaki binaların bir noktadan olan görüntüsü olarak tanımlanabilir. Literatürde, kentsel alanlardaki binaların silüetlerini üreten çeşitli çalışmalar bulunmaktadır. Örneğin, Nasar ve Terzano (2010) tarafından, dijital fotoğraflar kullanarak doğal ve kentsel alanların silüetlerini kıyaslayan bir çalışma yapılmıştır. Yusoff vd. (2014) tarafından, 3B kent modeli kullanarak Kuala Lumpur şehrinin silüetinin korunmasına yönelik çalışmalar yapılmıştır. Czyńska (2015) tarafından yapılan çalışmada, kentsel alanlarda yüksek yapılı binaların tarihi yapılar üzerindeki etkisi incelenmiştir. Tafahomi vd. (2016) tarafından yapılan çalışmada ise İran'ın ikinci büyük şehri olan Mashhad şehrindeki binalara ait silüet çalışmaları

yapılmıştır. Tavernor ve Grassner (2010) ise Londra'daki yüksek yapılı kulelerin, Waterloo köprüsü ve St. Paul katedrali üzerindeki görsel etkisini incelemişlerdir.

Tarihi ve kültürel varlıklara sahip şehirlerde kent silüetinin korunması yasalarla güvence altına alınmaya çalışılmaktadır. İstanbul Boğaziçi Alanının kültürel ve tarihi değerlerini ve doğal güzelliklerini kamu yararı gözetilerek korumak ve geliştirmek ve bu alandaki nüfus yoğunluğunu artıracak yapılanmayı sınırlamak için uygulanacak imar mevzuatını belirlemek ve düzenlemek amacıyla yürürlüğe giren 18/11/1983 tarih ve 2960 sayılı Boğaziçi Kanunu buna örnek olarak gösterilebilir. Boğaziçi Kanunu'nun 3. maddesindeki genel esaslar incelendiğinde, dolaylı olarak İstanbul'daki silüetin önemine ve korunmasına vurgu yapıldığı anlaşılmaktadır. Buradan hareketle, İstanbul şehrinin silüetinin son on yıl içindeki değişimini inceledikleri çalışmalarında Akdag vd. (2010), kentin kimliğini korumak için sistematik bir yaklaşıma ihtiyaç duyulduğunu dile getirmişlerdir. Akdag vd. (2010) çalışmalarında örnek olarak, İstanbul'da Zincirlikuyu-Malak yolunda yer alan yüksek katlı binaların İstanbul Boğazındaki silüetin değişimine olan etkilerini incelemişlerdir. Güney vd. (2012), kentsel alanların planlamasında kent silüeti öneminin vurgulandığı çalışmalarında, İstanbul'un Levent semti civarında pilot bir bölge belirleyerek çalışma alanının 3B kent modelini oluşturmuşlardır. Boğaziçi Köprüsü, Fatih Sultan Mehmet Köprüsü, Harem ve Çamlıca Tepesi gibi şehrin önemli noktalarından çalışma alanının görünürlüğü incelenmiş ve çeşitli 3B analizler gerçekleştirmişlerdir. Bu tezin ilgi alanı, kentsel alanların planlanmasında ve kent silüetinin korunmasında, silüet analizlerinin önemine vurgu yapmak ve bu amaçla bir silüet analiz yazılımı geliştirmektir.

1.2. Problemin Tanımı

Şehirlerin doğal ve tarihi güzellikleri, kültürü, hatta geçmiş medeniyetlere ait kültürel etkileri, bir şehrin cadde, sokak ve binalarına tarihi doku olarak yansımaktadır. Şehirlerin köprüleri, kaleleri, camileri, kuleleri, duvar ve surları gibi tarihi yapıları zamanla o şehirle özdeşleşen objeler haline gelmektedir. Benzer şekilde, söz konusu yapıların silüetleri de buldukları şehirleri temsil eden simgelere dönüşmektedir. Kent silüetleri sadece simge olarak kullanılmamaktadır. Kent silüetleri, aynı zamanda, çevre düzeni planları, mekansal strateji planları, kentsel dönüşüm planları ve imar planları gibi kentsel alanlara yönelik planlamalarda ihtiyaç duyulan önemli bir veridir.

Kültürel değerler ile doğal güzelliklerin bulunduğu ve doğal yapının korunması gerektiği alanlarda mevcut silüetin korunması, şehir simgesi haline gelmiş yapıların silüetinin korunması veya turistik öneme sahip stratejik noktaların turistik önemini kaybetmemesi için silüetinin korunması gibi durumlar, silüet analizlerine olan gereksinimi ortaya koymaktadır. Bununla birlikte, hızlı nüfus artışı ve kırsal kesimden kentlere olan yoğun göç nedeniyle konut talebinin artması, buna bağlı olarak hızlı yapılaşma faaliyetlerinde rant kaygısı ve yerel yönetimlerin yetersizlikleri, kentlerin doğal ve tarihi çevresine zarar vermekte ve kentlerin silüetleri bozulmaktadır.

Kuala Lumpur, Singapur, Chicago, Miami, Londra ve İstanbul gibi mega kentlerde hızla yükselen gökdelenlerin kentin silüeti üzerindeki olumsuz etkileri birçok kez gündeme getirilmiştir. İstanbul'da Dolmabahçe Sarayı ve arkasında hızla yükselen binalara ait silüet ile Haliç Metro Geçiş Köprüsü'nün Unkapı Köprüsü ve Tarihi Yarımada'daki silüete etkileri çeşitli çalışmalarda incelenmiştir. Zincirlikuyu ve Maslak gibi hızla gelişmekte olan ve bir yatırım fırsatı haline gelen bölgelerdeki binaların, İstanbul'un kentsel silüetine etkileri geçtiğimiz yıllarda sıkça gündeme getirilmiş hatta bazı binalar için Tarihi Yarımada'daki silüet dokusunu bozduğu için mahkeme tarafından yıkım kararı bile verilmiştir.

İmar planlarına uygun olarak inşa edilen ruhsatlı binalar için silüeti bozduğu gerekçesiyle yıkım kararı verilmesi bir yandan önemli ekonomik kayıplara neden olmakta diğer yandan da "bu binalar inşa edilmeden önce daha proje aşamasında iken silüeti bozup bozmadığı belirlenemez miydi?" sorusunu akla getirmektedir. Bu soru, bu tez çalışmasının çıkış noktası olmuştur. Tezde ele alınan problem ise bunun nasıl başarılabileceğidir.

1.3. Çalışmanın Amacı

Günümüzde yaygın olarak kullanılan ticari veya açık kaynak kodlu CBS yazılımları, sundukları 3B analiz fonksiyonları ile çeşitli analizlerin yapılmasına olanak sağlamalarına rağmen silüet analizi konusunda yetersiz kalmaktadırlar. Silüet analizinin yetersizliği, "hangi binaların silüeti bozduğu, hangilerinin bozmadığı, silüetin bozulmaması için nereye ne kadar yükseklikte bina yapılması gerektiği, farklı noktalardan bakıldığında silüetin nasıl değiştiği" gibi soruları yanıtsız bırakmaktadır. Kentsel alanların planlanması ve imar planı uygulamaları sürecinde, kentsel silüet analizlerine olanak sağlayan yazılımların bulunması, karar vericilerin benzer sorulara cevap bularak doğru karar vermelerine katkı sağlayacaktır.

ESRI (Environmental Systems Research Institute Inc) firması tarafından geliştirilen ArcGIS yazılımı, CBS'nin veri girişi, işleme, sorgulama, analiz ve sunum gibi temel modülleri ile ilgili birçok fonksiyon sağlamakta ve tüm Dünya'da yaygın olarak kullanılmaktadır. ArcGIS yazılımı, verileri 3B olarak temsil edebilmekte ve sunduğu "3D Analyst" modülü ile çeşitli konumsal analizlere olanak sağlamaktadır. 3D Analyst modülünün görünürlük analizleri alt modülü; görüş hatları çizme analizi, görüş analizi, görüş hattı analizi, gözlemci noktaları analizi, gölge analizi, görüş alanı analizi, görünürlük analizi ile skyline, skyline bariyer ve skyline grafik analizleri gibi görünürlük analizlerinde kullanılan çeşitli fonksiyonlara sahiptir. Bu analiz fonksiyonlarının birçoğu, genel olarak, arazi yüzeyi, 3B bina modelleri veya diğer 3B objeleri dikkate alarak bir noktadan görülebilen veya görülemeyen alanların belirlenmesine olanak sağlamaktadır. Söz konusu analiz fonksiyonları kullanılarak, 3B binaların silüet görüntüsü üretilmemekte veya yeni yapılacak binaların tarihi veya kültürel öneme sahip binaların silüetini bozmaması için sahip olması gereken yükseklik veya kat adedi gibi özellikleri hesaplanamamaktadır. Önceki sürümlerinde Visual Basic for Applications (VBA) kullanılarak genişletilme imkânı sunan ArcGIS programı, günümüzde nesne yönelimli (object-oriented) bir programlama dili olan Python programlama dili kullanılarak genişletilme imkânı sunmaktadır. Bu tez çalışmasında, Python programlama dili kullanılarak ArcGIS CBS yazılımında silüet analizlerinin gerçekleştirilmesine olanak sağlayan yeni bir modülün tasarlanması ve gerçekleştirilmesi amaçlanmıştır.

1.4. Metodoloji

Bu çalışmada, 3B kent modelini kullanarak silüet analizi gerçekleştirme yeteneğine sahip olan bir modülün geliştirilmesi için sırasıyla aşağıdaki işlem adımları izlenmiştir.

- Örnek bir çalışma alanına ait bina katmanının ve sayısal yükseklik modelinin üretilmesi,
- CityEngine programında objelerin modellenmesi ve 3B bina modellerinin üretilmesi,
- 3B bina modellerinin ArcGIS ortamına aktarılması,
- Python programlama dili ve PyScripter programı ile silüet analizi modülünün geliştirilmesi,

- ArcGIS Python Add-In Wizard programı ile silüet analiz modülü için araç çubuğunun (toolbar) tasarlanması ve ArcGIS programına entegre edilmesi,
- Geliştirilen silüet analiz modülü kullanılarak çalışma alanındaki binaların silüet görüntüsünün üretilmesi ve analiz raporunun elde edilmesi.

1.5. 3B Konumsal Veri Modelleri

Konumsal modelleme, gerçek dünya nesnelere dijital temsillerinin oluşturulmasını amaçlamaktadır. Bununla birlikte, gerçek dünyadaki nesnelere karmaşık ve belirsiz bir şekilde tanımlanırken, bilgisayarlar bu nesnelere soyutlanmış ve tam olarak tanımlanmış dijital benzerleri (ya da muadilleriyle) çalışabilirler. Konumsal modelleme süreci, gerçekliği, gerçek dünya nesnelere ve süreçlerine benzeyen üst düzey somut varlıklar dizisi olarak yorumlayarak başlar ve nihai olarak bilgisayarda depolanan yakın olan soyut düşük düzeyli temsiller oluşturmayı amaçlayan bir dizi ilerici soyutlama yaklaşımları kullanır (Ogori, 2016).

Birçok CBS kullanıcısı, 3B CBS'ni, standart 2B veya 2.5B konumsal verinin 3B bir görünümü olarak görmektedir. Veriler üç boyutta görselleştirilebilmesine rağmen konumsal analiz, veri işleme ve diğer yaygın CBS işlemlerinin çoğu 2B bir ortamda gerçekleştirilir. Birçok uygulama için basitçe 3B verilerin görüntülenmesi yeterli olabilir, ancak bazı disiplinler, modellenecek özelliklerin karmaşık doğası veya 3B konumsal analizlere izin vermek için nesnelere gerçek 3B temsiline ihtiyaç duyarlar (Ford, 2004).

İki boyutta her konum bir (x, y) koordinat çifti ile temsil edilmektedir. Fakat 2.5 boyutta ise belirli bir (x, y) konumu için yalnızca bir yükseklik değeri söz konusudur ve bu yükseklik değeri öznitelik tablolarında yer almaktadır (Cambray ve Yeh, 1994). Böylelikle geometrik olarak 2B bir veri, 3B bir veri gibi algılanabilmektedir. Örneğin, bir 3B yüzey olarak algılanan ve fonksiyonel yüzeyler olarak da tanımlanan Sayısal Yükseklik Modeli (SYM), öznitelik tablolarında yükseklik bilgisine sahip poligonların birleşimi ile oluşan yüzeyler olarak düşünülebilir, ancak yüzeyin bir kalınlığı yoktur ve dolayısıyla 2,5 boyutludur (Ford, 2004). Bir (x, y) konumu için farklı yükseklik değerlerine sahip olan karmaşık yapıları objeler ancak üçüncü boyut ile temsil edilebilmektedir. Bu nedenle belirli bir (x, y) konum için birden fazla z-koordinatı gerektiriyorsa bu tür temsiller tam bir 3B temsil (True 3D) olarak tanımlanmaktadır. Farklı disiplinlerdeki birçok uygulama için

gerçek dünya nesnelерinin gerçek 3B olarak modellenmesi gerekmektedir (Cambray, 1993).

Literatürde 3B objelerin temsilleri ile ilgili farklı sınıflandırmalarla karşılaşmak mümkündür. Cambray'a (1993) göre Bilgisayar Destekli Tasarım (Computer Aided Design, CAD) modelleri; hacim (volume) ve sınır (boundry) temsili olarak iki grupta toplanmaktadır. Stoter ve Zlatanova (2003)'ya göre 3B objelerin temsili; Constructive Solid Geometry (CSG), voksel (voxel), tetrahedron (TEN) ve sınır temsili olarak dört gruba ayrılmaktadır. Rahman ve Pilouk (2007)'a göre ise konumsal obje temsilleri; hacim (volume) ve yüzey (surface) temsili olarak iki gruba ayrılmaktadır. Tuan (2013) tarafından yapılan bir diğer çalışmada ise 3B CBS uygulamaları için üç boyutlu veri modelleri genel olarak dört grupta sınıflandırılmaktadır.

Bunlar;

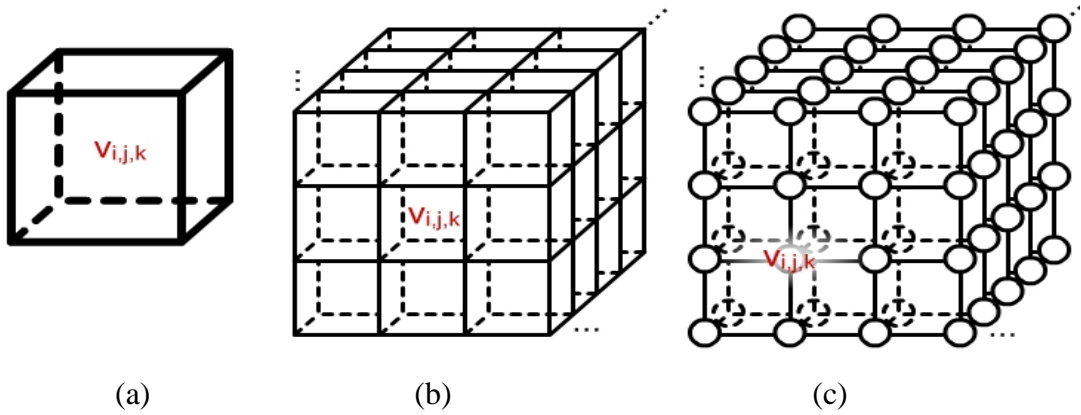
- 3B objelerin voksel elementleri ile temsili
- 3B objelerin sınır temsili
- 3B objelerin 3B temel blokların kombinasyonu ile temsili
- 3B objelerin yukarıdaki temsillerin birleşimi ile temsili

CBS'de ve 3B uygulamalarda yaygın olarak kullanılan ArcGIS yazılımı, veri yapılarının ilişkisel veri tabanı yönetim sistemi (İVTYS) içerisinde yönetilmesini ArcSDE teknolojisi ile sağlamaktadır. Konumsal veri yapılarının depolandığı ArcSDE tabloları, standart nokta, çizgi ve poligon gibi çeşitli geometri türlerini desteklemektedir. Standart ilişkisel veri tabanında, poligon veri yapıları kullanılarak 3B temel geometrik elemanlar (primitives) oluşturmak için ya çoklu poligon (multi-polygon) yaklaşımı ya da ilişkisel tablolar (linking tables) gibi 2B sayısız veri yapısı gerekmektedir. Oracle 9i Spatial bu çoklu geometriler olan multiple geometries veri yapılarını desteklemekte iken ESRI çoklu geometrileri desteklememektedir. Bu nedenle ESRI veri tabanında tek bir geometri alanına sahip bir 3B geometrik veriyi standart bir tablo yapısında depolanmasına olanak sunan multipatch veri yapısını önermektedir. Sonuç olarak ESRI, 3B veri yapılarının modellenmesi için nokta, çizgi ve poligon veri yapılarına ek olarak, Open Graphics Library (OpenGL) üç boyutlu üçgen geometri elemanları üzerine kurulu bir "multipatch" geometri veri yapısını sunmaktadır (Ford ve James, 2005).

1.5.1 Voksel Temsili

Voksel, bir pikselin üç boyutlu karşılığı olarak tanımlanmaktadır. Piksel, bir noktayı 2B olarak tanımlarken voksel bir noktayı üç boyutlu uzayda tanımlayan grafik bilgisini ifade etmektedir (URL-1, 2017). Şekil 1’de tek bir vokselin görünümü (a), bir voksel grubu (b) ve bir voksel grubunun grid görünümü (c) gösterilmektedir (URL-2, 2017).

2B piksel kavramından yola çıkılarak üretilen ve küp şeklindeki birim elemanlardan oluşan voksel veri yapısı, 3B nesnelerin oluşturulması ve bazı temel geometrik hesaplamaların gerçekleştirilmesinde oldukça faydalı bir modeldir (Karaş, 2007). Raster veri yapısındaki üç boyutlu veri modelleri birinci dereceden ayrıntı düzeyi olan blok modelleme yapılabilir ve büyük ölçekli veri boyutları ile çalışabilmektedir. Fakat tablo bilgileri içermemekte ve topolojik veri yapılarının oluşturulmasında yetersiz kalmaktadır. 3B diziler (3D arrays) modeli ve sekizli ağaç (octree) modeli üç boyutlu objelerin voksel temsiline örnek olarak verilebilmektedir (Tuan, 2013).



Şekil 1. Voksel temsil ile gösterim (URL-2, 2017)

1.5.2 Sınır Temsili

Boundary Representations (BREP), Türkçe’de Sınır Temsili olarak tanımlanmaktadır (Kaya, 1999). BREP, önceden tanımlanmış nokta, kenar, yüz ve hacim gibi 3B temel geometrik elemanların birleşimiyle oluşan nesnelere olarak tanımlanmaktadır. BREP temsili, köşeler, kenarlar ve yüzeyler kullanılarak katı cismin yüzey sınırları ile temsilinden oluşturulmaktadır (Ekberg, 2007).

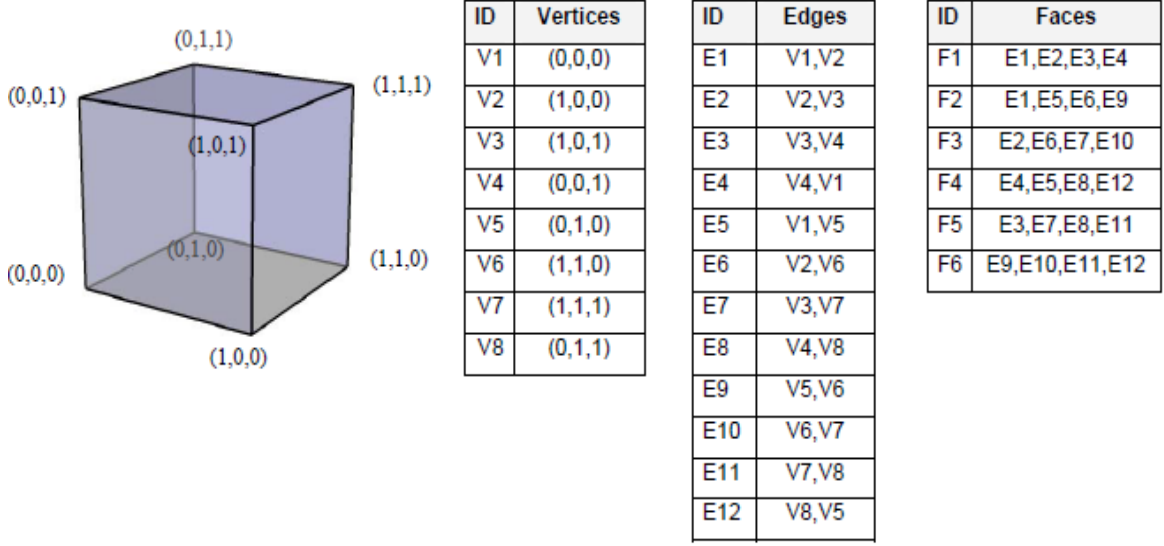
Her biri yüz adını alan düzlemsel çokgenlerle sınırlanan katı cisimlere çokyüzlüler denmektedir. Üç veya daha fazla yüzün kesiştiği noktaya ise köşe, yüzlerin birbiriyle kesiştiği doğrular ise kenar (ayrıt) olarak adlandırılmaktadır. Bütün yüzleri özdeş düzgün çokgenlerden oluşan çokyüzlüler düzgün çokyüzlü olarak bilinmektedir. Bu düzgün çokyüzlüler, her köşesinde aynı sayıda özdeş düzgün çokgenin kesiştiği 3B Platonik cisimler olarak tanımlanmaktadır.

Bir çokyüzlü için köşe sayısı ile yüzey sayısının toplamından kenar sayısı çıkarıldığında daima sabit bir değer olan 2 sayısı elde edilmektedir. Matematikçi Leonhard Euler (1707-1783) tarafından bulunan (1) nolu formüle Euler çokyüzeyle formülü denmektedir. Bu formüle göre her düzgün çok yüzlü için bu denklem eşitliği sağlanmaktadır (URL-3, 2017). Burada V köşe sayısı, E kenarların sayısı ve F yüzey sayısı olarak tanımlanmaktadır. BREP temsili ile oluşturulmuş bir düzgün çokyüzlü olan Şekil 2'deki küp için Euler' in formülü eşitliğini $8-12+6=2$ olarak sağlanmaktadır (Ekberg, 2007).

$$\text{Köşe Sayısı (V) - Kenar Sayısı (E) + Yüzey Sayısı (F) = 2} \quad (1)$$

BREP, gerçek dünya objelerinin temsili için en uygun modelleme tekniği olarak bilinmekte ve çoğu modelleme motorlarının BREP kullanması bir avantaj olarak görülmektedir (Stoter ve Zlatanova, 2003). Ayrıca Foley vd.'e (1990) göre BREP, Euler formülüyle topolojik tutarlılığı korumak için yararlı bir yaklaşım olarak görülmektedir. Fakat BREP için veri girişi yani köşeler, kenarlar ve yüzler için farklı tabloların eklemek gerekmesi ve karmaşık yapıları objeler için yüzler ile kenarların kesişmesi gerekmesi BREP temsili için dezavantaj olarak bilinmektedir (Ekberg, 2007).

Son yıllarda 3B objelerin sınır temsili ile modellenmesine olanak sağlayan modeller önerilmiştir. Three Dimention Formal Data Structure (3DFDS), Tetrahedral Network (TEN), Simplified Spatial Model (SSM), Object Oriented (OO) Model, Solid Object Managment System (SOMAS), Urban Data Model (UDM), OO 3D ve City Geographic Markup Language (CityGML) modelleri 3B objelerin sınır temsiline örnek olarak verilebilmektedir (Tuan, 2013). Ayrıca ArcGIS yazılımının da kullanmış olduğu multipatch veri yapısı da sınır temsili örneklerinden biridir (Edvardsson, 2013).



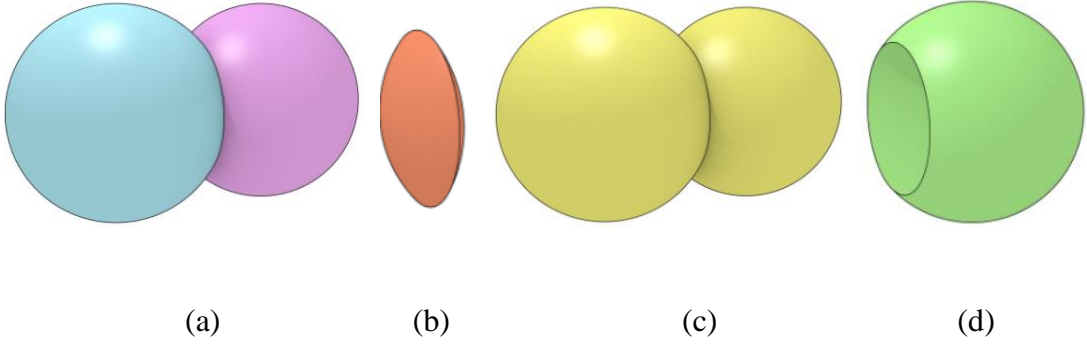
Şekil 2. BREP temsil ile gösterim (Ekberg, 2007).

1.5.3 3B Temel Blokların Kombinasyonu

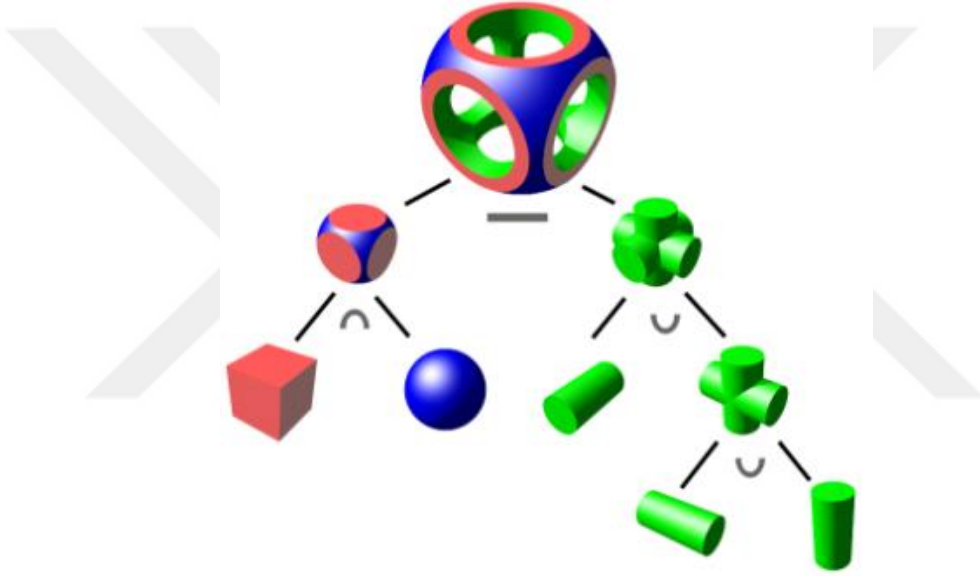
CSG, Türkçe’de “Yapısal Katı Geometri” olarak tanımlanmaktadır (Kaya, 1999). Yazılımlarda hazır bulunan küre, küp ve silindir gibi basit yapıları 3B temel geometrik elemanlar kullanarak, 3B yeni bir katı cisim oluşturan modelleme tekniklerindedir (Ekberg, 2007).

Primitives olarak isimlendirilen 3B temel geometrik elemanlar, birleşim (union), kesişim (intersect) ve fark (difference) gibi Boolean operasyonları kullanarak oluşturulmaktadır. Şekil 3’de küre şeklindeki iki objenin konumsal ilişkisi (a) ve bu objelerin kesişimi (b), birleşimi (c) farkı (d) gibi başlıca Boolean operasyonları gösterilmektedir (Ohori, 2016).

Şekil 4’de bir adet küp, bir adet küre ve 3 adet silindir üzerinde Boolean operasyonları kullanılarak temsil edilmiş bir CSG objesi gösterilmektedir (Ohori, 2016). Boolean operasyonları kullanılarak CSG temsili; CAD sistemlerinde yaygın olarak kullanılması, temsiline daha anlaşılır ve basit olması nedeniyle bir avantaj olarak görülmektedir. Fakat Jarroush ve Even-Tzur’a (1987) göre 3B topolojik ilişkiyi tutamaması ve Stoter’a (2004) göre karmaşık yapıları gerçek dünya objelerinin modellenmesi için detaylı ağaç (tree) yapısı gerekmesi CSG temsil için dezavantajlar olarak belirtilmektedir (Ekberg, 2007).



Şekil 3. Boolean operasyonları (Ohori, 2016)



Şekil 4. CSG örneği (URL-4, 2017)

1.5.4 Birleşik Modeller

CBS'nde dünya yüzeyi ve üzerine kurulu yapıların 3B modellenmesi üzerine yapılan araştırmalar ve önerilen modelleme teknikleri zamanla gelişim göstermekte ve ihtiyacı karşılamaya yönelik farklı modelleme teknikleri ileri sürülmektedir. Belli başlı 3B modelleme temsillerinin yanında bu modelleme temsillerinin bir arada kullanılmasıyla geliştirilmiş modelleme temsilleri de bulunmaktadır. V3D modeli ve CSG-BREP modelleri 3B objelerin birleşik model temsillerine örnek olarak verilebilmektedir (Tuan, 2013). Tuan (2013), 2000 yılında önerilen V3D modelinin, raster görüntüleri ve vektör verilerin

bütünleştirilmesi fikrine dayandırıldığını belirtmektedir. 2009 yılında 3B nesnelerin CSG ve BREP metotlarının kombinasyonu temsil edilmesi önerilmiştir.

Günümüzde kullanılan 3B CBS veri modellerinin karşılaştırması Tablo 1’de verilmiştir.

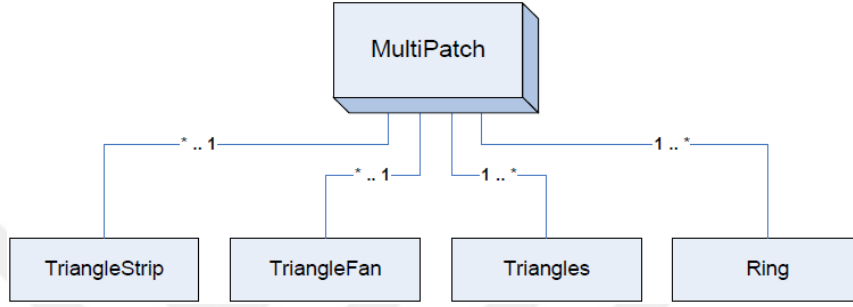
Tablo 1. 3B veri modellerinin karşılaştırılması (Tuan, 2013)

Temsil	Model	Veri Yapısı	Veri Boyutu	Topoloji	Obje İçi Temsil
BREP	3DFDS	Vektör	Büyük	Evet	Hayır
	TEN	Vektör	Büyük	Hayır	Evet
	SSM	Vektör	Büyük	Hayır	Evet
	OO	Vektör	Küçük	Evet	Hayır
	SOMAS	Vektör	Büyük	Hayır	Hayır
	UDM	Vektör	Küçük	Hayır	Hayır
	OO 3D	Vektör	Büyük	Hayır	Evet
	CityGML	Vektör	Büyük	Hayır	Hayır
Voksel	3D Array	Raster	Çok Büyük	Hayır	Evet
	Octree	Raster	Çok Büyük	Hayır	Evet
CSG	CSG	Vektör	Küçük	Hayır	Evet
Birleşik	V3D	Vektör Raster	Büyük	Hayır	Evet
Modeller	BREP-CSG	Vektör	Büyük	Hayır	Hayır

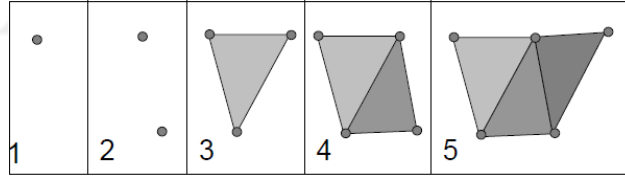
1.5.5 Multipatch Geometri Tipi

Multipatch, ESRI tarafından 1997 yılında geliştirilen ve 3B objelerin temsilinde kullanılan nokta, çizgi ve poligon gibi bir geometri tipidir (ESRI, 2008). Multipatch obje aslında sınırları ile temsil edilen bir katı cisim (solid feature) objesi olarak tanımlanmaktadır (Ford, 2004). Bir multipatch obje, her biri bir yüzey (surface) olarak tanımlanan yüzey parçalarının (patch) birleşiminden oluşmaktadır. Bu yüzey parçaları, basit yüzey parçaları oluşturmak için üçgen (triangle), üçgen şerit (triangle strip) ve üçgen yelpaze (triangle fan) kullanırken, içinde poligon gibi kapalı alanlardan oluşan boşluklar (hole) bulunan yüzey parçalarını oluşturmak için ring olarak adlandırılan geometri tipleri

kullanmaktadır (ESRI, 1998). Bir multipatch objesi, Şekil 5’de gösterildiği gibi üçgen, üçgen şerit, üçgen yelpaze ya da ring tiplerinden bir veya birkaçının birleşiminden meydana gelmektedir (ESRI, 2008). Multipatch objeyi oluşturan noktasal verilerin eklenme sırası bu tiplerden hangisinin kullanılacağını belirlemektedir. Örneğin Şekil 6’da gösterildiği gibi bir üçgen şerit, eklenen her bir noktanın önceki iki noktayla bağlantılı olarak bir üçgen oluşturduğunu varsayarak üçgen şeritler oluşturmaktadır (Ford, 2004).



Şekil 5. Multipatch objesi için varlık ilişki veri modeli (ESRI, 2008)



Şekil 6. Üçgen şerit oluşturulması (Ford, 2004)

Multipatch objeler, ArcGIS’de direk olarak oluşturulamadığı için ya CityEngine programı kullanarak 3B objeleri multipatch geometri tipine dönüştürerek kullanmakta ya da SketchUp ve Collaborative Design Activity (COLLADA) gibi farklı programlarda üretilen multipatch objeleri kullanmaktadır (ESRI, 2008).

Multipatch, bir coğrafi veri tabanı kaydının geometri alanına atanabildiğinden, ArcObjects aracılığıyla temsil edilmeleri herhangi bir standart ESRI özelliğini aynı şekilde temsil edilmektedir. Multipatch temsili için ilk olarak konumsal referanslı (spatial reference) bir obje sınıfı (feature class) oluşturulması gerekmektedir. Multipatch, çoğunlukla bir girdi obje sınıfından (input feature class) oluşturulduğundan, konumsal referans doğrudan aşağıda gösterildiği gibi türetilmektedir (Ford, 2004):

```
Dim pGeoDataSet As IGeoDataset
Set pGeoDataSet = pInputFeatureClass

Dim pSR As ISpatialReference
Set pSR = pGeoDataSet.SpatialReference
```

Multipatch objeleri temsil etmek için başlangıçta bir *IMultipatch* feature, bir *IGeometryCollection* ve bir *IPointCollection* ile birlikte ArcObjects olarak sunulmuştur. *IGeometryCollection*, aslında üçgen şerit veya üçgen yelpaze gibi geometrilerin bir birleşimi olduğu için *IMultipatch* bir arabirim olarak kullanılmıştır (Ford, 2004).

```
Dim pMultiPatch As IMultiPatch
Dim pGCol As IGeometryCollection
Dim pStrip As IPointCollection

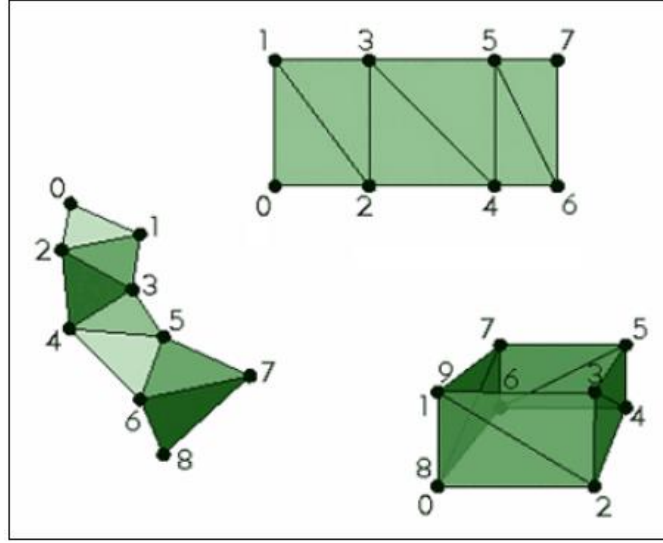
Set pMultiPatch = New MultiPatch
Set pGCol = pMultiPatch
Set pStrip = New TriangleStrip
```

Multipatch objeyi oluşturan her üçgen, üçgen şerit, üçgen yelpaze veya ring aslında bir nokta (point) özelliklerinin toplamı olduğu için bu objelerin temsilinde *IPointCollection* nesnesine ihtiyaç duyulmaktadır. Bu objeler tamamlanmaya kadar her nokta sırayla *PointCollection*'a doğru sırada eklenmektedir. Ardından ArcObjects her üçgenin hangi tarafının dış tarafta olduğunu ve normalleri hesaplayabilmesi içinde üçgeni oluşturan noktaların saat yönünde sıralanmaktadır (Ford, 2004).

Tüm noktalar üçgen, üçgen şerit, üçgen yelpaze veya ring objeleri olarak eklendikten sonra, bu geometri *IGeometryCollection*'a ve dolayısıyla *IMultipatch*'a eklenmektedir. Ardından bir Insert Feature Buffer aracılığıyla Output Feature Class'a eklenmektedir. Böylelikle, ArcObjects tarafından hassas yükseklik bilgisi içeren 3B bir multipatch objesi oluşturulmaktadır (Ford, 2004).

1.5.5.1 Üçgen Şerit

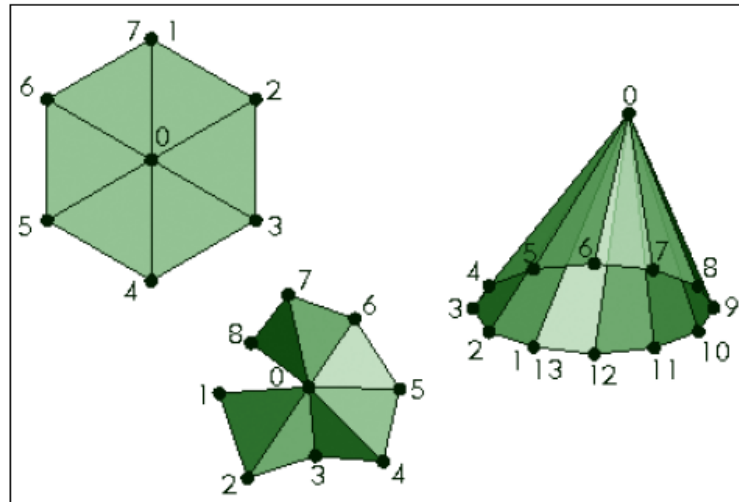
Bir üçgen şerit, köşe noktaları ile birbirine bağlı üçgen parçalarından oluşan sürekli yüzeydir. Her nokta kendinden sonraki ilk iki nokta ile yeni bir üçgen oluşturur (Şekil 7). Örneğin 6 noktalı bir üçgen şerit: (0, 1, 2), (2, 1, 3), (2, 3, 4) ve (4, 3, 5) noktaları ile tanımlanır (ESRI, 2008).



Şekil 7. Üçgen şerit objesi (ESRI, 2008)

1.5.5.2 Üçgen Yelpaze

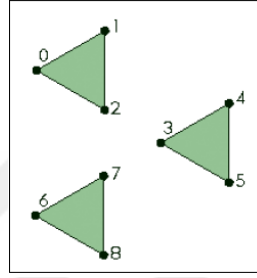
Bir üçgen yelpaze, ilk noktası tepe noktası ya da merkez noktası olarak kabul edilen birbirine bağlı üçgen parçalarından oluşan sürekli yüzeydir. Her nokta kendinden sonraki ilk iki nokta ile yeni bir üçgen tanımlar ve yeni üçgen her zaman köşe noktalarından birbirine bağlıdır (Şekil 8). Örneğin 6 noktalı bir üçgen yelpaze: (0, 1, 2), (0, 2, 3), (0, 3, 4) ve (0, 4, 5) noktaları ile tanımlanır (ESRI, 2008).



Şekil 8. Üçgen yelpaze objesi (ESRI, 2008)

1.5.5.3 Üçgen

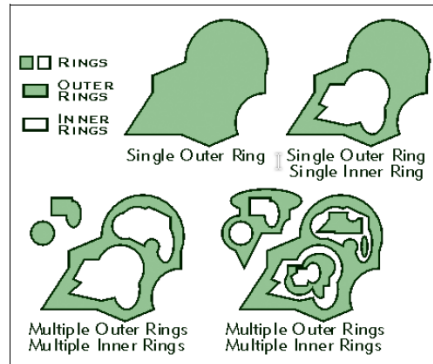
Bir üçgen, birbirlerinden bağımsız üçgen parçalarından oluşan yüzeylerdir. Her ardışık üç nokta ile yeni bir üçgen tanımlanır (Şekil 9). Bu yüzeyler ilk nokta ile birleşerek kapalı alan temsil etmek zorunda değildirler. Birbirinden bağımsız yüzeyler oluşturulması için geliştirilmiştir. Örneğin 6 noktalı bir üçgen: (0, 1, 2) ve (3, 4, 5) noktaları ile tanımlanır (ESRI, 2008).



Şekil 9. Üçgen objesi (ESRI, 2008)

1.5.5.4 Ring

Bir ring, içinde boşluk bulundurabilen poligonlardan oluşan yüzeylerdir (Şekil 10). Ring yüzeyler ile multipatch tanımlanacağı zaman outer ring, inner ring, first ring ve ring adı altında farklı şekilleri temsil ederler. Boşluklar genellikle bir outer ring ve inner ring tanımlanarak temsil edilirler (ESRI, 2008).



Şekil 10. Ring objeleri (ESRI, 2008)

1.6 3B Kent Modelleme

3B CBS için temel altlık olan 3B kent modelleri; arazi yüzeyi, binalar, yollar, ağaçlar ve diğer yeryüzü objeleri gibi konumsal verilerin bilgisayar ortamında 3B temsillerinden oluşmaktadır. Bir 3B kent modelinin temel bileşenleri sayısal arazi modelleri, bina modelleri, sokak objeleri modelleri ve yeşil alanlara ait modellerdir (Döllner vd., 2006). 3B kent modelleri için “Virtual City”, “Cybertown”, “Cybercity” ve “Digital City” gibi farklı isimlerde kullanılmaktadır (Sadek vd., 2002). 3B kent modellerinde, özellikle son yıllarda, fiziksel gerçekliğe mümkün olduğunca yakın bir görünüm elde edilmesine ve kentsel nesnelerin fotogerçekçi görünümünün oluşturulmasına odaklanılmaktadır (Döllner vd., 2006). Kentsel planlama, tesis yönetimi (facility management) ve kişisel yön bulma programları (navigasyon) gibi birçok uygulama, geometri modelinin yanı sıra kentsel objeler hakkında semantik bilgilerin de barındırılmasını gerektirmektedir (Mao, 2011). 3B kent modelleri üretimi aşamasında eğer sadece sunum odaklı modeller üretip tematik/semantik veya topolojik bilgileri yönünden modellenmeleri ihmal edilirse, bu modeller yalnızca görselleştirme amaçları için kullanılabilir, fakat tematik sorgular, analizler ya da konumsal bilgi yönünden kullanılamayacaklardır (Döllner vd., 2006).

3B modelleme kapsamında çeşitli amaçlar için gerekli olan farklı içerik ve detay seviyesindeki gösterimlere olan gereksinim ve bu ihtiyaçların farklı kaynaklardan farklı veri türleri ve araçlar kullanılarak karşılanması zaman içerisinde 3B modelleme konusunda da belirli standartların geliştirilmesi ihtiyacını doğurmaktadır (Doğru ve Şeker, 2009). Farklı veri yapıları ve farklı programlarda geliştirilen 3B kent modellerini depolamak, yönetmek ve entegre etmek için geliştirilmiş bazı standartlar mevcuttur (Mao, 2011). Ağustos 2008’de Open Geospatial Consortium (OGC) tarafından önerilen CityGML, 3B kent modellerini temsil eden ilk uluslararası standarttır (Mao, 2011). 2011 yılında geliştirilen VDI 3805 ve Industrial Foundation Class (IFC) ise diğer standartlar arasında yer almaktadır (Mao, 2011). Gelişen bilgisayar teknolojileri ve programları ile 3B veri yapıları kullanılarak kent modelleri farklı şekillerde oluşturulabilmektedir. Örneğin Türkiye’de İzmir ilinin 3B kent modeli, coğrafi verileri internet üzerinden 3B olarak görebilmeyi ve analizler yapabilmeyi sağlayan CitySurf programı kullanılarak oluşturulmuştur (URL-5, 2017). Google firmasının kullanmakta olduğu SketchUp yazılımı ve ESRI firmasının kullanmakta olduğu CityEngine yazılımı 3B modellemeler yapan programlara örnek olarak verilebilmektedir.

1.6.1 CityGML

Günümüzde birçok CBS yazılımı bulunmakta ve bu yazılımların birçoğunda da kendilerine ait veri depolama formatları bulunmaktadır. Kendi formatlarında depolama yapan CBS yazılımlarından başka bir CBS yazılıma veri paylaşımında ek işlemler yapılması kaçınılmaz olmaktadır (Emem vd., 2008).

Bir yazılım formatında depolanmış grafik bilgilerin bir başka yazılım tarafından okunabilmesi için Veri Değişim Formatı (Data eXchange Format, DXF) ve Genişletilebilir İşaretleme Dili (Extensible Markup Language, XML) gibi formatlar veri değişimini sağlamaktadır. XML, DXF'e göre veri değişimini daha da kolaylaştırmakta ve taşınan verilere de zenginlik katmaktadır. Coğrafi Veri İşaretleme Dili (Geographic Markup Language, GML) ve Ölçeklenebilir Vektör Grafikleri (Scalable Vector Graphics, SVG) coğrafi veriler ile ilişkili XML temelli uygulamalardandır (Emem vd., 2008).

CBS yazılımları arasında veri alışverişi yaparken bir veri yapısındaki verinin başka bir veri yapısında doğrudan depolanmasını sayılabilen GML gibi formatlar kullanılmaktadır. Paylaşılacak olan veriler GML dokümanı haline dönüştürülerek diğer yazılım tarafından okunabilir ve kullanılabilir hale gelmektedir (Emem vd., 2008).

GML, OGC tarafından geliştirilmiş ve 2007 yılında Uluslararası standartlar teşkilatı (International Organization for Standardization, ISO) tarafından 19136 uluslararası standardı olarak kabul edilmektedir. GML, kavram olarak temel şema, uygulama şemaları ve veri dokümanlarını içermektedir. GML veri yapısı verilerin tür, geometri ve topoloji bilgileri başta olmak üzere tamamlayıcı bilgileri içerdiği belirtilmektedir. GML uygulama şemalarından biri ise CityGML olarak verilmektedir. CityGML, kentlerin 3B gösterimi için tasarlanmış genel bilgi modelidir. CityGML'in amacı ise 3B kent modelleri verilerinin depolanması ve değişiminin sağlanması olarak tanımlanmaktadır (Emem vd., 2008).

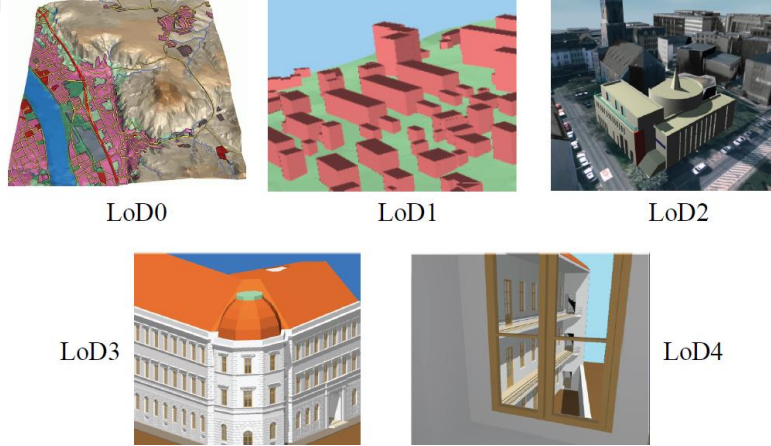
CityGML, 3B kentsel objelerin temsili için ortak bir bilgi modeli olarak tasarlanmıştır (Mao, 2011). Ayrıca CityGML, kent modellerinin sadece şekil ve grafik görünümünü göstermekle kalmamakta aynı zamanda objelere ait semantik bilgilerini de içermektedir (Kolbe, 2009; Mao, 2011). CityGML, Stuttgart, Bonn ve Berlin gibi bazı şehirlerin 3B kent modellenmesi, çevre koruma projeleri, gürültü analizleri, kamusal alanların güvenliği ve trafik simülasyonları gibi alanlarda kullanılmaktadır (Mao, 2011).

Büyük bölgeler için 3B kent modellerinde verilerin fazla olması, veriler üzerinde düzenleme yapmayı ve gösteriminin hızlı olmasını engellemektedir. Günümüzde bu sorun,

modellemenin amaca yönelik olarak farklı detay seviyesinde (Level of Detail, LoD) yapılması ile giderilmektedir (Yücel ve Selçuk, 2009). Farklı detay seviyesinde modelleme ile karmaşık yapı ve büyük kapasiteli 3B kent modellerin iletişimi, paylaşımı, sunumu ve kullanımı daha verimli bir biçimde sağlanabilmektedir (Özdoğan ve Başaraner, 2013).

CityGML, 3B kent modelleri için Şekil 11’de gösterilen 5 farklı detay seviyesini LoD önermektedir (Kolbe vd., 2005). Sıfırıncı seviyeden başlayan bu detay seviyeleri ve kent modelindeki objelerin ne kadar detaylı gösterileceği aşağıda açıklanmaktadır.

- LoD 0, 2.5B sayısal arazi modelidir.
- LoD 1, herhangi bir çatı yapısı veya doku içermeyen blok modelidir.
- LoD 2, çatı yapısı ve dokusu ayır edilebilen modeller ile vejetasyon temsil edilmektedir.
- LoD 3, detaylı duvar ve tavan yapıları ve balkonları olan mimari modelleri ile detaylı vejetasyon ve ulaşım objeleri gösterilmektedir.
- LoD 4, oda, iç kapılar, merdiven ve mobilyalar gibi iç yapıları da temsil edebilmektedir (Kolbe vd., 2005).

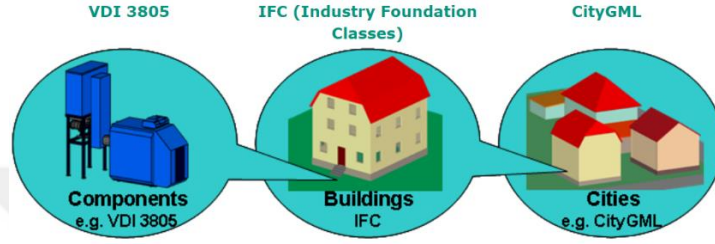


Şekil 11. CityGML tarafından tanımlanan LoD gösterimi (Kolbe vd., 2005)

1.6.2 Diğer Standartlar

CityGML'in yanında, 3B kent modelleriyle ilgili VDI 3805 ve IFC gibi bazı standartlar da bulunmaktadır (Mao, 2011). VDI 3805, Bina Servisleri'nde (BS) ürün veri alışverişi için geliştirilmiş bir endüstriyel standart olarak tanımlanmaktadır. VDI 3805 veri

standartı, ısıtma teknolojisi, havalandırma ve iklimlendirme teknolojisi, sıhhi tesisat mühendisliği gibi tüm bina hizmetleri ekipmanlarını modellemek için bir standart olarak kullanılmaktadır (Mao, 2011). IFC, binaların parçaları ve konumsal ilişkilerini temsil eden veri öğelerini tanımlamaktadır (El-Mekawy, 2010). IFC bir uluslararası standart olup, Danimarka gibi bazı ülkeler tarafından kamuya açık binaları zorunlu olarak IFC formatında modellenmektedir (Mao, 2011).



Şekil 12. VDI 3805, IFC ve CityGML arasındaki ilişki (Mao, 2011)

VDI 3805, IFC ve CityGML arasındaki ilişki Şekil 12'de gösterilmektedir. VDI 3805, bir binadaki bileşenlerin modellenmesi ile ilgili bir standart olarak kullanılmakta iken IFC, bir binadaki parçaların modellenmesi ile bina modelinin oluşturması için kullanılmaktadır. CityGML ise şehir seviyesinde tasarlanmış modellemeler olarak tanımlanmakta ve bitki örtüsü, yol gibi daha fazla şehir nesnelere içermektedir (Mao, 2011).

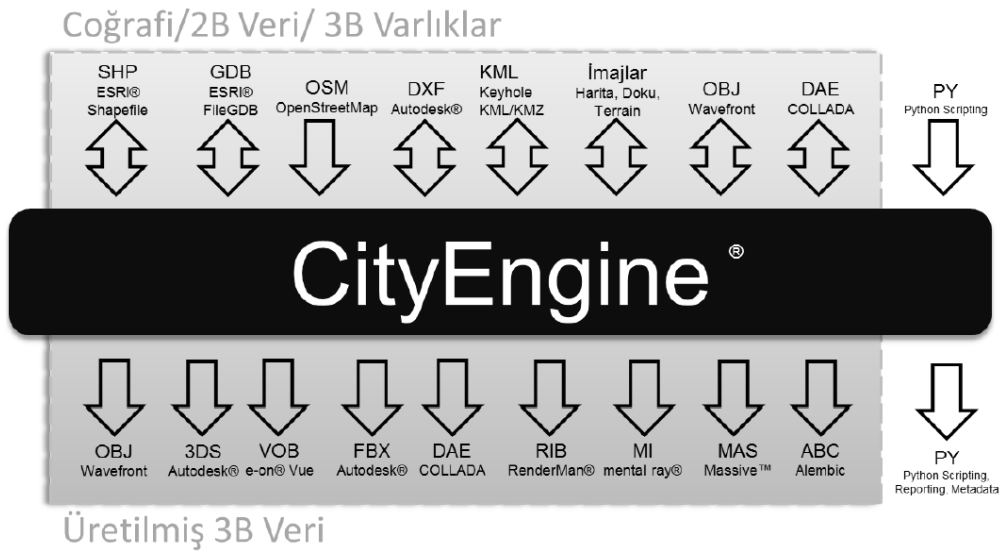
1.6.3 CityEngine

Trimble firmasının kullanıma sunduğu SketchUp programı gibi CAD tabanlı araçlarla bina bazlı 3B modellemeler gerçekleştiren programlar, çeşitli analizler için kullanılacak büyük boyutlu alanlarının 3B kent modellerinin üretimi ekonomik bulunmamaktadır (Müller vd., 2006; Edvardsson, 2013). Ayrıca görsel anlamda inandırıcı bilgisayar modelleri oluşturmak için kullanılan Autodesk, 3D Studio Max veya Maya gibi standart 3B modelleme programları, 3B modelin geometrisini değiştirmek için birçok farklı araç kullanılmaktadır. Bu programlar yardımıyla farklı stillerde bina tipleri oluşturulabilmektedir. Ancak detayların manuel olarak düzenlenmesi fazla detay içeren

binalar ya da kent modelleri üretileceği zaman modelleme maliyetini arttırmaktadır. Bu nedenle 3B modelleme için bir procedural modelleme tekniği olan Computer Generated Architecture (CGA) kullanılmaya başlanmıştır. CGA, 3B modeller üretirken maliyetin düşürülmesini amaçlamaktadır (Lipp, 2007).

Kısa sürede ve düşük maliyetle büyük boyutlu bir 3B kent modeli yaratmak için en iyi yöntem, kurallar (rules) ve algoritmalar kullanarak 3B model oluşturma olan procedural modelleme yöntemidir. Kullanılan kurallar ve algoritmalar ile modelin kullanıcı tarafından düzenlenmesi yerine kuralların düzenlenmesi ile model oluşturmaya odaklanılmaktadır (Singh vd., 2014). ESRI firması tarafından Ağustos 2008’de kullanıma sunulan CityEngine programı, procedural modelleme yaklaşımı ile büyük boyutlu alanların 3B kent modellemesini gerçekleştirebilmektedir (Dobroja, 2015). Böylelikle CityEngine programı, kullandığı CGA veri mimarisi sayesinde binalar, yollar, parklar ve ağaçlar gibi bütün detaylarıyla 3B kent modellerini kısa sürede, düşük maliyetle ve otomatik olarak üretebilmektedir.

CityEngine, bir alana ait 3B kent modelini oluşturmakta ve kent modelini diğer kullanıcılara paylaşmak için doğrudan internette yayımlayabilmektedir (Singh vd., 2014). Ayrıca ESRI Shapefile, File Geodatabase ve OpenStreetMap gibi veri formatlarını destekleyerek herhangi bir coğrafi veriyi içe ve dışa (import ve export) aktarmaya izin vermektedir (URL-7, 2017). Şekil 13’de içe ve dışa aktarılabilen veri formatlarının genel görünümü yer almaktadır (ESRI, 2013).



Şekil 13. CityEngine için içe ve dışa aktarılabilen formatlar (ESRI, 2013)

Singh vd. (2014) tarafından yapılan çalışmada CityEngine programı ile 3B kent modelleri üretmenin bazı avantajları ve dezavantajları aşağıdaki gibi sıralanmaktadır.

CityEngine programı kullanmanın avantajları;

- Bir şehrin geniş bir alanı kısa sürede ve az maliyetle modellenebilmektedir.
- Bir şehrin geniş bir alanı az kuraldan üretilebilmektedir.
- Kullanıcı şehir modelini manuel olarak düzenleyebilmektedir
- Procedural modelleme ile bir kural setinden bir model oluşturulmaktadır
- Kent modeli, diğer formatlardan alınabilmektedir
- Kent modeli, çeşitli uygulamalar için diğer yazılımlara aktarılabilir
- Mevcut binaların güncellenmesi ve yeni binaların modele eklenmesine olanak sağlamaktadır.
- Gerçek zamanlı 3B kent modelleme imkânı sunmaktadır.
- Katman bilgileri kolay yoldan değiştirilebilmekte ve eklemeler yapılabilmektedir.
- OpenStreetMap verileri CityEngine programında doğrudan açılabilir.
- Cephe modellemeleri için özel bir fotoğraf formatına ihtiyaç duyulmamaktadır.

CityEngine programı kullanmanın dezavantajları;

- Procedural programlama dilini anlamak kolay olmadığı için modelleme için programlama bilgisi gerekmektedir.
- 3B modelleme yapabilmek için Python programlama dilinin bilinmesi gerekmektedir.
- Bina boyunu doğrudan ölçme imkânı sunmamaktadır.
- Animasyonlar üretilememekte ve oynatılamamaktadır.
- Cephe modellemesi için fotoğrafın dikdörtgen veya paralel olması gerekmektedir.
- Cephe modellemesinde Snapping gibi kullanım kolaylığı sunan bir araç bulunmamaktadır.
- Cephe modellemesinde fotoğraflarda düzenleme yapmak gerekmektedir.
- Her cephe için bir görüntü gerekmesi projedeki iş yükünü arttırmaktadır.

1.7 CBS Yazılımları Tarafından Sunulan 3B Analiz Fonksiyonları

ArcGIS programı 3D Analyst, Earth Resources Data Analysis System (ERDAS) programı Imagine VirtualGIS, Intergraph programı GeoMedia Terrain ve PCIGeomatics programı PAMAP GIS Topographer programları ve sunmuş oldukları analiz araçları CBS pazarının büyük bir kısmını oluşturmaktadır. Bunlar, 3B sunum ve analizler için bir çözüm sağlamaya çalışan sistemler olarak sıralanabilmektedir. Bu programların sahip oldukları 3B analiz modülleri Zlatanova vd. (2002) tarafından yapılan çalışmada ayrıntılı olarak açıklanmaktadır.

ArcGIS yazılımının sunmuş olduğu 3D Analyst modülü kullanılarak çeşitli konumsal analizler gerçekleştirilebilmektedir. Ağırlıklı olarak vektör veriler ile çalışmakta olan 3D Analyst modülü temelde, SYM gibi 2.5B modeller üretebilmektedir. Ayrıca arazi örtüsünün görselleştirilmesi gibi 2.5B vektör verilerinin görüntülenmesini iyileştirmek için raster verilerini kullanabilmektedir (Zlatanova vd., 2002).

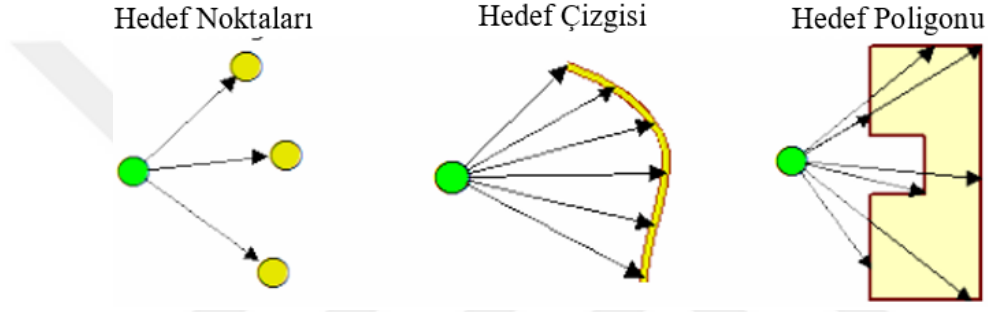
ArcGIS, 3B objelerin temsili için multipatch veri yapısını kullanmaktadır. ArcGIS programlarından biri olan ArcScene arayüzü ile birlikte 3B objelerin gösterimini sağlanabilmekte fakat 3B modellerin “True 3D” olarak üretimi gerçekleştirilememektedir. Bu nedenle ArcGIS programları, SketchUp ve CityEngine gibi programlarda multipatch veri yapısında üretilen 3B modellerinin kullanılmasını sağlamaktadır. Böylece ArcGIS ortamında 3B modeller görüntülenebilmekte ve 3D Analyst araçlarının sunduğu analizler gerçekleştirilebilmektedir.

Görünürlük analizi, en sık kullanılan CBS analiz fonksiyonlarından biridir (Popelka ve Vozenilek, 2015). ArcGIS yazılımının sunmuş olduğu 3D Analyst modülünün alt modüllerinden biri olan Visibility modülü kullanıcılara çeşitli görünürlük analiz fonksiyonları sağlamaktadır.

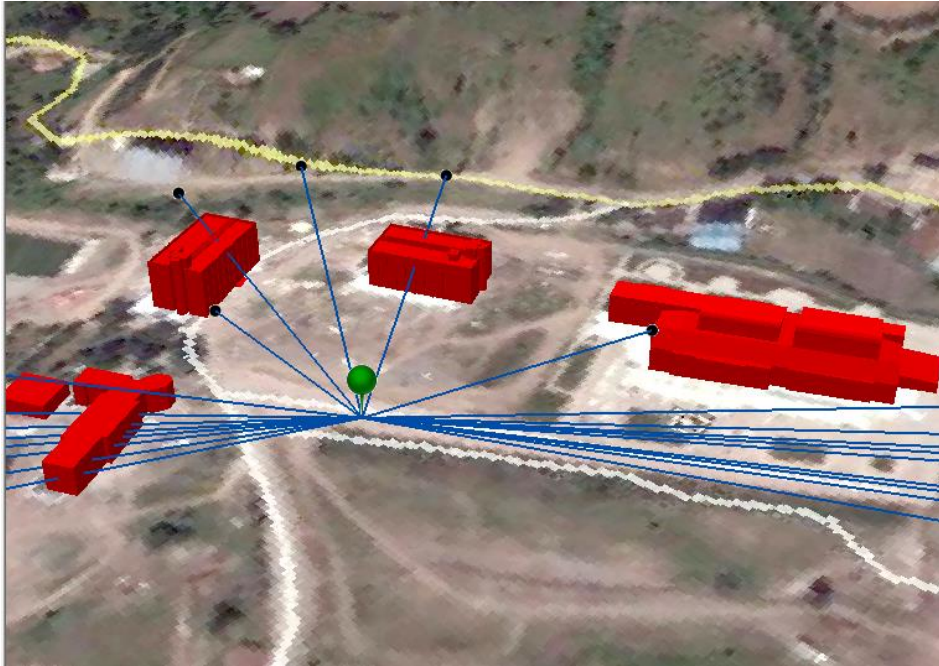
ArcGIS 3D Analyst modülünün içerdiği görünürlük analizleri; Görüş Hatları Çizme Analizi (Construct Sight Lines), Görüş Analizi (Intervisibility), Görüş Hattı Analizi (Line of Sight), Gözlemci Noktaları Analizi (Observer Points), Gölge Analizi (Sun Shadow Volume), Görüş Alanı Analizi (Viewshed), Görünürlük Analizi (Visibility), Skyline, Skyline Bariyer Analizi (Skyline Barrier) ve Skyline Grafik Analizi (Skyline Graph) olarak sıralanmaktadır. Bu analizlerin açıklamaları ve kullanım şekilleri ilerleyen bölümlerde açıklanmaktadır.

1.7.1 Görüş Hatları Çizme Analizi

Görüş hatları çizme analizi, belirlenecek bir durulan noktadan bakılacak olan nokta, çizgi veya poligon gibi bir hedefe görüş çizgileri oluşturmaktadır. Şekil 14’de gösterildiği üzere durulan yer nokta özelliğinde, bakılan hedef ise nokta, çizgi veya poligon özelliğinde belirlenebilmektedir. Bakılan hedef çizgi veya poligon olarak belirlendiği zaman üretilecek görüş çizgileri arasındaki örnekleme sıklığı (Sampling Distance) değeri ayarlanarak belirlenebilmektedir. Şekil 15’de bir 3B model üzerinde görüş hattı analizi gösterilmektedir.



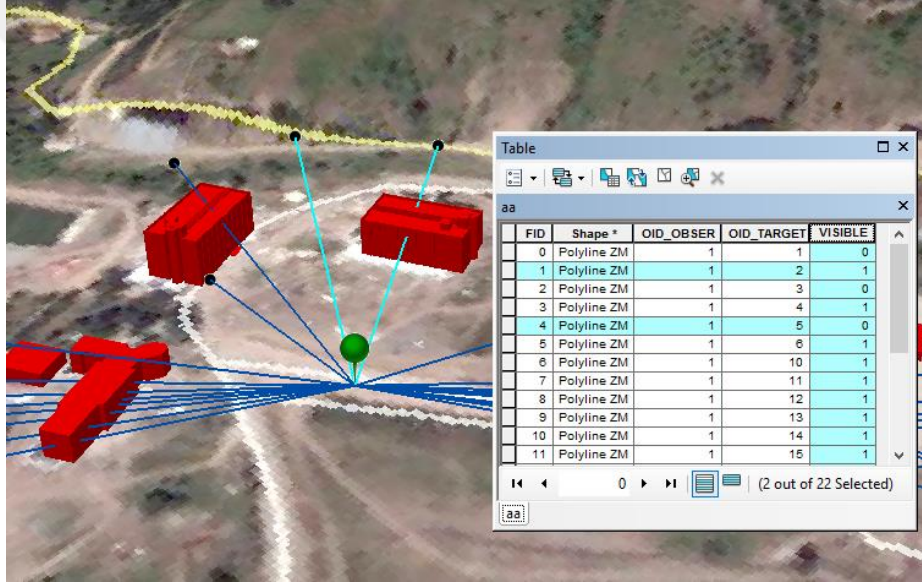
Şekil 14. Görüş hatları için durulan nokta ve bakılan hedefler (ArcGIS, 2013)



Şekil 15. Görüş hatları çizme analizinin gösterimi

1.7.2 Görüş Analizi

Arazi yüzeyleri, binalar veya engel teşkil eden herhangi bir objeden geçerek görüş hatlarının (Sight Lines) görünürlüğüne belirlemek için kullanılmaktadır (ArcGIS, 2013). Görüş hattının görünürlüğüne belirleyen engeller için raster, TIN, yükseklik atanmış poligon (extrud of polygon) veya multipatch objeler kullanılmaktadır. Görüş hattı bu engeller ile kesiştiği zaman hattın öznitelik tablolarına eklenecek bir sütun ile görünür anlamına gelen 0 veya görünmez anlamına gelen 1 değerleri atamaktadır. Böylelikle durulan bir noktadan bakılan hedefin kesintisiz olarak görüp görünmediği belirlenebilmektedir. Şekil 16’da bir 3B model üzerinde görüş analizi gösterilmektedir.

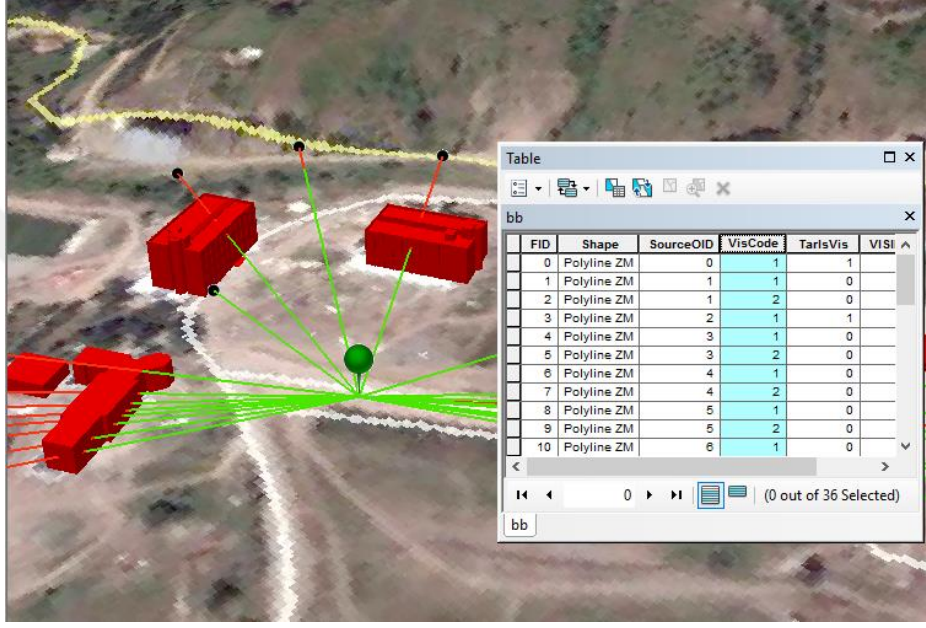


Şekil 16. Görüş analizinin gösterimi

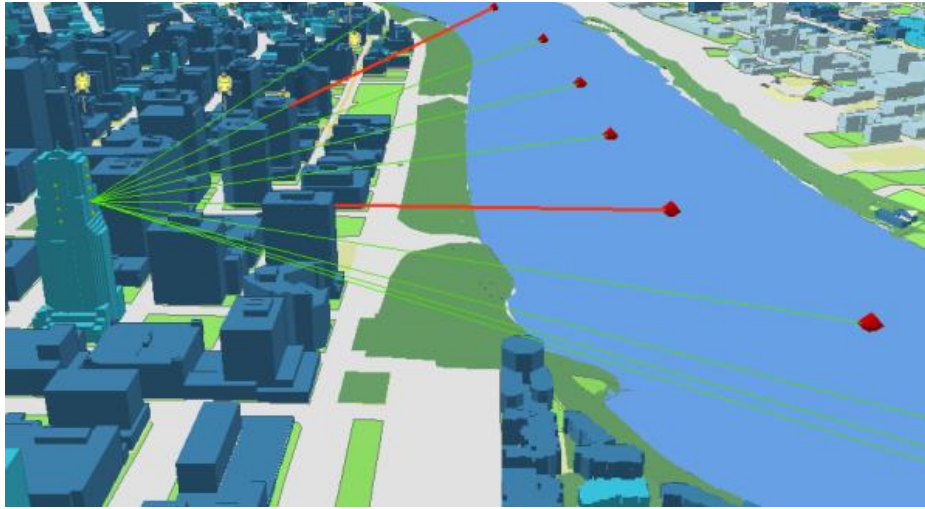
1.7.3 Görüş Hattı Analizi

Görüş hattı analizi, görüş analizine benzer şekilde, engel teşkil eden herhangi bir objeden geçerek görüş hatlarının (sight lines) görünürlüğüne belirlemektedir. Fakat Görüş hattı analizi, önceden üretilmiş görüş hatlarını, dikkate alınan engele göre parçalara ayırmaktadır ve isteğe bağlı olarak parçalara ayrılan çizgilerin yerlerini noktasal veri olarak da üretebilmektedir. Ayrıca görüş hattının öznitelik tablolarına VisCode adıyla bir sütun ekleyerek görünür ve görünmez doğru parçalarına görünür anlamına gelen 1 ve

görünmez anlamına gelen 2 değerleri atamaktadır. Böylelikle görüş hattının görünen parçalarını ve görünmeyen parçalarını ayrı ayrı görsel olarak sunmaktadır. Şekil 17’de VisCode değeri 1 olan ve görünür kısım olarak belirtilen görüş hattı parçaları yeşil renkte, değeri 2 olan ve görünmeyen kısım olarak belirtilen görüş hattı parçaları ise kırmızı renkte gösterilmektedir. Görüş hattı analizi, Şekil 18’de gösterildiği gibi şehir için önemli bir noktanın görünürlüğünü analiz edilecek uygulamalarda kullanılabilir.

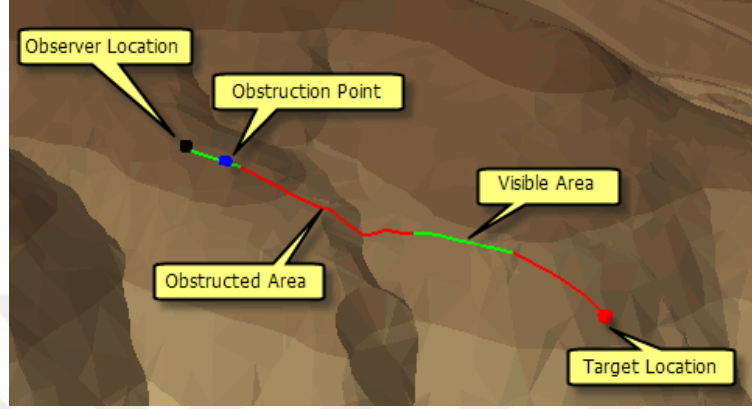


Şekil 17. Görüş hattı analizi gösterimi



Şekil 18. Görüş hattı analizi uygulaması (URL-8, 2016)

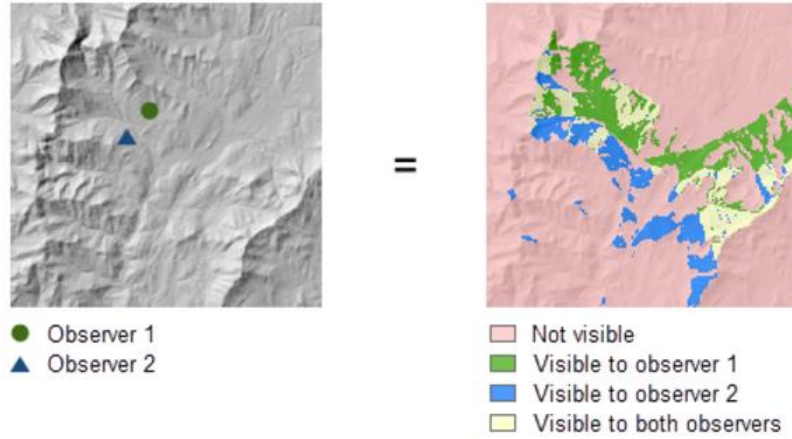
ArcGIS 3D Analyst modülünün içerdiği görüş hattı analizinin dışında ArcMap arayüzü ile sunmuş olduğu 3D Analyst araç çubuğu da görüş hattı analizi gerçekleştirmektedir. Yaygın olarak kullanılan bu analizin sonuç ürünleri genellikle Şekil 19'daki gibi yüzey modellerinde gösterilmektedir (Liu vd., 2008).



Şekil 19. ArcMap görüş hattı analizinin uygulanması (URL-9, 2016)

1.7.4 Gözlemci Noktaları Analizi

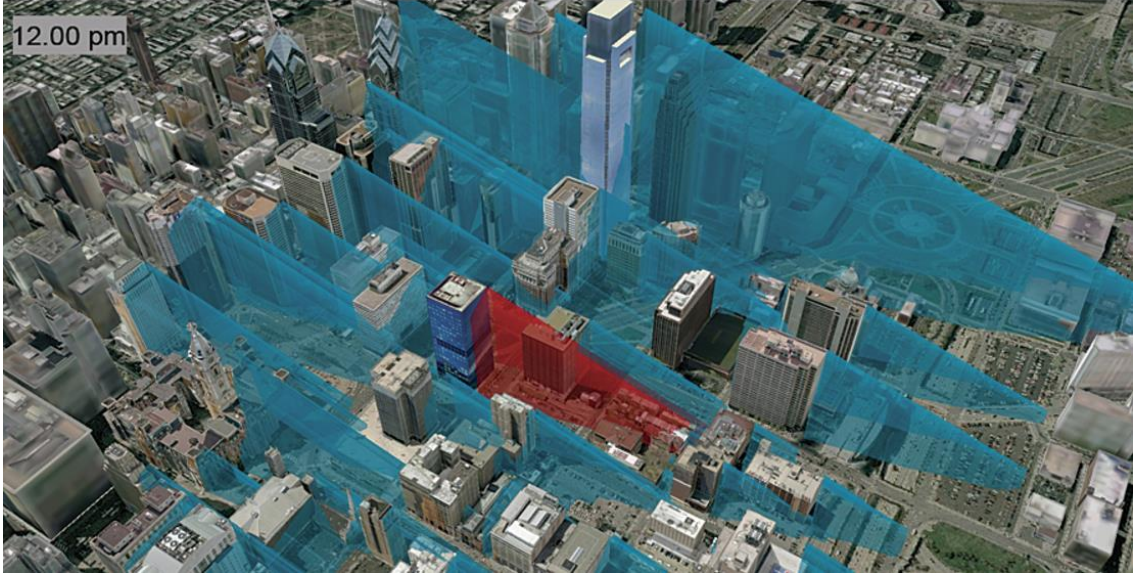
Durulan noktanın raster yüzeyden görülebilen yerlerini bir raster veri olarak üretmektedir. Maksimum 16 noktadan görünürlüğü değerlendirebilmektedir. Şekil 20'de iki ayrı durulan noktanın, yüzeyin tamamına ait görünürlükleri gösterilmektedir (URL-10, 2017).



Şekil 20. Gözlemci noktaları analizi gösterimi (URL-10, 2016)

1.7.5 Gölge Analizi

Gölge analizleri, belirli bir tarih ve saat aralığındaki güneş ışığını kullanarak her bir obje için multipatch veri yapısında gölgeler temsil edilen kapalı hacimler oluşturmaktadır. Gölge analizi için yükseklik atanmış çizgi, poligon veya multipatch objeler kullanılmalıdır. Gölgeleri modellemek için güneş ışığının yörüngesinin hesaplanacağı tarih ve saat belirlenmelidir (ArcGIS, 2013). Şekil 21’de geçici ya da zamana duyarlı bir analiz olan gölge analizleri, belirlenen bir binanın gölgesinin farklı zamanlardaki etkisi gösterilmektedir. Şekilde mavi renk mevcut binalardaki gölgeleri temsil ederken, kırmızı renk öğleden sonraki yeni gölgeyi göstermektedir (URL-11, 2016).

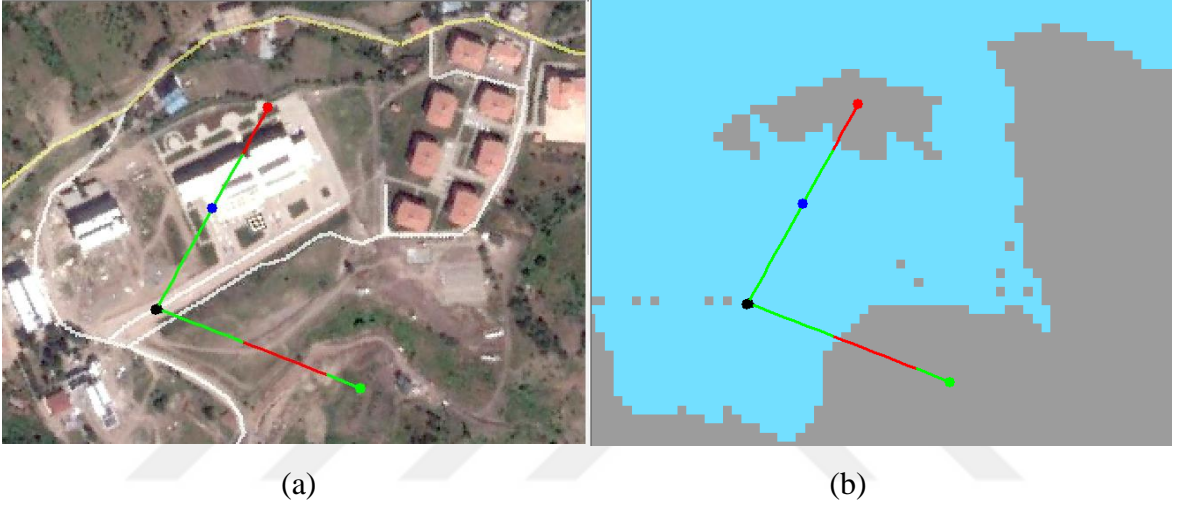


Şekil 21. Gölge analizi gösterimi (URL-11, 2016)

1.7.6 Görüş Alanı Analizi

Görüş alanı analizi, gözlem yapılan durulan noktadan raster formattaki arazi yüzeyindeki konumlardan nerelerin görünemez olduğunu belirlemektedir. Görüş alanı analizi sonuç haritasını, görünür ve görünmez olarak iki değerle gruplandırılan bir raster veri olarak üretmektedir. Şekil 22 (a)’da durulan noktadan yüzey modeline bir görüş alanı analizi uygulanmış ve sonuç ürünü Şekil 22 (b)’de gösterilmiştir. Burada gri alanlar görünemeyen, mavi alanlar ise görünen alanları göstermektedir.

Görüş alanı analizi, gözlemcinin (durulan noktanın) bulunduğu pikseldeki ve hedef pikselindeki yükseklik farkını belirleyen bir algoritma kullanmaktadır. Hedef pikselin görünürlüğünü belirlemek için, gözlemci ile hedefi birbirine bağlayan çizgide yer alan her piksele görüş hattı analizi uygulanarak görünürlük belirlenmektedir. Gözlemci ve hedef pikselleri arasında bunların yüksekliğinden daha yüksek bir değere sahip bir piksel olduğu durumda hedef piksel “görünür değildir” olarak işaretlenmektedir (Ghosh ve Rao Tv, 2014).



Şekil 22. Görüş alanı analizi gösterimi

1.7.7 Görünürlük Analizi

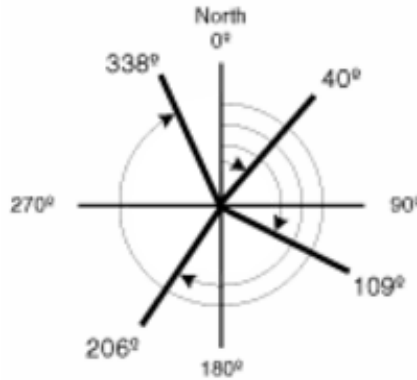
Görünürlük analizi, görüş alanı analizi gibi gözlem yapılan durulan noktandan arazi yüzeyi konumlarının görünürlüğünü değerlendirmektedir. Ayrıca ArcGIS Pro versiyonu ile binalar gibi yüzey üzerinde bulunan ve durulan nokta için engel teşkil eden verileri de analize katarak engeller üzerindeki görünürlüğü değerlendirebilmektedir. Üretilen harita öznetelik tablolarına eklenen Value sütünü ile görünür alanları 1, görünemeyen alanları ise 0 değeri atayarak belirlemektedir. Görünürlük analizi, farklı alanlarda Çoklu Görüş Alanı (Multiple Viewshed) olarak da adlandırılabilir. Görünürlük analizi, “Verilen gözlem yerinden hangi yerler görülür?” ve “Verilen nesne veya yerden kaç tane gözlem yeri görülür?” sorularına cevap bulmaktadır. Örneğin bazı önemli yapıların aynı anda nerelerden görüldüğünü belirlemekte kullanılabilir (Ghosh ve Rao Tv, 2014).

1.7.8 Skyline Analizi

Skyline analizi, arazi yüzeyi, bina modelleri veya engel teşkil eden herhangi bir obje modelini dikkate alarak gözlem yapılacak bir durulan noktadan görünen yerleri belirleyen bir polyline veya multipatch veri yapısında sonuçlar üretmektedir. Skyline analizi sonuç ürününde her köşe noktası, bir görüş hattı boyunca en uzak görünür noktayı temsil etmektedir (URL-12, 2017).

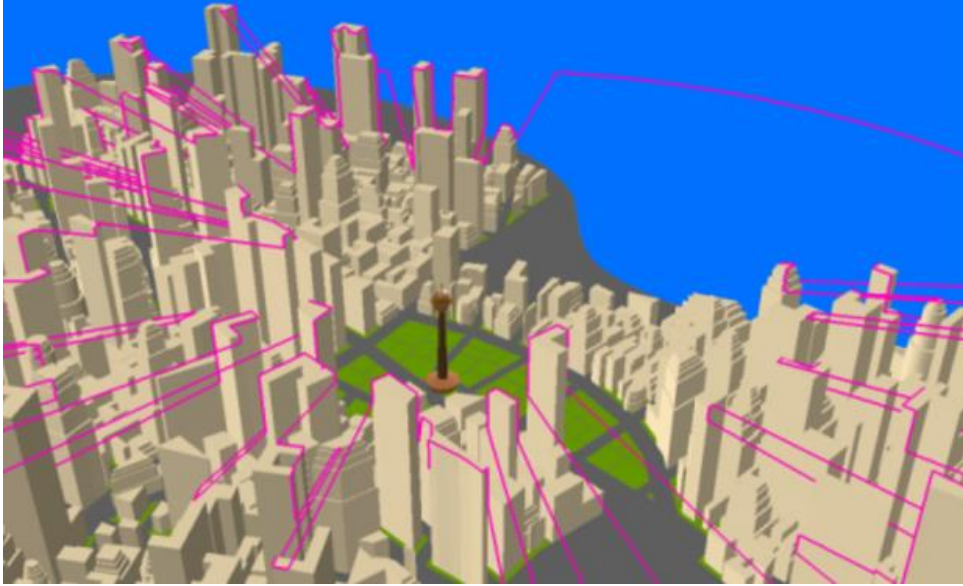
Skyline analizi, bir yüzey modeli dikkate alınmadan gerçekleştirileceği zaman bir ufuk tanımlayabilmek için Virtual Surface Radius veya Virtual Surface Elevation parametreleri kullanarak bir sanal yüzey sınırları belirlemeye imkân sunmaktadır. Ayrıca Maximum Horizon Radius kullanılarak gözlemcinin bulunduğu yerden bir ufuk aranması gereken maximum mesafe belirlenebilmektedir. Bununla birlikte Feature Level of Detail parametresi ile görünürlük analizi için dikkate alınacak obje modellerinin her köşe noktası (üçgen yüzeylere ait), sadece yüzey üzerindeki noktaları (footprints) veya dış yüzeyinin yükseltilmiş noktaları gibi vektör verilerine ait detay sınırlamasını yapabilmektedir.

Skyline analizi, Azimuth parametreleri kullanılarak, durulan noktadan görüş açısını belirleyebilmek için başlangıç (From Azimuth) ve bitiş (To Azimuth) açılarda değerleri ve görüş açısı aralığında kaç derecede aralıklarla analiz yapılacağı (Azimuth Increment) derece olarak belirlenebilmektedir. Azimuth Increment değeri azaldıkça daha dar açılarla analiz yapılarak daha çok obje dikkate alınmakta ve böylece analiz detayı artmaktadır. ArcGIS ortamında From Azimuth ve To Azimuth değerleri Şekil 23'de gösterilen kuzey azimuth açısı sistemi ile temsil etmektedir. Burada bir çizginin azimutu, kuzeyden saat yönünde ölçülen yatay açı değeri olarak belirlenmektedir (URL-13, 2017).



Şekil 23. ArcGIS Azimuth açısı temsili (URL-13, 2)

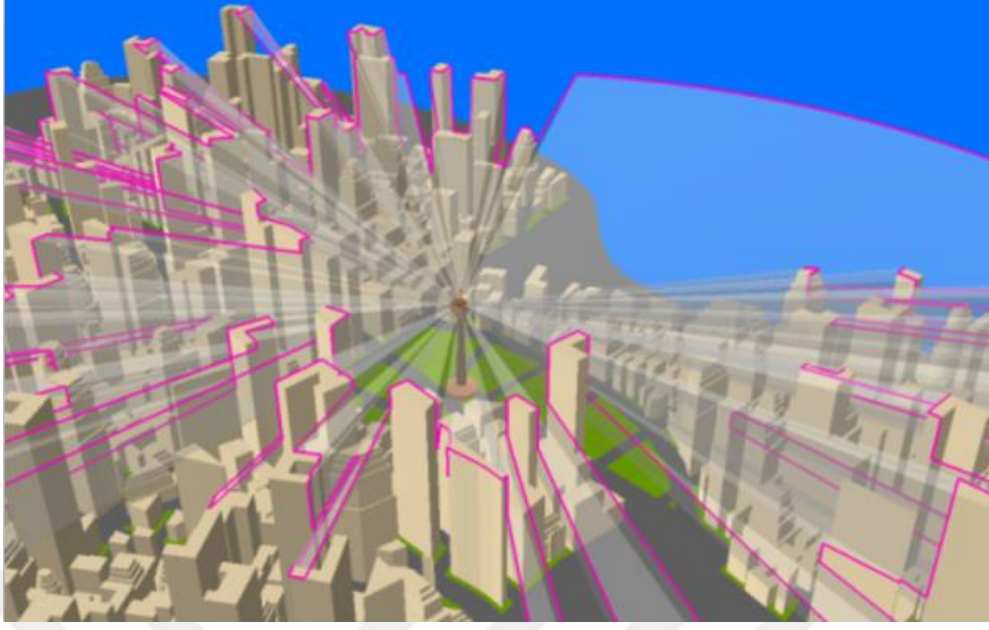
Skyline analizi, Skyline veya Silhouette analizi olarak iki ayrı analiz yapılabilmektedir. Create Silhouette parametresi kullanılarak gölge hacimleri (Shadow Volume) oluşturmak için skyline bariyer aracı tarafından kullanılabilen multipatch veri yapısında bir sonuç üretilmektedir. Silüet, genellikle dikey olacak ve gözlemcinin perspektifinden özelliklerin hemen arkasına yerleştirilecek çoklu parçadır (URL-12, 2017). Gerçekleştirilecek analiz eğer bir Skyline analizi ise sonuç ürünü bir polyline veri yapısında üretilmektedir. Skyline analizi üretileceği zaman No Segment Skyline parametresi kullanıldığı zaman sonuç polyline öznelik tablolarında tek bir özellik olarak kayıt ederken, Segment Skyline kullanıldığında her kırılma noktasını ayrı bir özellik olarak kayıt etmektedir. Şekil 24'de şehrin merkezinde bulunan kuleden yüzey modeli ve mevcut binalar kullanılarak üretilmiş Skyline analizi gösterilmektedir (URL-14, 2016).



Şekil 24. Skyline analizi gösterimi (URL-14, 2016)

1.7.9 Skyline Bariyer Analizi

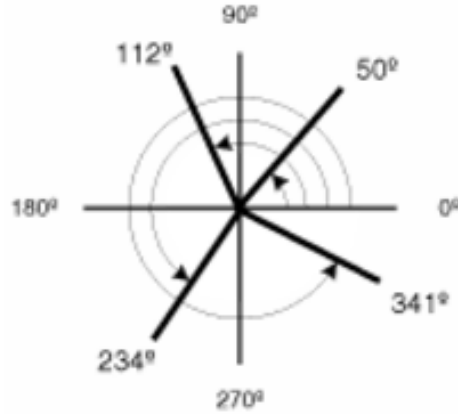
Skyline Bariyer ya da gölge hacimleri olarak temsil edilen multipatch veri yapısında bir sonuç üretmektedir (ArcGIS, 2013). Skyline Bariyer analizi, Skyline analizi ile üretilmiş bir Skyline verisi ile durulan nokta arasında bulunan alanları multipatch veri yapısında kapalı alanlar olarak üretmektedir. Şekil 25'de şehrin merkezinde bulunan kuleden Skyline analizi kullanılarak üretilmiş skyline bariyer analizi gösterilmektedir.



Şekil 25. Skyline bariyer analizi gösterimi (URL-14, 2016)

1.7.10 Skyline Grafik Analizi

Skyline grafik analizi, Skyline analizi ile üretilmiş bir Skyline ürününün, durulan noktadan her bir skyline noktası için yatay ve düşeydeki açı değerlerini içeren bir tablo ve grafik üretmektedir. ArcGIS’de bu grafikler kutupsal açıları Şekil 26’da gösterildiği gibi pozitif x ekseninden saatin ters yönünde ölçülerek temsil edilen Polar Graph olarak üretilmektedir (URL-13, 2017).

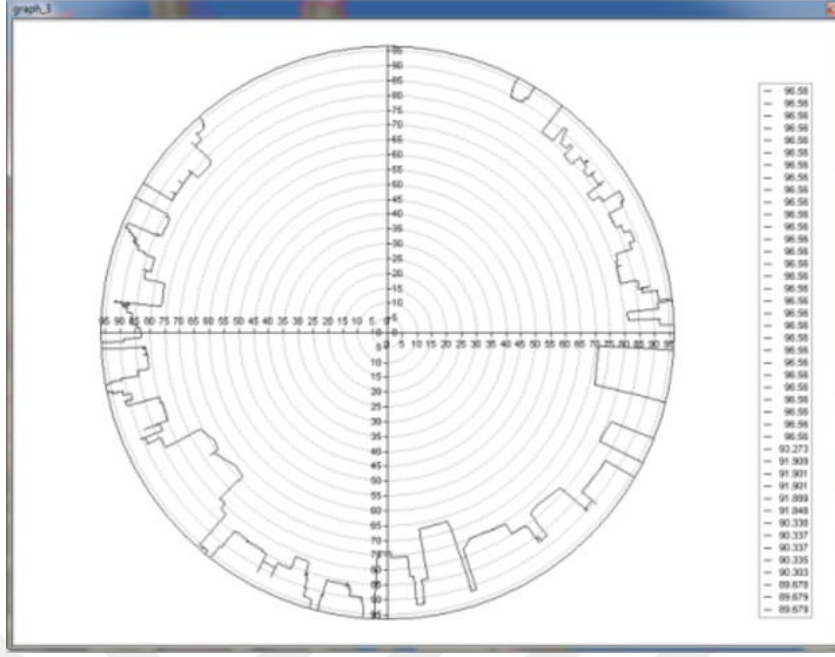


Şekil 26. ArcGIS kutupsal açı temsili (URL-13, 2017)

Polar Graph, gözlem yapılan merkez noktasından yakınlık derecesini tasvir etmek için kullanılan kutupsal bir grafik olarak tanımlanmaktadır (URL-14, 2016). Polar Graph'da grafiğin üzerindeki halkalar gözlem yapılan durulan noktadan olan mesafeyi temsil etmektedir. Ayrıca Skyline analizdeki Virtual Surface Radius ve Maximum Horizon Radius parametresi ile bu mesafe sınırlandırılabilir. Yani Skyline analizinde durulan noktadan en uzaktaki skyline çizgisi, grafiğin en dış halkasını göstermektedir. Şekil 24'deki Skyline analizi kullanılarak üretilmiş olan Skyline grafik analizine ait yatay ve düşey açı değerlerinin bulunduğu tablo Şekil 27'de gösterilmektedir. Zenit açılarının gösterildiği grafik ise Şekil 28'de verilmektedir (URL-14, 2016).

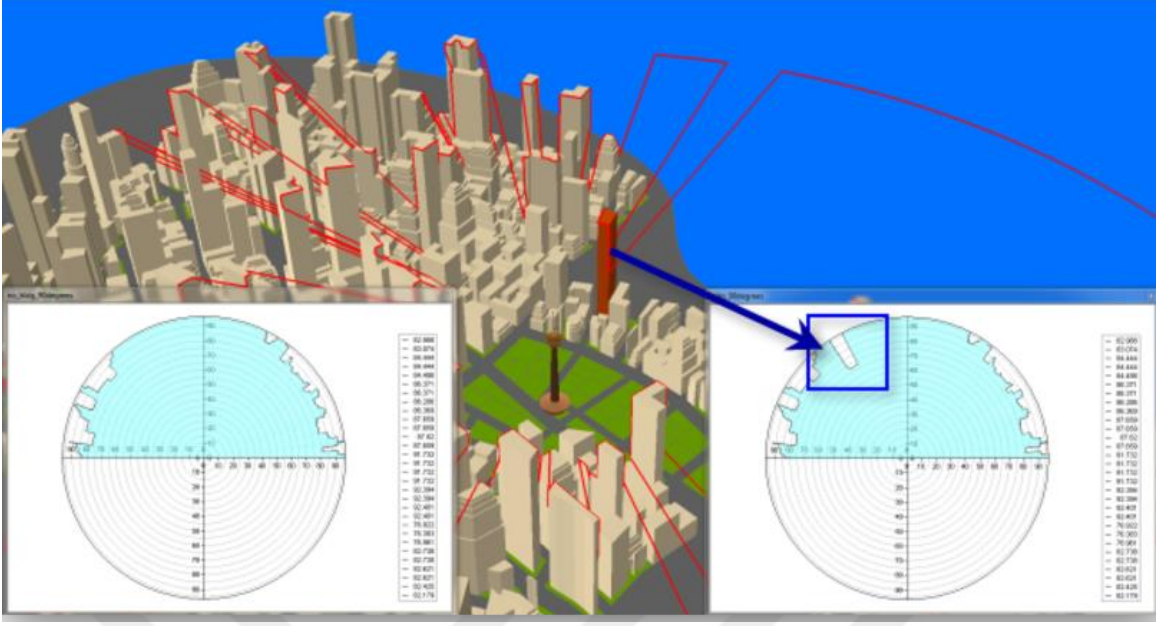
OBJECTID *	HORIZ_ANG	ZENITH_ANG
1	90	96.560196
2	89	96.560196
3	88	96.560196
4	87	96.560196
5	86	96.560196
6	85	96.560196
7	84	96.560196
8	83	96.560196
9	82	96.560196
10	81	96.560196
11	80	96.560196
12	79	96.560196
13	78	96.560196
14	77	96.560196
15	76	96.560196
16	75	96.560196
17	74	96.560196
18	73	96.560196
19	72	96.560196
20	71	96.560196
21	70	96.560196
22	69	96.560196
23	68	96.560196

Şekil 27. Skyline grafik analizi tablo gösterimi (URL-14, 2016)



Şekil 28. Skyline grafik analizi grafik gösterimi (URL-14, 2016)

Bir şehir içindeki binalar, ağaçlar ve sokak mobilyaları gibi yoğun fiziksel yapı yapılarının yanı sıra parklar veya şehir meydanları gibi açık alanların bulunduğu özel yerlerin farkındalığını artırmak için kentsel alanlar Skyline analizi ile daha iyi analiz edilebilmektedir. Örneğin kent simgesi haline gelmiş turistik bir kuleden okyanus manzarası görünümünü korumak için Skyline analizleri kullanılabilir. Yüksek katlı binalar etrafındaki turizm açısından önemli binaların, otellerin veya sahildeki dairelerinin de görüş alanını etkileyerek gayrimenkulü değerini değiştirmektedir. Bu nedenle kentsel alanlar planlanırken plancılar ve geliştiriciler yüksek katlı yeni bir bina inşa edileceği zaman görünürlük analizlerine ihtiyaç duymaktadırlar. (ArcGIS, 2013). Böyle bir analiz skyline, skyline bariyer ve skyline grafik analizleri kullanılarak gerçekleştirilebilir. Şekil 29’da yeni yapılacak bir binanın şehir merkezindeki kulede bulunan restoranın deniz manzarasına etkisini analiz etmektedir. Burada önce mevcut binalara ait skyline, skyline bariyer ve skyline grafik üretilmektedir. Ardından yeni yapılacak bina da mevcut kent modeline eklenerek tekrar bu analizler gerçekleştirilmekte ve üretilen skyline grafikleri karşılaştırılmaktadır. Böylelikle yeni yapılacak binanın etkileri skyline analizleri ile değerlendirilmiştir.



Şekil 29. Yeni binanın silüet görüntüsüne etkisi (URL-14, 2016)

ArcGIS ortamında sunulan görünürlük analizleri kullanılarak 3B kentsel alanlarda kullanılabilir çeşitli görünürlük analizleri yapılabilmektedir. Ancak sunulan görünürlük analizleri ile kentsel alanlardaki silüet durumunun analiz edilebilmesi için yapılması gereken birçok adım ve bilinmesi gereken birçok parametre bulunmaktadır. Bu işlem adımları ve parametreler kullanılarak elde edilen sonuç ürünleri ile yeni yapılacak bir binanın silüeti ve korunmak istenilen binalara ait silüet durumunu analiz edilememektedir. Yeni yapılacak binanın silüet durumunu bozmaması için inşa edilmesi gereken yükseklik veya kat adedi ile ilgili bilgilere ulaşılamamaktadır.

2. YAPILAN ÇALIŞMALAR

Bu tez kapsamında yapılan çalışmalar üç başlık altında ele alınmaktadır. Birincisi; CityEngine programı kullanılarak örnek bir çalışma alanındaki binaların 3B bina modelinin üretilmesi 2.1 başlığında anlatılmaktadır. Üretilen 3B modeldeki binalar için Python programlama dili kullanılarak silüet analizini gerçekleştirecek olan silüet analiz modülünün geliştirilmesi 2.2 başlığında verilmektedir. Ayrıca geliştirilen silüet analiz modülünün ArcGIS yazılımına entegre edilmesi için bir araç çubuğu tasarımı yapılmış ve araç çubuğu tasarımı 2.3 başlığında açıklanmıştır. Son olarak, silüet analiz modülünün çalışma prensibi ve analiz çıktıları örnek bir uygulama ile 2.4 başlığında anlatılmıştır.

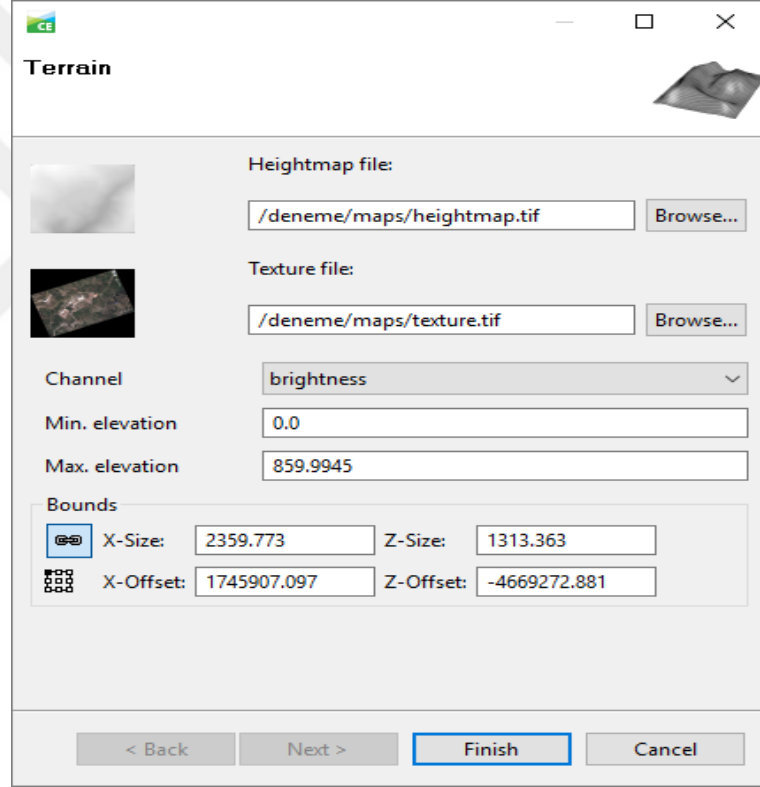
2.1. 3B Bina Modeli Üretimi

ArcGIS programında silüet analizini gerçekleştirmek için ilk olarak örnek bir çalışma alanının 2B bina katmanları ve SYM üretilmiştir. Günümüzde CBS için kullanılan halihazır haritalar 2B olduğu için çoğu verilerde yükseklik bilgisi olan z değeri bulunmamaktadır. Bu nedenle bu çalışmada 2B verilerden coğrafi referanslı 3B bina modellerinin otomatik olarak üretilmesi amaçlanmaktadır. Çalışmada, 3B bina modelleri üretiminde kullanılan bina katmanları, öznelik tablolarında bina kat adetleri bulunan ve yükseklik bilgisi içermeyen poligon geometri tipindedir. Çalışmada kullanılan SYM ise eşyükselti eğrileri kullanılarak üretilmiştir. 3B kent modelini tamamlayıcı nitelikte olan yollar ve diğer yeryüzü objeleri de 2B haritalardan üretilebilmektedir.

ESRI firmasının kullanıma sunduğu CityEngine v.2015.2 programı kullanılarak çalışma alanına ait 2B bina katmanından 3B bina modelleri üretilmiştir. CityEngine'de bulunan Navigator ana penceresinden proje dosyalarının depolanarak veri yönetimi sağlanmaktadır. Burada CityEngine programına veri aktarılacağı zaman Import özelliği kullanılabilir veya sürükleyip bırakılarak ekranda gösterimi sağlanabilmektedir. Her format (ESRI Shape, geodatabase, DXF, KML, vb.) için farklı arayüzde tasarlanmış aktarım penceresi ekranda açılarak o formata uygun bilgiler ayarlanması ile verinin programda gösterimi sağlanmaktadır.

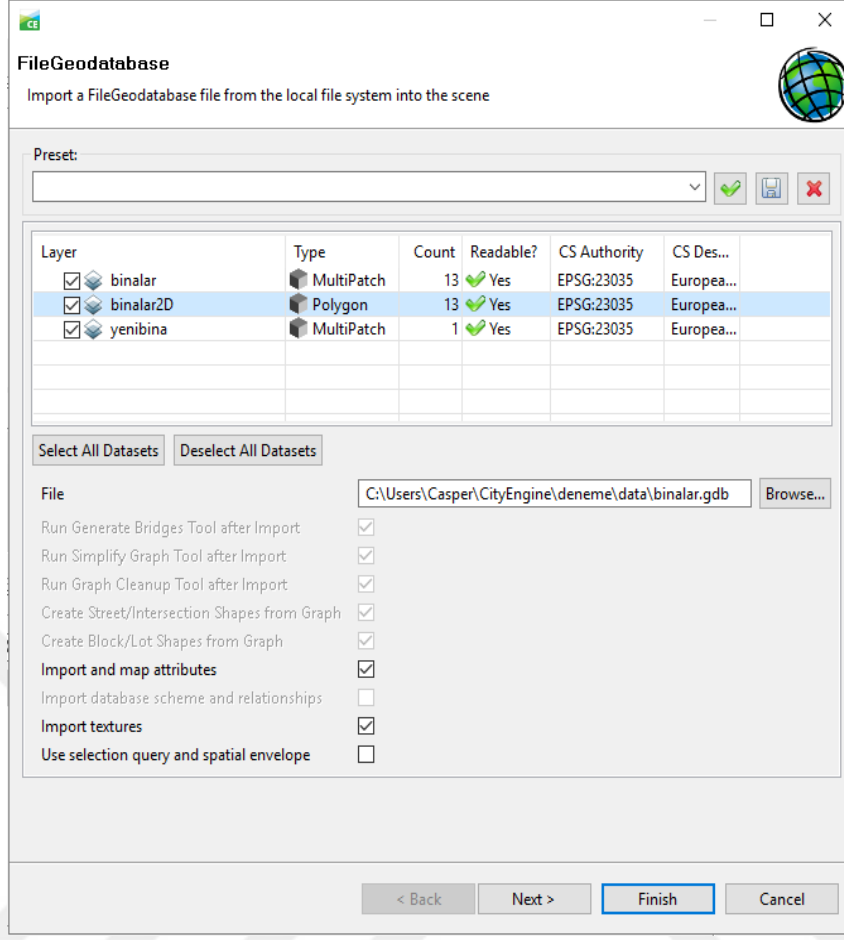
Üretilen yüzey modeli verilerinin CityEngine programına aktarımı için Import pencerelerinden biri olan Terrain penceresi açılacaktır. Ekranı gelen Terrain penceresi

Şekil 30’da gösterilmektedir. Burada Heightmap File (SYM dosyası) ve Texture File (doku dosyası) kısmından raster formattaki veriler seçilerek arazi yüzeyinin programda görüntülenmesi sağlanmaktadır. Heightmap File ve Texture File verilerinin koordinat sistemleri ve hücre boyutlarının aynı olması gerekmektedir. Coğrafi referanslı verilerde Terrain penceresindeki minimum ve maximum yükseklikler ve sınır bilgilerinin bulunduğu diğer parametreler otomatik olarak doldurulmaktadır. Sınır bilgileri olan X Size ve Z Size metre cinsinden arazi boyutlarını, X Offset ve Z Offset ise ekranda konumlandırılacak noktanın koordinatlarını belirtmektedir. Çözünürlük gibi parametreler ise Inspector ana penceresinden değiştirilebilmektedir.



Şekil 30. Terrain penceresi

Çalışmada kullanılan bina katmanının CityEngine programına aktarımı için ise File Geodatabase isimli Import penceresi açılacaktır. Ekranı gelen File Geodatabase penceresi Şekil 31’de gösterilmektedir. Burada ESRI Shapefile formatındaki nokta, çizgi, poligon veya multipatch geometri tipindeki verilerin programda gösterimi sağlanmaktadır (ESRI, 2013).

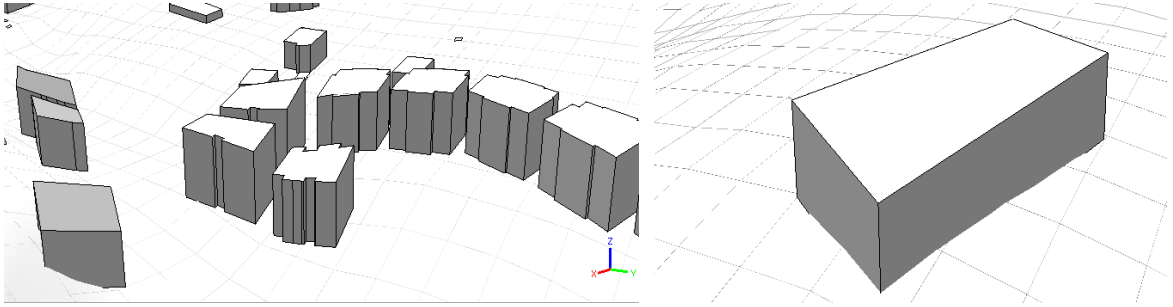


Şekil 31. File Geodatabase penceresi

CityEngine ile 3B modeller üretileceği zaman yükseklik değeri (CityEngine yüksekliği y değeri ile temsil etmektedir) içeren bina katmanları kullanılarak kural tabanlı modelleme yaparken binaların her bir verteksi kural dosyasındaki değeri kadar yükseltilerek modellenecektir. Bu durumda oluşan bina modelleri Şekil 32’de gösterildiği gibi olacaktır. Bu nedenle 2B bina katmanlarından 3B bina modelleri üretileceği zaman 2B verilerin yani bina taban yüzeyinin arazi yüzeyine oturtulması gerekmektedir. Özellikle de eğimli araziler üzerine kurulu kentler için bu işlem büyük önem arz etmektedir. CityEngine, katmanların arazi yüzeyine oturtulması işlemleri için Align Terrain to Shapes ve Align Shapes fonksiyonlarını kullanmaktadır (ESRI, 2013). Align Terrain to Shapes, arazi yüzeyini belirlenen şekillere oturtmaktadır. Align Shapes ise kullanıcıya farklı fonksiyon seçenekleri sunarak şekilleri belirlenen bir arazi yüzeyine oturmaktadır. 3B bina modelleri üretilirken şekilleri araziye ortması konusunda dikkat edilmesi gereken bazı

durumlar bulunmaktadır. Aşağıda verilen bilgiler ile Align Shapes fonksiyonunun şekillerin arazi yüzeyine oturma yöntemleri açıklanmaktadır.

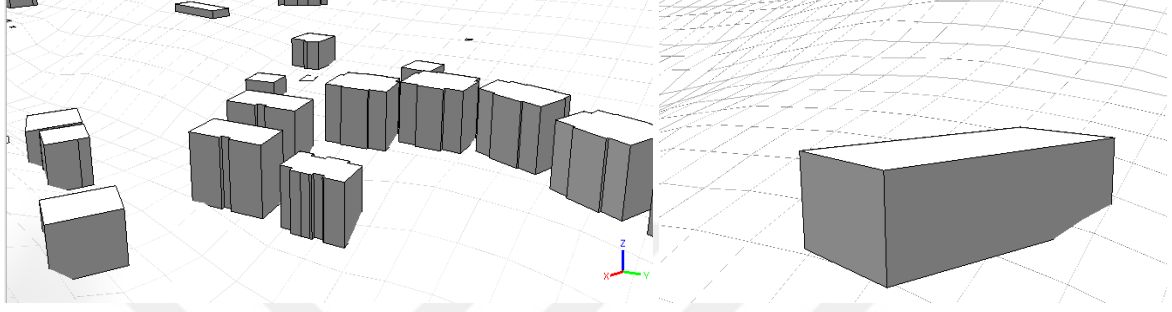
- Project All, seçili bütün şekillerin köşe noktalarının yükseklik bilgisini arazide karşılık bulunduğu pikseldeki yükseklik değerine ayarlamaktadır.
- Project Below, oturulacak olan yüzeyin altında kalan bütün şekillerin köşe noktalarının yükseklik bilgisini arazide karşılık bulunduğu pikseldeki yükseklik değerine ayarlamaktadır.
- Project to Object Average, seçili bütün şekillerin köşe noktalarının yükseklik değerinin ortalamasını belirleyerek arazide ayarlamaktadır.
- Translate to Average, seçili bütün şekillerin köşe noktalarının yükseklik bilgisini arazide karşılık bulunduğu piksellerdeki ortalama yükseklik değerine ayarlamaktadır.
- Translate to Maximum, seçili bütün şekillerin köşe noktalarının yükseklik bilgisini arazide karşılık bulunduğu piksellerdeki en yüksek yükseklik değerine ayarlamaktadır.
- Translate to Minimum, seçili bütün şekillerin köşe noktalarının yükseklik bilgisini arazide karşılık bulunduğu piksellerdeki en düşük yükseklik değerine ayarlamaktadır.



Şekil 32. Project All kullanılarak üretilmiş 3B bina modelleri

Bir bina taban yüzeyi arazi yüzeyine oturtulacağı zaman bütün noktaların arazide karşılık geldiği yüksekliğe ayarlanması için Project All kullanılması mantıklı bir düşünce olacaktır. Fakat CityEngine ile kural tabanlı model üretileceği zaman, Project All kullanılarak bir bina modellendiğinde, binalar Şekil 32'de gösterildiği gibi modellenecektir.

Bu çalışmada 2B halihazır haritalardan 3B modeller üretmek amaçlandığı için ve 2B halihazır haritada bina katmanlarına ait yükseklik bilgisi bulunmadığı için bu tür çalışmalarda Translate to Average kullanılması en uygun yöntem olarak görülmektedir. Bina modelleri üretimi için Translate to Average kullanıldığı zaman Şekil 33'de gösterildiği gibi modellenecektir.



Şekil 33. Translate to Average kullanılarak üretilmiş 3B bina modelleri

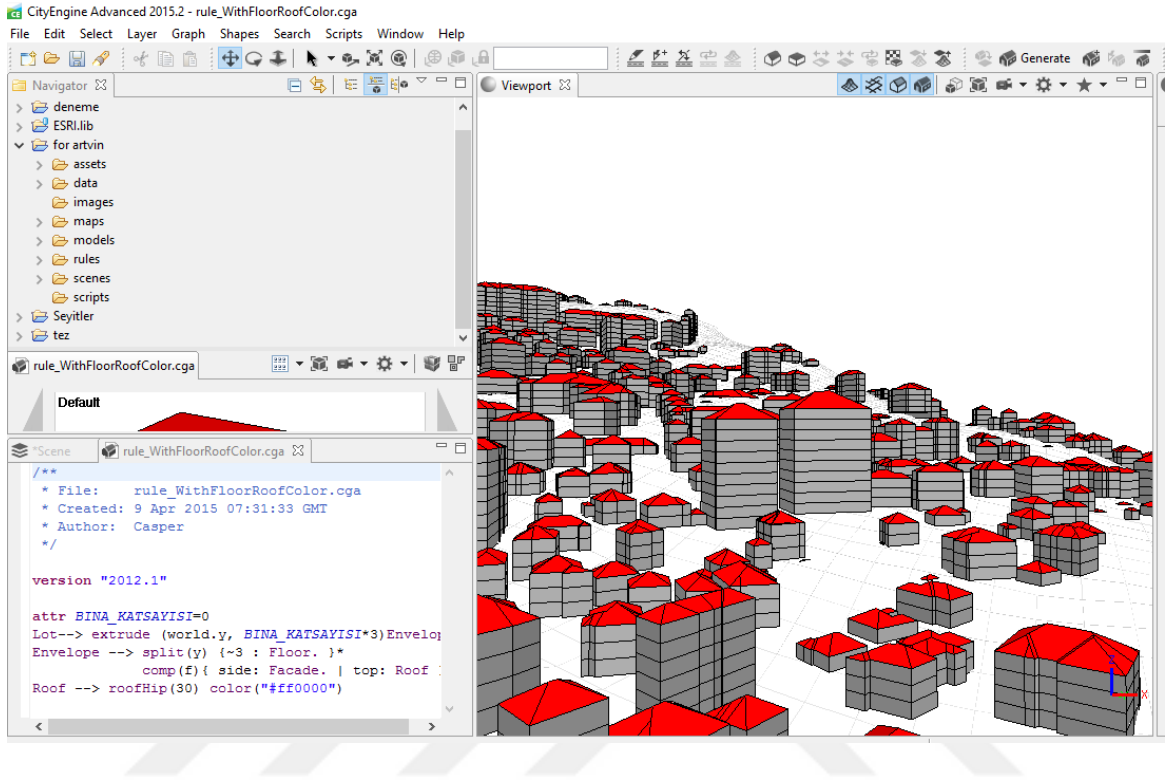
CityEngine, gerçekçi 3B model oluşturmak için geometri ve özniteliklerle tanımlanmış kuralları kullanmaktadır. Binaların 2B temsili olan footprint'ler yükseklik özniteliğine göre yükseltılarak (extrusion) 3B yapılara dönüştürülürler, daha sonra binalar kat adedi, zemin yüksekliği ve pencere konumu gibi diğer öznitelik bilgileri ile kurallara göre binaların 3B modelleri üretilmektedir (Şekil 34) (Ghosh ve Rao Tvb, 2014).



Şekil 34. Kurallar kullanılarak 3B bina modelleri üretimi (Ghosh ve Rao Tvb, 2014)

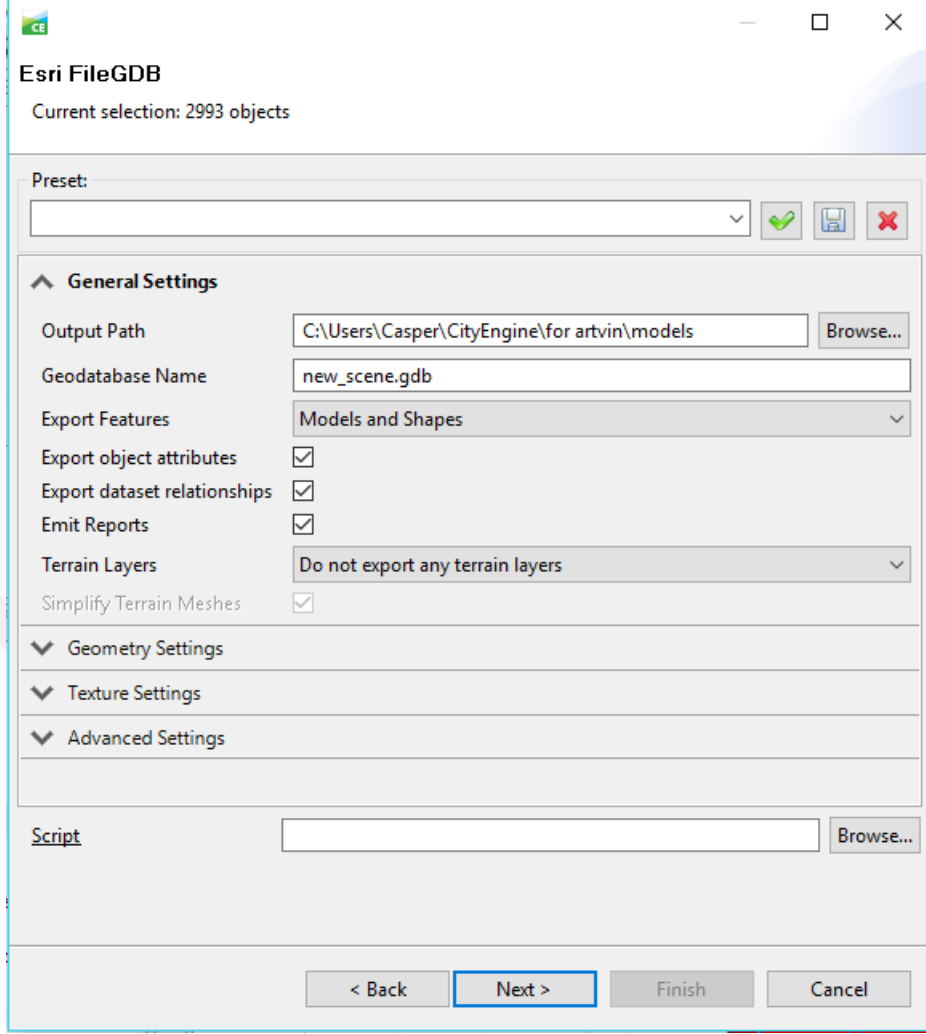
Bu çalışmada, 2B bina katmanlarının CityEngine programına aktararak gösteriminin sağlanmasının ardından kurallar kullanılarak 3B bina modellerinin pratik olarak üretimi sağlanmıştır. En basit şekliyle binaların kat ve çatıların modellenmesi için Şekil 35'de bulunan kodlar kullanılarak şekildeki bina modelleri otomatik olarak

üretilebilmektedir. Modelleme için bina katmanlarının öznelik tablolarında bina adetleri gibi özelliklerin bulunması gerekmektedir.



Şekil 35. Kurallar kullanılarak oluşturulan 3B bina modelleri

CityEngine programında, model üretimi tamamlandıktan sonra gerekli konumsal analizlerin gerçekleştirilebilmesi için verinin dışa aktarımının sağlanması gerekmektedir. Burada CityEngine programında bulunan verilerin dışa aktarımı için Export özelliği kullanılabilir. Programda üretilen modellerin dışa aktarımı ise Export Models fonksiyonu kullanılarak sağlanmaktadır. Bu çalışmada 3B bina modelleri ArcGIS ortamında kullanılmak üzere Esri FileGDB kullanılarak multipatch veri yapısında dışa aktarımı gerçekleştirilmektedir. Ekranı gelen Esri FileGDB penceresi Şekil 36'da gösterilmektedir.



Şekil 36. Esri FileGDB penceresi

2.2. Silüet Analizi Modülünün Geliştirilmesi

ESRI ArcMap ve ArcScene yazılımlarında bulunan 3D Analyst modülü kullanılarak çeşitli 3B konumsal analizler gerçekleştirilebilmektedir. ArcGIS, 3B modelleri ya 2B modeli extrude yaparak ya da multipatch veri yapısında temsil etmektedir. ArcGIS, 3B modellerin gösterimini ise sadece ArcScene veya ArcGlobe yazılımlarını kullanarak yapabilmektedir. ArcMap programı 3B görselleştirmeyi sağlayamamaktadır.

ArcGIS yazılımlarının, Python programlama dili kullanarak genişletilebilir olduğu önceki bölümlerde dile getirilmişti. ArcGIS, genişletilmek için sahip olduğu modülleri “arcpy” kütüphanesi ile birlikte kullanıcıya sunmaktadır. Python’da kod kütüphanesi “modül” terimi ile ifade edilmektedir (URL-15, 2016).

Arcpy modülüne ait özel fonksiyonların kullanılabilmesi için verilerin kaynak alındığı bir proje dosyası gerekmektedir. arcpy.mapping ile mapping fonksiyonu işlemlerin arka planda gerçekleştirileceği bir “.mxd” dosyası oluşturmaktadır. ArcGIS yazılımları, üzerinde çalışılan proje dosyalarını ArcMap programı için “.mxd”, ArcScene programı için “.sxd” ve ArcGlobe programı için “.3dd” dosyası olarak kayıt etmektedir.

Bu çalışmada, arcpy kütüphanesi kullanılarak silüet analizi gerçekleştiren bir modül geliştirilmiştir. arcpy.mapping fonksiyonu ArcScene için “.sxd” veya ArcGlobe için “.3dd” proje dosyaları oluşturma seçeneğini henüz sunmadığı için bu çalışmada geliştirilen modül, ArcMap ortamında kullanıma sunulmaktadır. Bu sebeple programlama sırasında ilk yapılması gereken şey arcpy kütüphanesinin içe aktarılması olmalıdır.

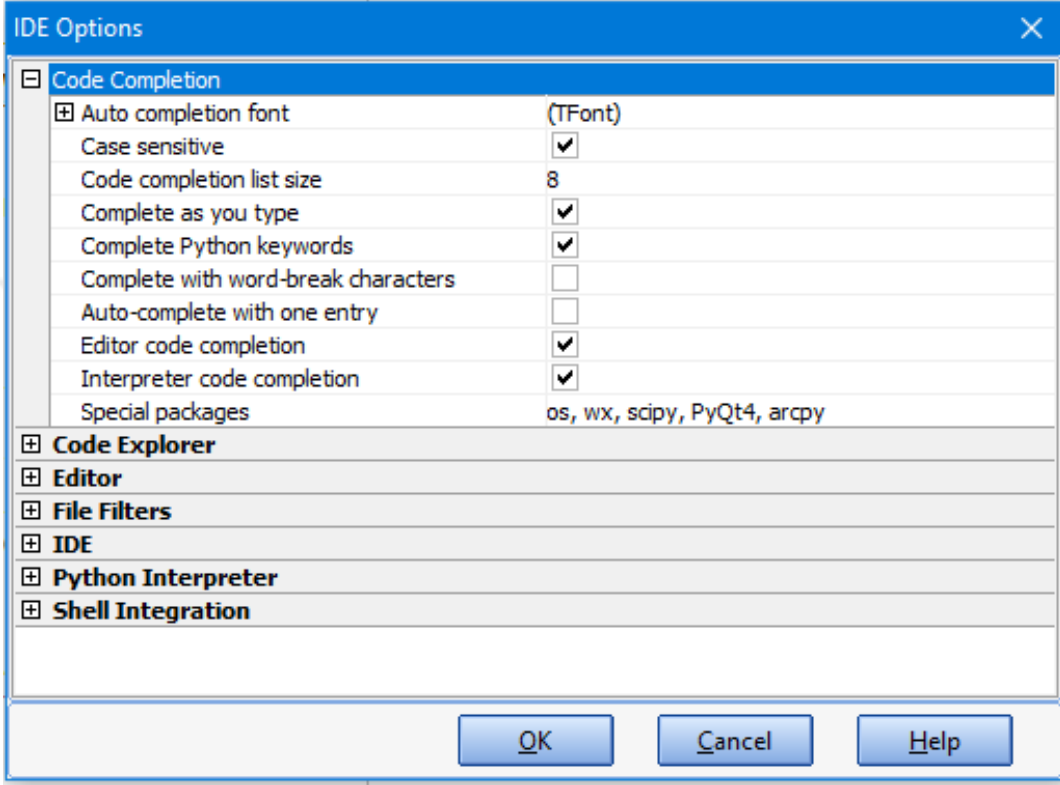
Programlamaya yönelik bilgiler URL-15’da (2016) Python programlama diline ait kütüphaneler, fonksiyonlar, modüller, değişkenler, koşula bağlı durumlar, döngüler, listeler, demetler ve sözlükler gibi birçok temel bilgi ve açıklamalar ise URL-16 (2016) ve URL-17’de (2016) detaylıca açıklanmaktadır.

Özet olarak, arcpy.py dosyası içinde bazı özel fonksiyonlar bulunan bir modül olarak düşünülürse, import arcpy kodu ile arcpy.py dosyasındaki arcpy modülünün bütün içeriğini programlama yapılacak olan programın içine aktarılmaktadır. Bir modülün fonksiyonları içe aktarılırken, bütün fonksiyonların aktarılması yerine, belirli bazı fonksiyonları içe aktarmakta mümkündür. Fonksiyonların aktarımının ardından arcpy kütüphanesinde daha önceden geliştirilmiş olan Buffer_analysis, Buffer3D_3d, DeleteFeatures_management veya Skyline_3d gibi fonksiyonlar, fonksiyonların içinde bulunan değişkenler tanımlanmak üzere kodlama yapılacak programda hazır hale gelmektedir. Örneğin DeleteFeatures_management fonksiyonu için silinecek verini tanımlandığı “in_features” parametresi bu fonksiyon için tanımlanmış bir değişkendir.

ArcMap programı için geliştirilen silüet analiz modülünün içerdiği işlemler Install klasöründe bulunan Kod_addin.py dosyasına kodlanması gerekmektedir. Bilgisayar terminolojisinde, uygulama geliştirmek için gerekli kod yazma ortamları ya da kod editörleri, Integrated Development Environment (IDE) olarak isimlendirilmektedir. Python programlama dili kullanılarak uygulama geliştirebilecek Komodo, Netbeans, PyCharm, Eclipse, Wing, Pyshield, Spyder, IDLE, IdleX, IEP, PyStudio, Sublime Text, PyScripter ve IDLE gibi birçok IDE bulunmaktadır (URL-18, 2016).

ArcGIS programı kullanıcılarına masaüstü CBS yazılımları ile birlikte IDLE editörünü de sunmaktadır. Fakat bu çalışmada, daha pratik olduğu düşünülen PyScripter

editörü kullanılarak silüet analizi için gerekli kodlar yazılmıştır. PyScripter kurulumu tamamlandıktan sonra arcpy modülünün etkin olarak kullanılması için tek yapılması gereken işlem arcpy modülünün Şekil 37'deki gibi Special Packages kısmına arcpy yazılarak programa entegre edilmesidir.



Şekil 37. IDE Options penceresi

Bu çalışmada, tasarlanan silüet analiz modülü ile 3B kentsel alanlarda silüet analizleri gerçekleştirmek amaçlanmıştır. 3B kentsel alanların kullanıcı tarafından belirlenecek olan bir durulan nokta ve bir görüş alanındaki belirli binalara ait silüet görüntüsü üretilmektedir. Ayrıca mevcut binalardan silüetinin korunması önem arz eden bina modeli seçilerek, silüeti korunmak istenen bina ile yeni yapılacak olan bina arasında silüet analizi gerçekleştirilmektedir. Bu analiz ile yeni binanın, korunan bina silüetini bozmaması için maksimum bina yüksekliği ve yapılabilecek maksimum kat adedi değerleri hesaplanabilmektedir. Böylelikle kentsel alanlarda yeni bir bina yapılacağı zaman silüet analizi uygulanarak yeni yapılacak binanın silüet durumu hem görsel hem de sayısal olarak yorumlanabilmektedir. Geliştirilen silüet analiz modülü, kentsel planlamada plancılara ve

imar planı uygulamalarında belediye yetkililerine karar verme aşamasında öncülük edeceği düşünülmektedir.

Silüet analiz modülü ile analizlerin gerçekleştirilmesi için SYM ile 3B bina modellerine ihtiyaç duyulmaktadır. Silüet analizi için 3B modelde kullanıcı ekranda herhangi bir yerden durulan noktayı ve silüet hattını belirleyebilmektedir. Silüet hattının başlangıç ve bitiş noktalarının, durulan noktaya göre semt (azimuth) açıları hesaplanmakta ve bu değerler Skyline analizinde bulunan From Azimuth ve To Azimuth parametreleri olarak kullanılmaktadır. Ayrıca geliştirilen silüet analiz modülü, silüet hattı kullanılması ile modeldeki bütün binalar yerine, silüet hattındaki binalar işleme olarak, analiz performansını arttırmaya yönelik tasarlanmıştır.

Analize giren binaların belirlenmesinin ardından Skyline analizleri yardımı ile binaların durulan noktaya göre görünür olan noktaları belirlenmektedir. Görünür noktalar, multipatch geometri tipindeki 3B bina modellerinden üretildikleri için 3B veri yapısında üretilmektedir. Silüet görüntüsü ise 2B bir görüntü olduğu için görünür noktaların 2B temsil edilmesi gerekmektedir. Bu nedenle görünür noktaların yükseklik bilgisi korunarak noktalara koordinat dönüşümü uygulanmaktadır.

Dönüştürülmüş görünür nokta koordinatları Minimum Bounding Geometry analizinin Convex Hull özelliği kullanılarak çizgisel olarak birleştirilmekte ve böylelikle mevcut binaların silüet görüntüsü oluşturulmaktadır. Kullanıcı tarafından belirlenen yeni bina modelinin, durulan noktaya göre görünür noktaları belirlenmekte ve dönüşüm işlemleri uygulanmaktadır. Ardından mevcut ve yeni binaların dönüştürülmüş görünür noktalarına ait silüet görüntüsü üretilmektedir. Böylelikle yeni binanın silüet üzerindeki etkileri görsel olarak incelenebilmektedir.

Bunlara ek olarak; ArcMap ortamında çizdirilen silüet görüntüleri, bir Grafikselle Kullanıcı Arayüzü (Graphical User Interface, GUI) kullanılarak SVG formatında ayrı pencere olarak gösterimi sağlanmaktadır. GUI oluştururken Python ile uyumlu PyQt4 kütüphanelerinden QtGui ve QtSvg sınıfları kullanılmıştır.

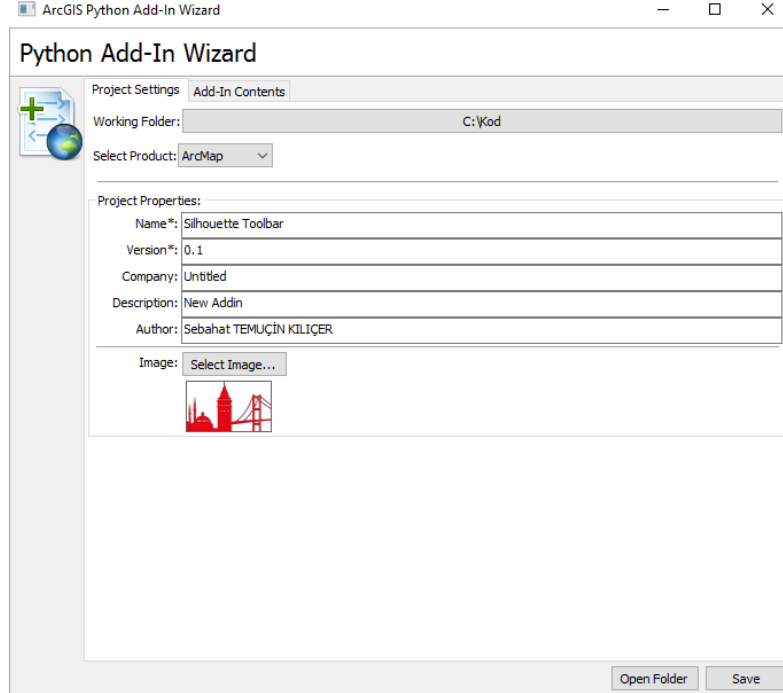
Silüet analiz modülünde son adım olarak mevcut binalar arasından silüeti korunmak istenen bir bina modeli kullanıcı tarafından seçtirilmekte ve korunan bina ile yeni bina arasında silüetin bozulup bozulmadığına dair analizler gerçekleştirilmektedir. Yeni bina, korunan binanın silüetini bozmuyorsa; bozmadığına dair bilgi mesajları üretilmekte, bozuyorsa; silüeti korunacak binanın silüetinin bozulmaması için yapılacak yeni binanın maksimum yüksekliği ve yapılabilecek maksimum kat adedi bilgileri hesaplatılmaktadır.

Böylelikle 3B bina modellerine ait silüet görüntüleri hem görsel olarak değerlendirilmekte hem de sayısal değerler ile silüet durumları yorumlanabilmektedir.

2.3. Silüet Analiz Modülü İçin Araç Çubuğunun Tasarımı

ArcGIS programı 9.0 sürümünden sonra Python programlama dili kullanılarak ek modüller tasarımı ile genişletilme imkânı sunmaktadır. Python programlama dili nesne yönelimli bir programlama dilidir. Python kullanılarak ArcGIS’de iş akışları pratikleştirilmiştir. ArcGIS programı kurulumuyla Python yazılımı da kullanıma sunulmaktadır. ArcGIS masaüstü uygulamalarıyla birlikte Python penceresi sunmaktadır. Ayrıca Model Builder kullanılarak oluşturulmuş iş akışları Python dosyası olarak kayıt edebilmekte ve Field Calculator penceresinde parser olarak cad ile birlikte Python seçeneği de sunularak tasarım açıcından kullanıcıya kullanım kolaylığı sağlamaktadır.

Bunların yanında, ArcGIS Python Add-In Wizard programı pratik olarak araç çubuğu tasarımına imkân sunmaktadır. Programın kurulumunun ardından program başlatılarak araç çubuğu tasarım çalışmalarının kayıt edileceği dosya belirlenerek araç çubuğu tasarımı yapılmaktadır.

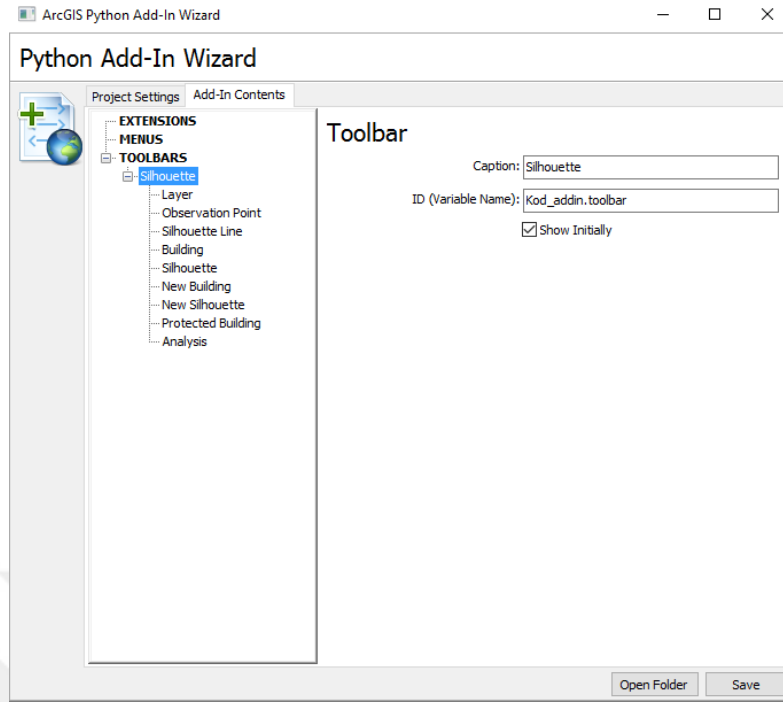


Şekil 38. Project Settings penceresi

Python Add-In Wizard programının, Şekil 38'de gösterilen Project Setting sekmesinden yeni araç çubuğu için yaratılan projeye ait bilgiler ayarlanabilmektedir. Project Setting sekmesi ile ArcGIS ürünlerinden ArcMap, ArcGlobe, ArcScene veya ArcCatalog programlarından hangisi için bir araç çubuğu tasarımı yapılacağı belirlenmektedir. Ayrıca araç çubuğunun adı, versiyonu, yazarı ve simgesi gibi bilgiler belirtilebilmektedir.

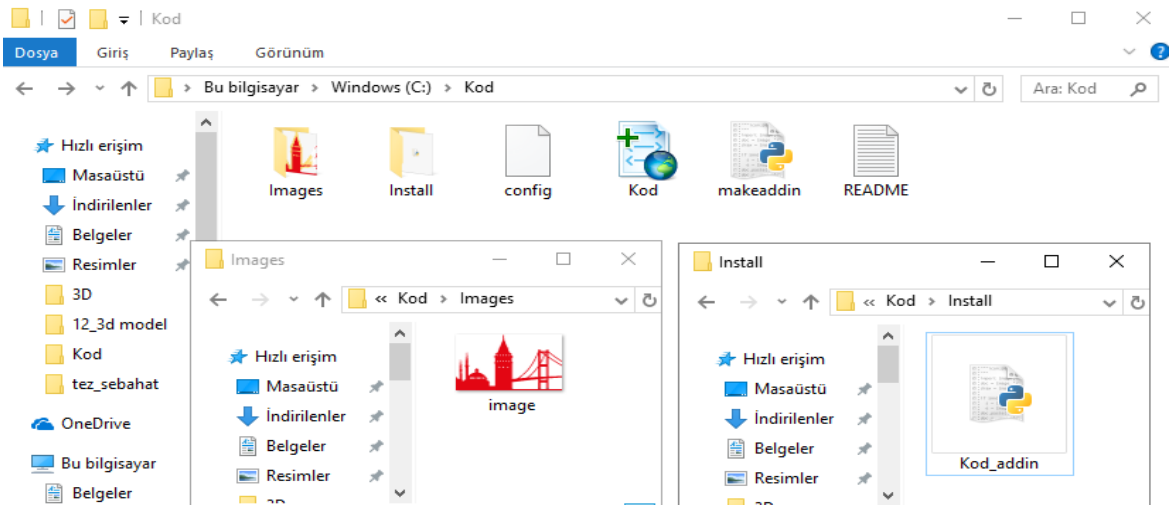
Python Add-In Wizard programının Şekil 39'da gösterilen Add-In Contents sekmesinden ise araç çubuğu tasarım içerikleri ayarlanabilmektedir. ArcGIS için Extension, Menu veya Toolbar olarak kullanılacak araç çubuğu tasarımı yapılabilmektedir. Burada Extension, Menu veya Toolbar isimleri ve bunlara ait kullanıcıya yönelik bilgi mesajları da Add-In Contents sekmesi parametrelerinden ayarlanabilmektedir. Python Add-In kullanılarak oluşturulan tipler aşağıda açıklanmaktadır;

- Button, butona tıklandığında kodlanan işlemi Full Extent gibi direk gerçekleştirmektedir.
- Tool, Button mantığında çalışmaktadır fakat düğmeye tıklandığında kullanıcının program ekranında (Data Frame) Zoom To gibi bir işlem gerçekleştirmesi gerekmektedir.
- Tool Palette, Tool mantığında çalışmakta ve birbiri ile bağlantılı Tool düğmelerinden oluşturmaktadır.
- Combo Box, açılır kutular şeklindedir ve tıklandığında içinde bulunması istenen verilerin kodlanması ile kullanıcının seçim yapacağı verileri sunmakta veya ArcMap'de bulunan Scale gibi kullanıcı bir değer yazabilmektedir.
- Extensions, programda bulunması istenilen veri kodlanması ile bu veri ekranda bulunduğu zaman ya birbirine bağlı işlemler gerçekleştirmeyi ya da Tool veya Button gibi bileşenlerin aktifleşmesini sağlamaktadır. Örneğin ArcMap ekranında bir SYM bulunduğu zaman 3D Analyst araç çubuğundaki 3D Analyst Layer'ın SYM'yi direk eklemesi Extension kullanılarak gerçekleşmektedir.
- Menu, buton mantığında çalışmakta ve açılır kutular ile bağlantılı düğmeler oluşturabilmektedir.
- Toolbar, içinde Button, Tool, Combo Box, Tool Palette veya Menu tiplerinden biri veya birkaçını içeren araç çubuklarıdır.



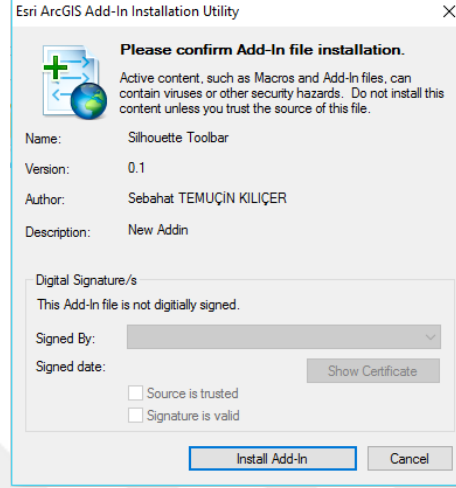
Şekil 39. Add-In Contents penceresi

ArcGIS programları ile ilişkilendirilecek olan araç çubuğu tasarımının tamamlanması ile çalışmaların kayıt edileceği ‘Kod’ isimli klasöre Şekil 40’daki gibi bazı klasör ve dosyalar otomatik olarak eklenmektedir. Burada Image klasöründe tasarım için yüklenen simge dosyaları, Install klasöründe ise kodların yazılacağı Python dosyası bulunmaktadır.



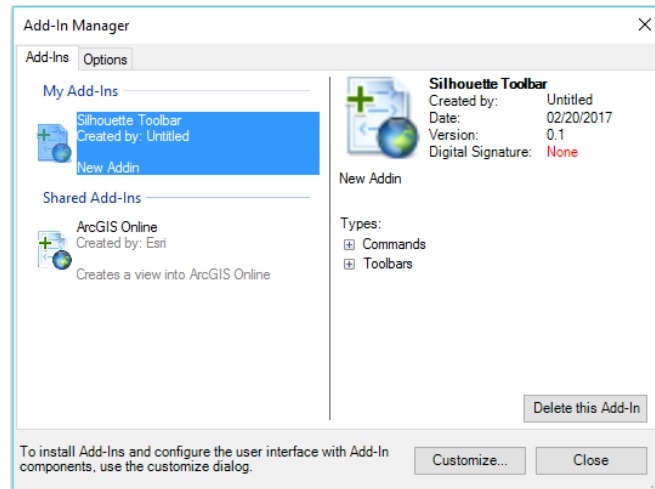
Şekil 40. Kod klasörü görünümü

Araç çubuğu tasarımının tamamlanmasından sonra Şekil 40'daki Esri Add In File uzantılı "Kod" dosyasına çift tıklanarak Şekil 41'de açılan pencereden Install Add-In kullanılması ile tasarlanan araç çubuğu ArcGIS programına entegre edilmektedir.



Şekil 41. Esri ArcGIS Add-In Installation Utility penceresi

Tasarlanmış kullanıcı araç çubuklarının ArcGIS programında yönetimi Add-In Manager ile sağlanabilmektedir. Şekil 42'de gösterilen Add-In Manager penceresi ile kaldırılmak istenen araç çubukları programdan silinebilmekte veya ArcGIS ile ilişkilendirilen araç çubukları programda gösterilmektedir.



Şekil 42. Add-In Manager penceresi

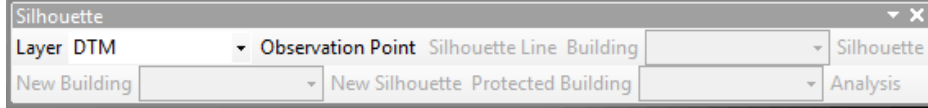
Bu çalışmada, silüet analizi modülü için Şekil 43’de gösterilen “Silhouette” isimli araç çubuğu tasarlanarak ArcMap programında kullanıma sunulmuştur.



Şekil 43. Silhouette araç çubuğu

Silhouette araç çubuğunun tasarımına ait bazı detaylar aşağıda ifade edilmektedir;

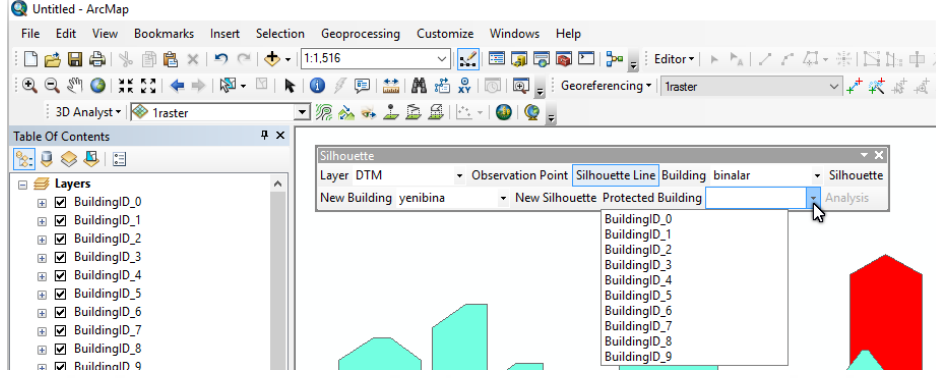
- Araç çubuğunda yer alan butonlar, silüet analizindeki işlem adımlarının akışına göre aktif hale gelecek şekilde kodlanmıştır. Örneğin, Şekil 44’de gösterildiği gibi Layer isimli açılır kutudan raster formatta bir yüzey modeli seçildiği anda bu açılır kutuya ait işlemler tamamlanmakta ve silüet analizi için gözlem yapılacak durulan noktanın belirlenmesine olanak sağlayan Observation Point isimli buton aktif hale gelmektedir. Durulan nokta seçildikten sonra işlem sırasına göre sonraki butonlar aktifleştirilmektedir.



Şekil 44. Silhouette araç çubuğunda butonların aktif ve pasif görünümü

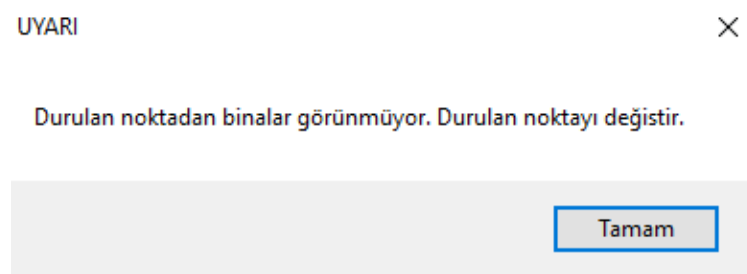
- Araç çubuğundaki Layer isimli açılır kutu sadece raster formattaki sayısal yükseklik modelini, Building ve New Building isimli açılır kutular sadece multipatch veri yapısındaki binaları, Protected Building isimli açılır kutu ise silüet analizine giren multipatch veri yapısındaki binaları listeleyebilecek şekilde kodlanmıştır. Örneğin, Şekil 45’de gösterildiği gibi Table of Context kısmında multipatch formatında birçok bina bulunurken, Protected Building sadece silüete giren multipatch binalardan bir seçim yaptırmaktadır. Ayrıca normalde açılır kutular belirtilen veri yapılarını görüntüleyebilmekte iken, kodlanan belirli

satırlar ile uygulamaya yeni veriler eklediği anda açılır kutular eklenen yeni veriyi de gösterebilmektedir.



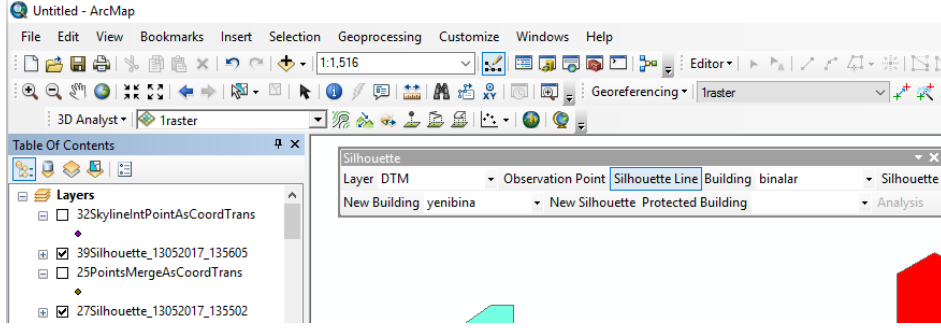
Şekil 45. Protected Building açılır kutusunda multipatch binaların gösterimi

- Silüet analiz modülünde analiz işlemleri bazı özel durumlarla karşılaşıldığında işlemi durdurarak dikkat edilmesi veya yapılması önem arz eden bilgi veya uyarıları mesaj kutusu olarak ekranda göstererek kullanıcının yapacağı düzeltmelerle işlemi kaldığı yerden devam ettirebilmektedir. Örneğin, durulan noktadan silüeti üretilecek binalar görünmesi mümkün değilse Şekil 46'da gösterildiği gibi durulan noktanın değiştirilmesi istenen bir mesaj vermektedir. Kullanıcının mesajı kapatması ile işlemleri en başa almasına gerek kalmadan sadece durulan noktayı değiştirerek işlemi devam ettirebilmektedir.



Şekil 46. Durulan noktanın değiştirilmesine dair uyarı mesajı

- Analiz sonucunda üretilen silüet görüntüleri, Şekil 47’de gösterildiği gibi dosya adına tarih ve saat bilgileri eklenerek kayıt edilmektedir.



Şekil 47. Silüet görüntüsünün üretilen tarih ve saatinin gösterimi

- Analiz sonucunda üretilen silüet görüntüleri hem ESRI shape formatında hem de ayrı bir pencerede SVG formatında gösterebilmektedir. Örneğin, New Silhouette butonu ile üretilen SVG formatındaki silüet görüntüsü Şekil 48’de gösterilmektedir.

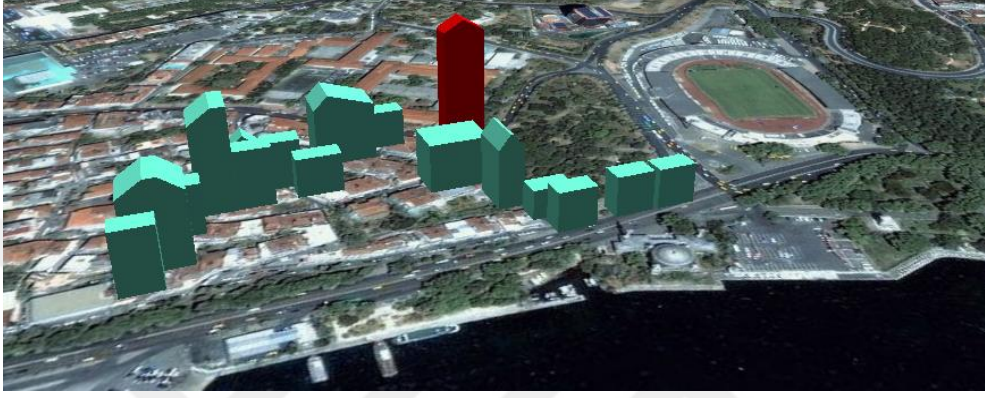


Şekil 48. Silüet görüntüsü penceresi

2.4. Uygulama

Bu çalışmada, silüet analizi çalışmaları yapılacak bölgeye ait 3B bina modellerini üretebilmek adına sayısal yüzey modeli ve CityGML standartlarına göre Lod 1 seviyesinde

blok modeller olarak farklı yapılarda 17 ayrı bina modeli üretilmiştir. CityEngine programı 2015.2 versiyonu kullanılarak oluşturulan 3B bina modellerinin ArcGIS programı 10.2 versiyonundaki görünümü Şekil 47’de gösterilmektedir. Üretilen 3B bina modellerinden silüet analizinde yeni bina olarak bahsi geçen ve silüet durumu değerlendirilecek olan bina kırmızı renkte gösterilmektedir.

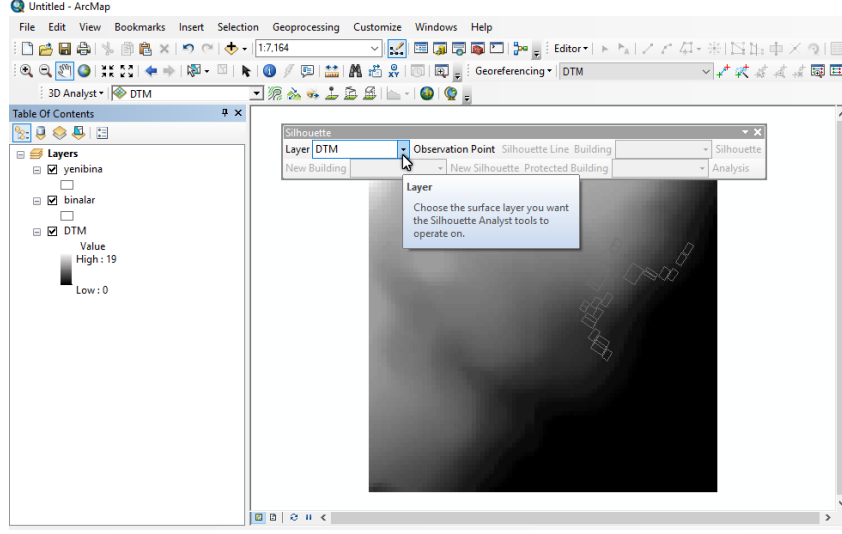


Şekil 49. 3B bina modelleri

Bu uygulama ile bir durulan nokta ve silüet hattı belirlenerek mevcut binaların silüet görüntüsü ve yeni yapılacak binanın mevcut binalarla olan silüet görüntüsü üretilmektedir. Ayrıca mevcut binalardan silüetinin korunması önem arz eden bina katmanı seçtirilmekte ve silüeti korunmak istenen bina ile yeni yapılacak olan bina arasında bir silüet analizi gerçekleştirmektedir. Bu analiz ile yeni binanın silüeti korunması istenilen binanın silüetinin bozup bozmadığı değerlendirilmektedir.

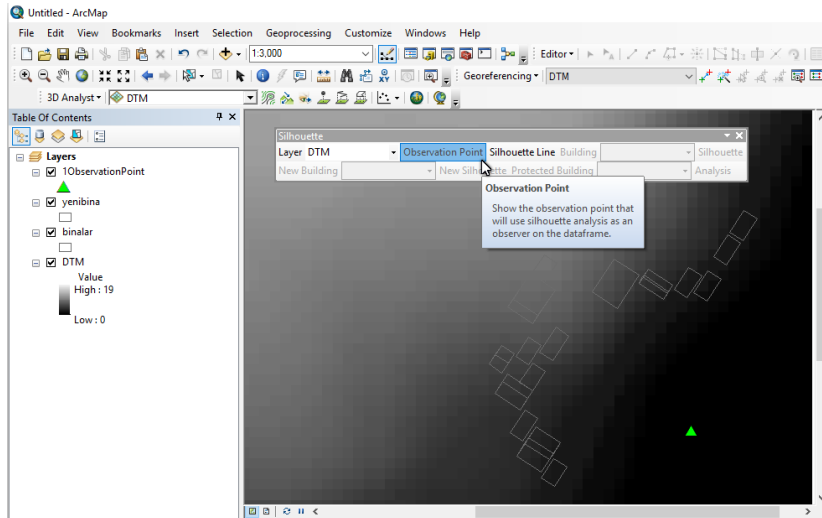
Yeni binanın korunan bina silüetini bozmaması durumunda, yeni binanın korunan bina silüetini bozmadığına dair rapor sunulmaktadır. Yeni binanın korunan bina silüetini bozması durumunda ise yeni bina ve korunan bina arasında analiz gerçekleştirilmektedir. Böylelikle yeni binanın korunan bina silüetini bozmaması için olması gereken maksimum bina yüksekliği ve yapılabilecek maksimum bina adeti değerleri hesaplatılarak bu değerler analiz raporu olarak sunulmaktadır.

1. Layer isimli açılır kutu kullanılarak çalışma alanına ait raster formatında bir yüzey modeli tanımlanmaktadır. Kullanılan yüzey modeli ilerleyen adımlarda bulunan analizler için kullanılacak arazi yüzeyini temsil etmesi gerekmektedir. Yüzey modelinin seçtirilmesi Şekil 50’ de gösterilmektedir.



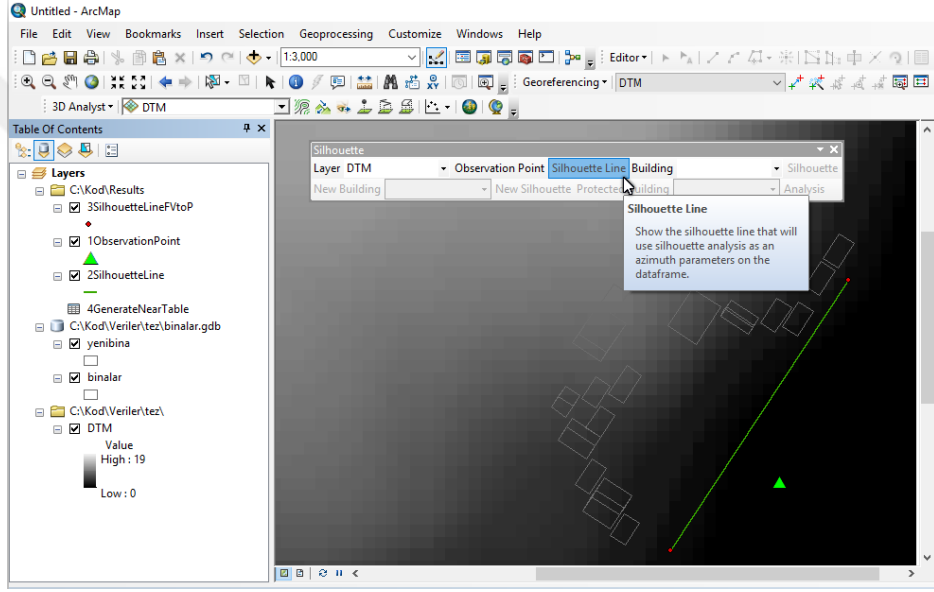
Şekil 50. Yüzey modelinin seçtirilmesi

2. Observation Point isimli buton kullanılarak gözlem yapılacak yer olan durulan nokta belirlenmektedir. Durulan nokta, kullanıcı tarafından ekrandan seçilme olanağı sunarak otomatik olarak üretilmektedir. Durulan nokta, yükseklik bilgisini yüzey modelinden okuyarak bu yüksekliğe ortalama kişi yüksekliği olan 1.75 m eklenecek şekilde üretilmektedir. Seçilen durulan noktadan gözlem yapılacak alanın görünürlüğü değerlendirilecektir. Durulan noktanın belirlenmesi Şekil 51’ de gösterilmektedir.



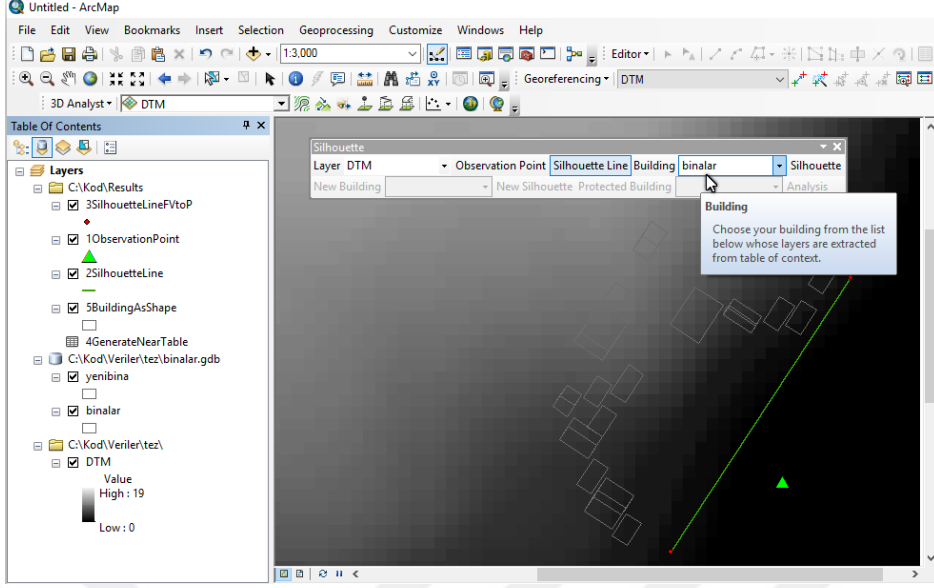
Şekil 51. Durulan noktanın belirlenmesi

3. Silhouette Line isimli araç (tool) kullanılarak gözlem yapılacak görüş alanındaki silüet hattı belirlenmektedir. Silüet hattı, durulan noktanın üretilmesi gibi kullanıcı tarafından ekrandan hattın başlangıç ve bitiş noktası seçilmesinin ardından otomatik olarak üretilmektedir. Silüet hattı kullanılarak durulan nokta ve silüet hattının başlangıç ve bitiş noktaları arasında semt açısı hesapları yaptırılarak skyline analizinde kullanılacak olan From Azimuth ve To Azimuth parametreleri otomatik olarak hesaplatılmaktadır. Silüet hattının belirlenmesi Şekil 52’ de gösterilmektedir.



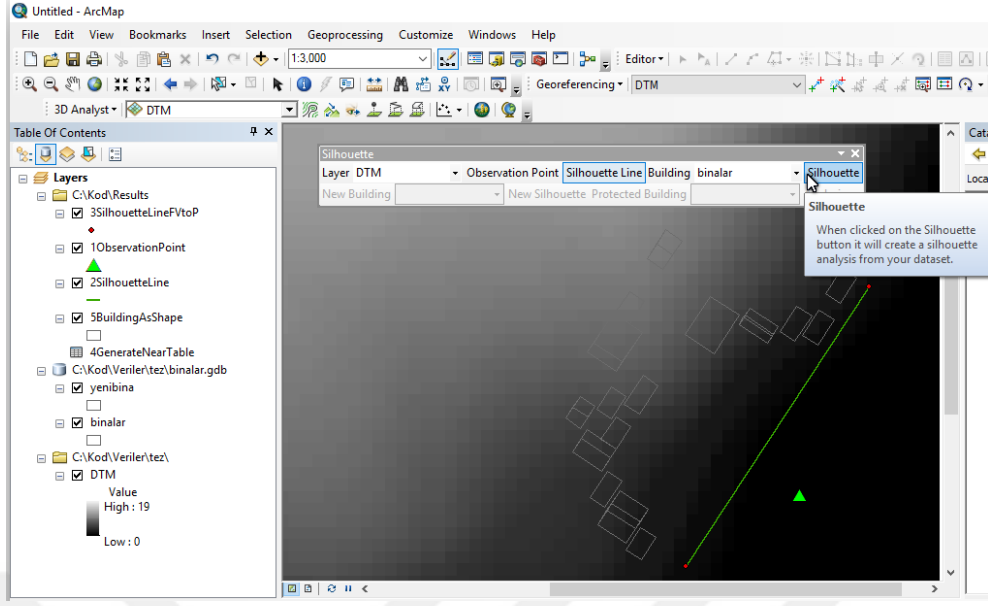
Şekil 52. Silüet hattının belirlenmesi

4. Building isimli açılır kutu kullanılarak belirlenen durulan noktadan ve belirlenen silüet hattından silüeti çıkartılacak olan multipatch formatındaki binalar katmanı seçilmektedir. Binalar katmanının seçtirilmesi Şekil 53’ de gösterilmektedir.

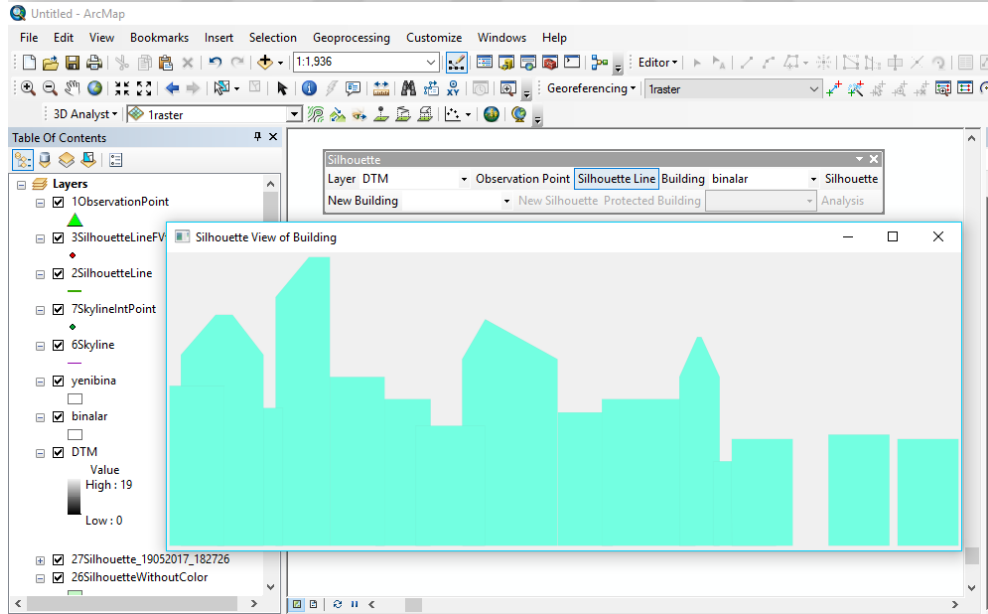


Şekil 53. Binalar katmanının seçtirilmesi

5. Silhouette isimli buton kullanılarak belirli bir durulan noktadan belirli bir görüş alanındaki mevcut binaların silüet görüntüsü üretilerek ekranda gösterimi sağlanmaktadır. Silüet butonu ile silüet analizine giren binalar belirlenmekte ve görünürlüğü olan bina noktaları tespit edilmektedir. Ardından 3B görünür noktaların 2B düzlemde temsili için koordinat dönüşüm parametreleri hesaplanmaktadır. Böylelikle görünen noktalara koordinat dönüşümü uygulanarak dönüşmüş koordinatlar ile silüeti temsil eden katman çizdirilmekte ve çizdirilen silüet katmanında binalara ait ID değerleri gibi öznitelik bilgilerine ulaşılabilmektedir. Çizdirilen görüntü de binalar durulan noktaya olan mesafesi belirlenerek önde olan binanın görüntüde de önde çizdirilmesi sağlanmaktadır. Böylelikle iç içe görüntü geçmesiyle bina katmanlarının görüntüde kaybolması önlenmektedir. Ayrıca üretilen silüet katmanın SVG formatına dönüşümü sağlanarak silüet görüntüsü olarak ekranda gösterimi sağlanmaktadır. Silüet üretiminin başlatılması Şekil 54' de gösterilmektedir. Binalar katmanına ait silüet görüntüsü ise Şekil 55' de verilmektedir.

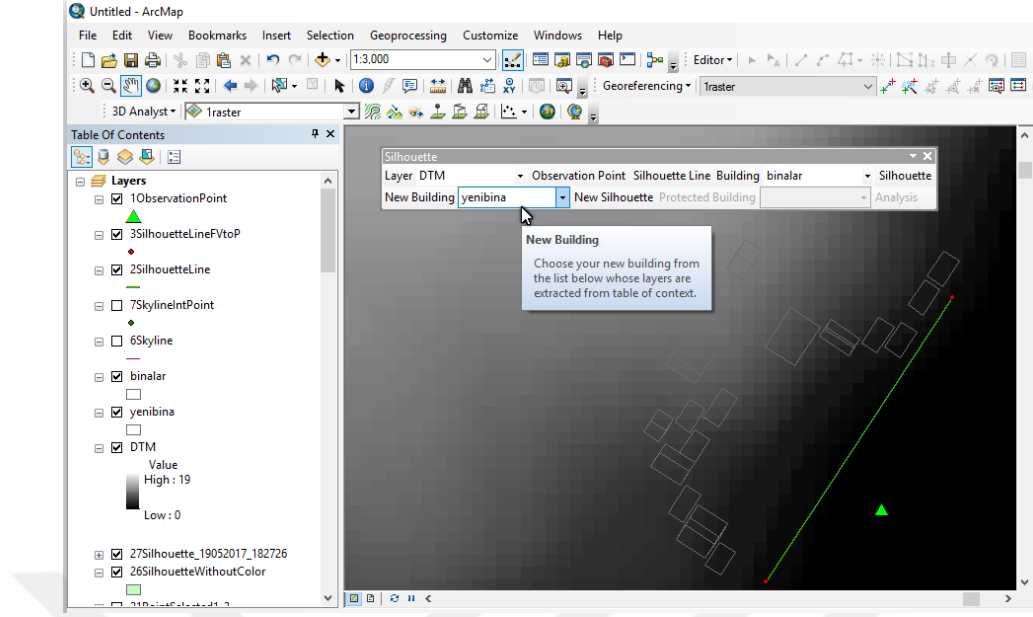


Şekil 54. Silüet üretiminin başlatılması



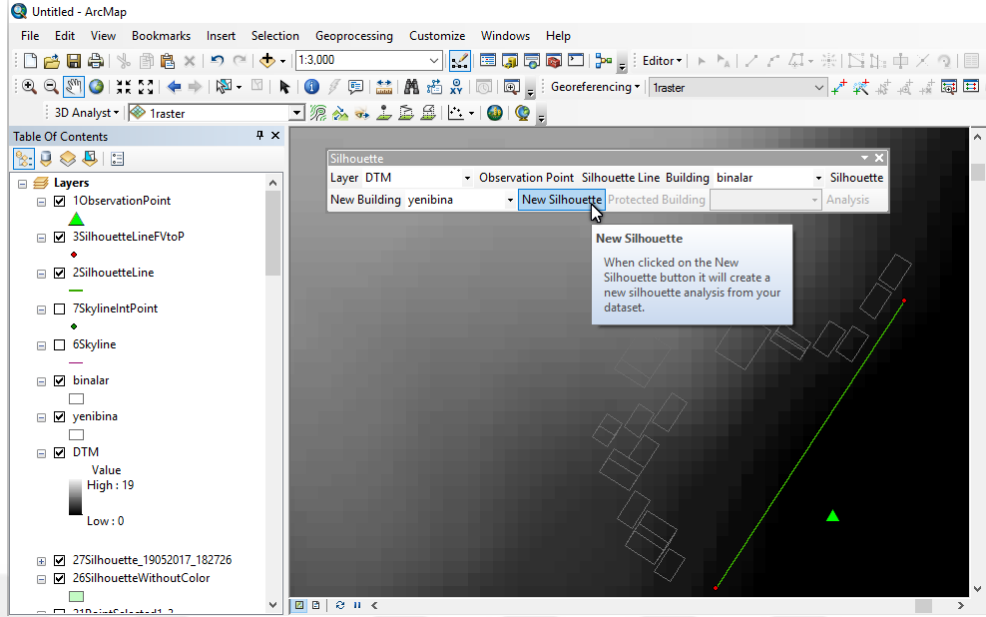
Şekil 55. Binalar katmanına ait silüet görüntüsü

6. New Building isimli açılır kutu kullanılarak kentsel alana yeni yapılacak olan multipatch formatında bina katmanı seçilmektedir. Yeni bina katmanının seçtirilmesi Şekil 56' da gösterilmektedir.

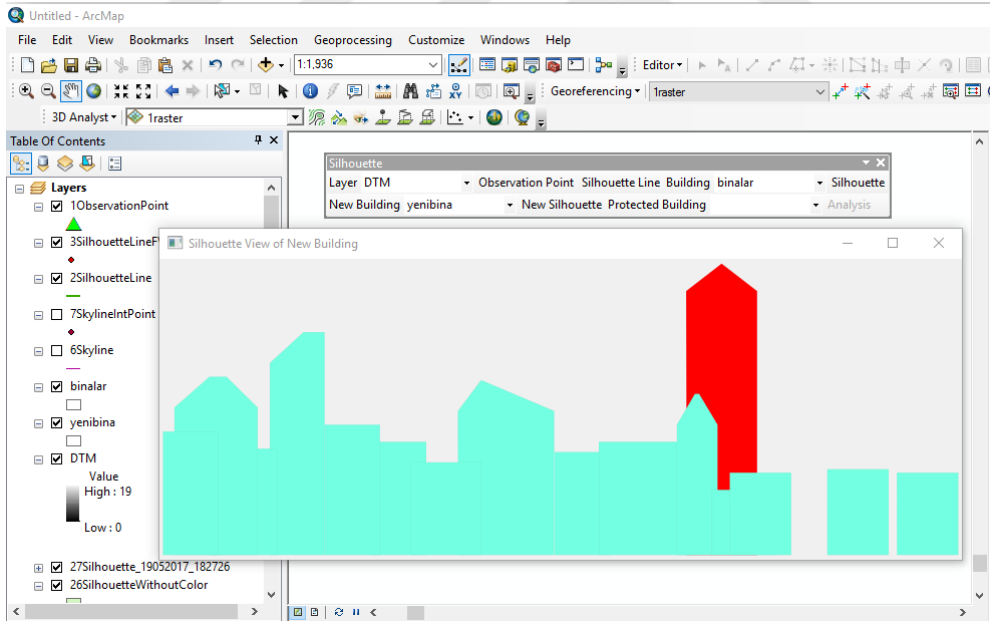


Şekil 56. Yeni bina katmanının seçtirilmesi

7. NewSilhouette isimli buton kullanılarak silüet görüntüsü üretilmiş alanın yeni yapılacak bina ile yeni durumunun silüet görüntüsü üretilerek ekranda gösterimi sağlanmaktadır. Yeni silüet butonu ile yeni binaya ait görünürlüğü olan bina noktaları tespit edilmektedir. Silüet butonunda hesaplanan koordinat dönüşüm parametreleri kullanılarak yeni binanın görünen noktalarına koordinat dönüşümü uygulanmaktadır. Yeni binanın dönüşmüş koordinatları ile mevcut binaların dönüşmüş koordinatları birleştirilerek yeni ve mevcut binalara ait silüeti temsil eden katman çizdirilmektedir. Burada işlem tekrarını önlemek adına mevcut binaların dönüşmüş koordinatları tekrar üretilmemektedir. Silüet butonunda olduğu gibi burada da çizdirilen silüet katmanında binalara ait öznitelik bilgilerine ulaşılabilmekte ve bu üretilen silüet katmanının SVG formatına dönüşümü sağlanarak yeni silüet görüntüsü olarak ekranda gösterimi sağlanmaktadır. Silüet butonundaki gibi çizdirilen görüntüde yeni binanın durulan noktaya olan mesafesi de belirlenmekte ve yeni binanın diğer binalarla olan konumu irdelenmektedir. Ayrıca burada yeni bina ekranda diğer binalardan farklı olarak kırmızı renkte görüntülenmektedir. Yeni silüet üretiminin başlatılması Şekil 57' de gösterilmektedir. Yeni ve mevcut binalar katmanına ait silüet görüntüsü ise Şekil 58' de verilmektedir.



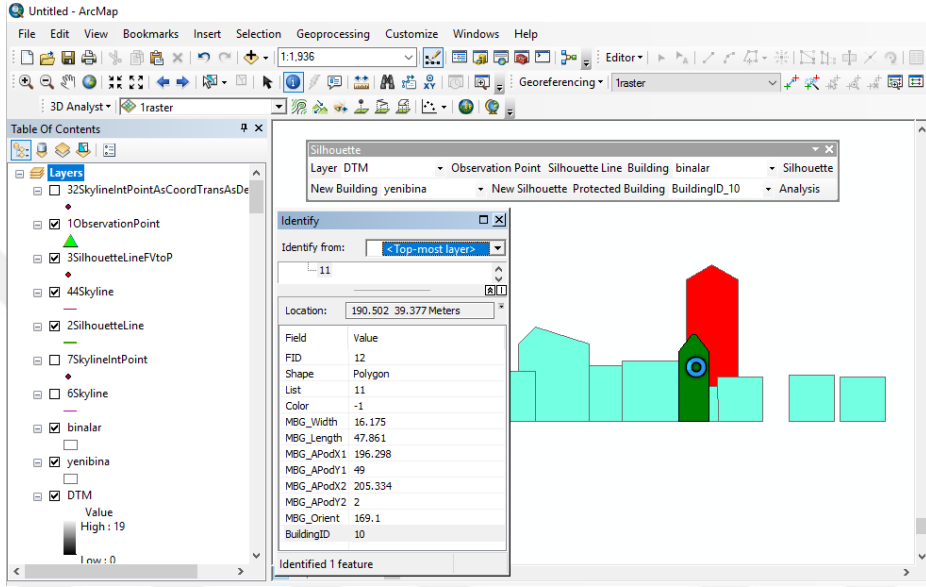
Şekil 57. Yeni silüet üretiminin başlatılması



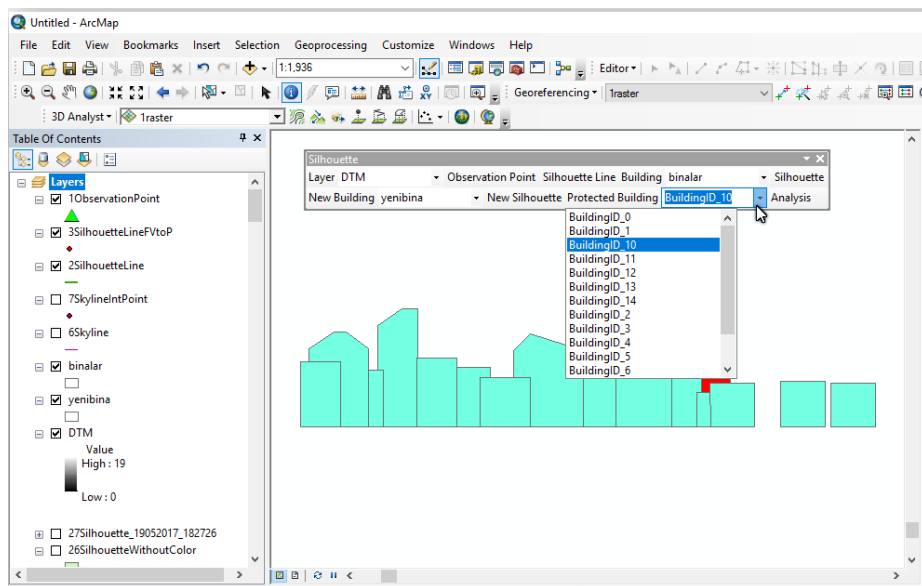
Şekil 58. Yeni ve mevcut binalar katmanına ait silüet görüntüsü

8. Protected Building isimli açılır kutu kullanılarak silüet analizine giren mevcut binalardan silüeti korunmak istenen multipatch formatında bina katmanı seçilmektedir. Burada korunan binalar olarak sadece silüete giren bina katmanları gösterilecektir. Silüet görüntüsünde sorgu özelliği kullanılarak silüeti korunmak

istenilen bina katmanına ait ID bilgisine ulaşılabilmektedir. Eğer korunan bina burada yer almıyorsa, korunması istenilen bina bu noktadan görünmediği veya silüet hattı görüş alanında olmadığı anlaşılmaktadır. Korunan bina katmanının sorgulanması Şekil 59’ da gösterilmektedir. Korunan bina katmanının seçtirilmesi ise Şekil 60’ da verilmektedir.

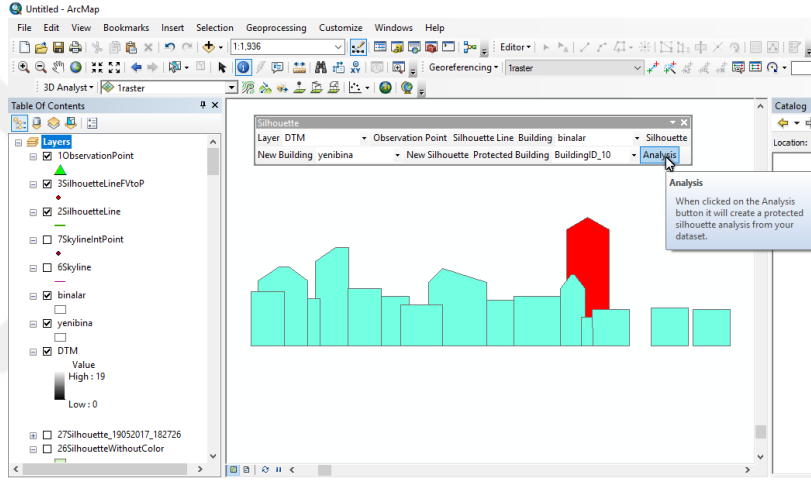


Şekil 59. BuildingID 10 için korunan bina katmanının sorgulanması

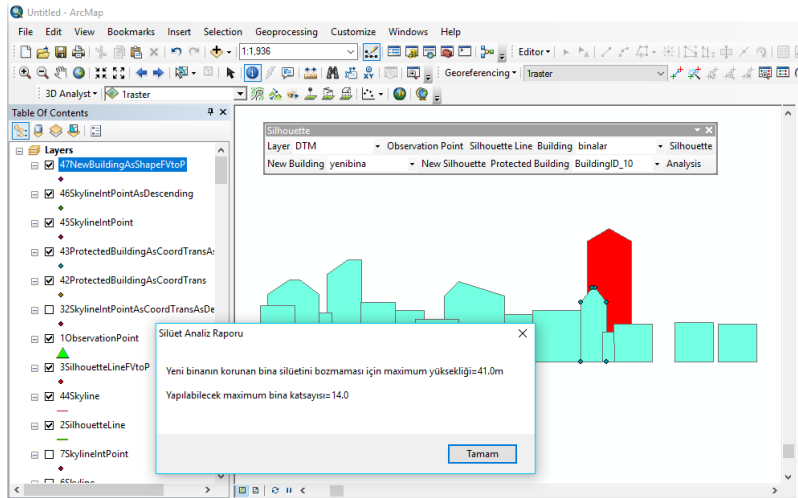


Şekil 60. BuildingID 10 için korunan bina katmanının seçtirilmesi

9. Analysis isimli buton kullanılarak yeni yapılacak bina ile silüeti korunması istenen bina arasında silüet analizi yaparak analiz raporu üretilmektedir. Silüet analizinin başlatılmasıyla silüeti korunmak istenen bina ile yeni yapılacak olan bina arasında bir silüet analizi gerçekleştirilmekte ve silüetin bozulup bozulmadığı değerlendirilmektedir. Silüetin bozulması durumunda, gerçekleştirilen işlemlerle yeni binanın korunan bina silüetini bozmaması için olması gereken maksimum bina yüksekliği ve yapılabilecek maksimum bina kat adedi değerleri hesaplanmaktadır. Silüet analizinin başlatılması Şekil 61’ de gösterilmektedir. Silüet analiz raporu ise Şekil 62’de verilmektedir.

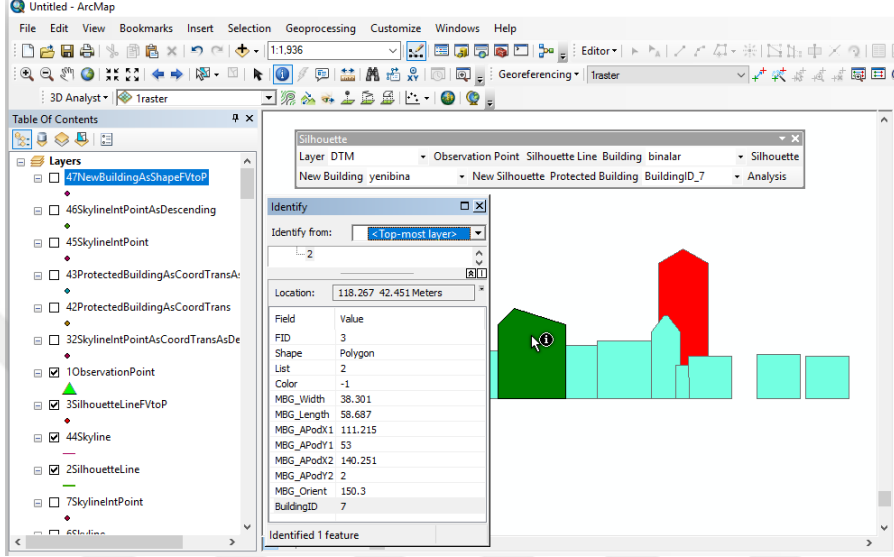


Şekil 61. Silüet analizinin başlatılması

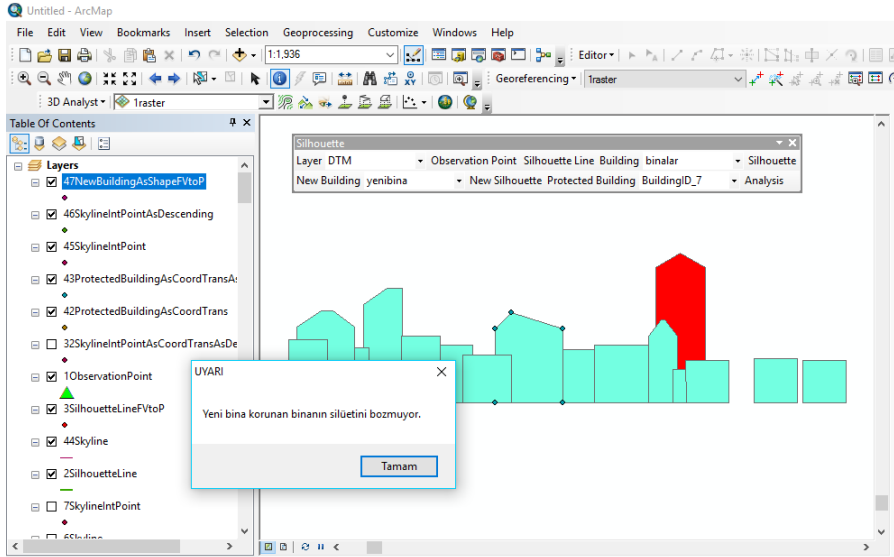


Şekil 62. Silüet analiz raporu

10. Yeni binanın korunan bina silüetini bozmaması durumunda, silüetin bozulmadığına dair rapor sunulmaktadır. Seçilen farklı bir korunan bina katmanının sorgulanması Şekil 63’ de gösterilmektedir. Silüeti bozulmadığına dair silüet analiz raporu ise Şekil 64’ de verilmektedir.



Şekil 63. BuildingID 7 için korunan bina katmanının sorgulanması



Şekil 64. BuildingID 7 için silüet analiz raporu

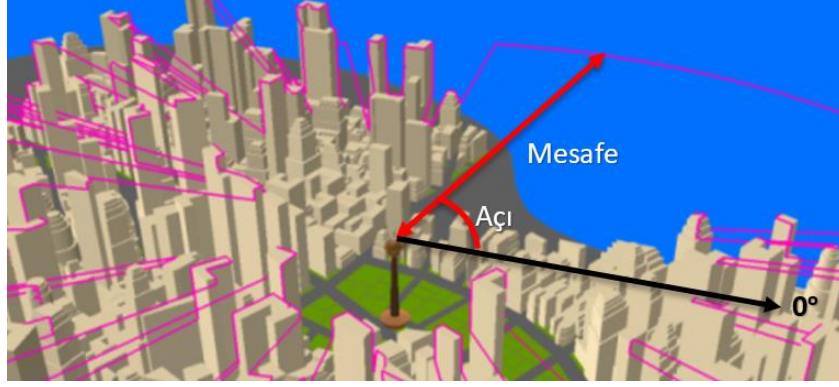
3. BULGULAR VE TARTIŞMA

Bu tez çalışmasında, gözlem yapılan noktadan belirli bir görüş alanında kalan binaların silüet görüntüsünü üreten ve aynı alanda yapılması düşünülen yeni binaların silüete etkilerini tespit eden yeni bir silüet analiz modülü geliştirilmiştir. Böylelikle kentsel planlama sürecinde ve imar planlarının uygulanmasında karşılaşılan kentsel silüetin bozulmasından kaynaklanan problemlerin çözülmesi amaçlanmıştır.

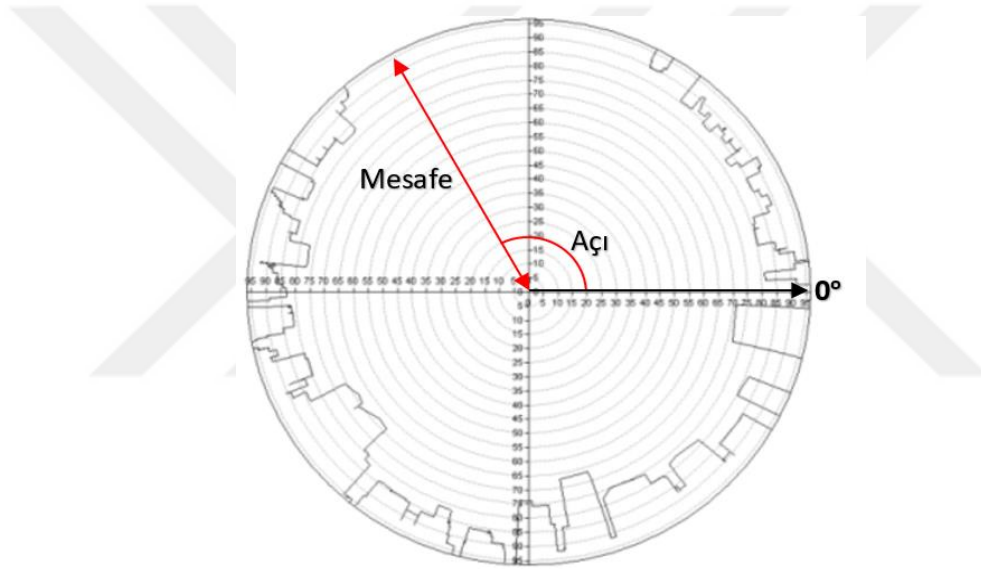
Silüet analiz modülü, ArcGIS CBS yazılımında çalışacak şekilde Phyton programlama dili kullanılarak geliştirilmiştir. Çalışmada kullanılan 3B binalar, CityGML standartlarına göre LoD 1 detay seviyesinde çalışılacak şekilde modellenmiştir. Çünkü LoD seviyesi arttıkça bina modellerinin içerdiği çatı yapıları, bina dokuları, duvar detayları ve balkonlar gibi mimari detaylar da artacaktır. Dolayısıyla multipatch geometri tipindeki detay noktaları da artacaktır. Böylelikle multipatch objeyi temsil eden yüzey parçalarındaki nokta koordinatlarının tekrar etmesi hem işlem adımlarında kullanılacak noktayı belirleme aşamasında sorunlar çıkaracak hem de performans açısından geliştirilen modülü yavaşlatacaktır.

Geliştirilen silüet analiz modülünün ArcGIS tarafından sunulan görünürlük analizi fonksiyonlarından ve bu fonksiyonlar kullanılarak literatürde yapılan çalışmalardan farkı aşağıdaki bölümlerde açıklanmıştır.

ArcGIS yazılımına ait görünürlük analizi fonksiyonları kullanılarak, 3B kentsel alanlarda çeşitli görünürlük analizleri yapılabilmektedir. Ancak, sunulan fonksiyonlar ile kentsel alanlardaki silüet durumunun analizi, hem çok sayıda işlem adımı ve parametre gerektirmekte hem de üretilen sonuç görsel olarak yorumlanmaktadır. Gözlem yapılan noktadan 3B kent modeline ait görünür alanlara, Şekil 65’de de gösterildiği gibi skyline analizi ile ulaşılabilmektedir. Görünürlüğün grafiksel olarak yorumlanabilmesi için ise Şekil 66’da verilen skyline grafikleri kullanılmaktadır.

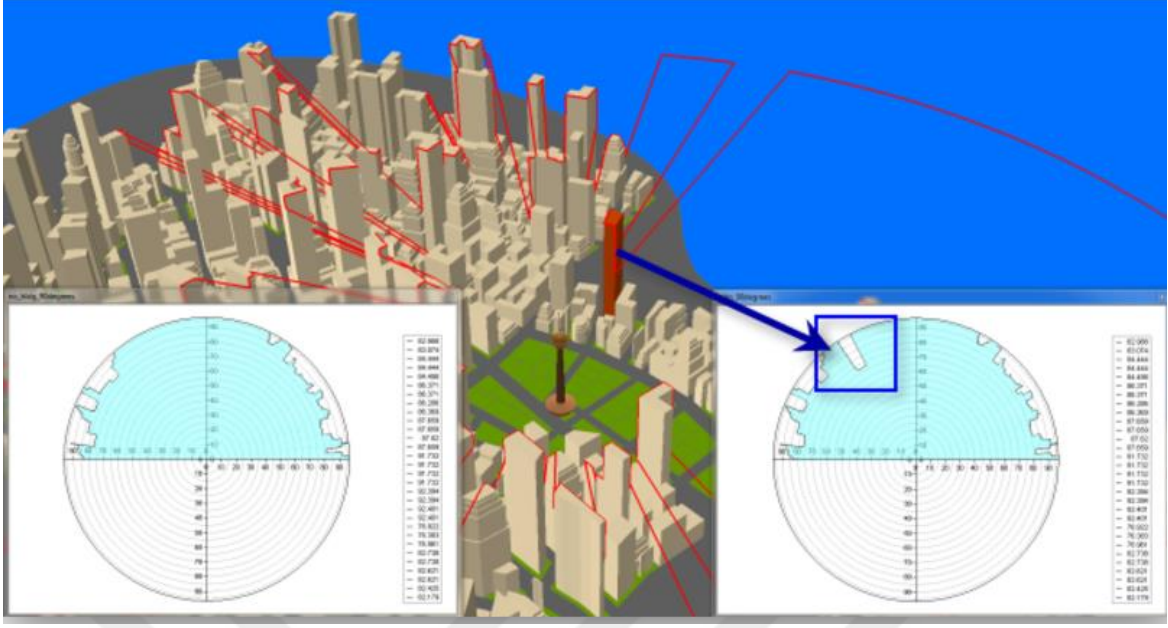


Şekil 65. Skyline analizi üzerinde açı ve mesafelerin gösterimi



Şekil 66. Skyline grafik analizi üzerinde açı ve mesafelerin gösterimi

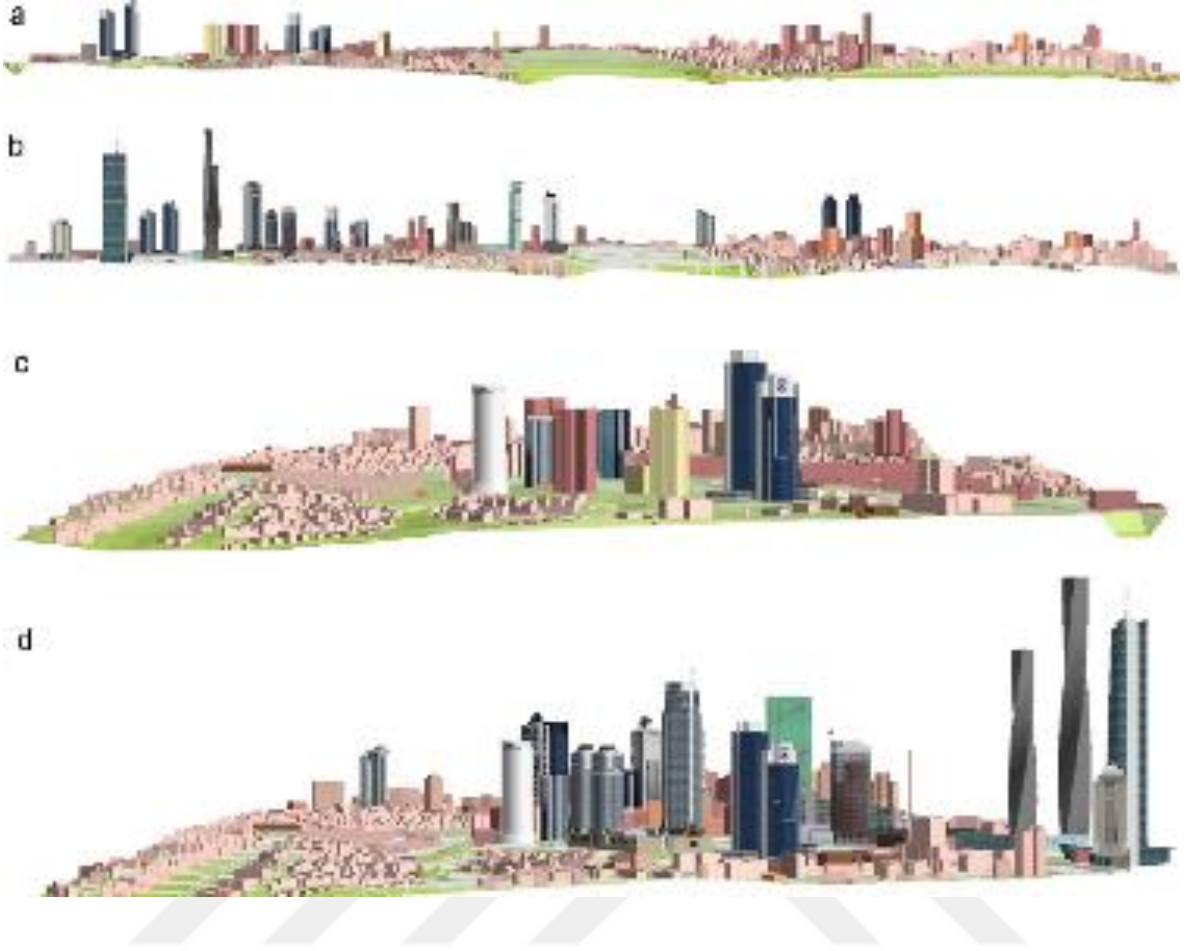
ArcGIS yazılımı tarafından sunulan görünürlük analizi fonksiyonları ile sadece silüete benzer grafik görüntüleri üretilebilmektedir. Grafiklerin görsel olarak yorumlanması ile silüetin bozulup bozulmadığı tespit edilebilmektedir (Şekil 67). Bu grafiklerden gözlem yapılan noktadan görünen binaların yükseklik veya kat adedi gibi bilgilere ulaşılamamaktadır. Üretilen grafiklerden, sadece gözlem yapılan noktadan ne kadar açı ve mesafede görünür objelerin bulunduğu tespit edilebilmektedir.



Şekil 67. Skyline grafikleri üzerinden silüet durumunun yorumlanması (URL-14, 2016)

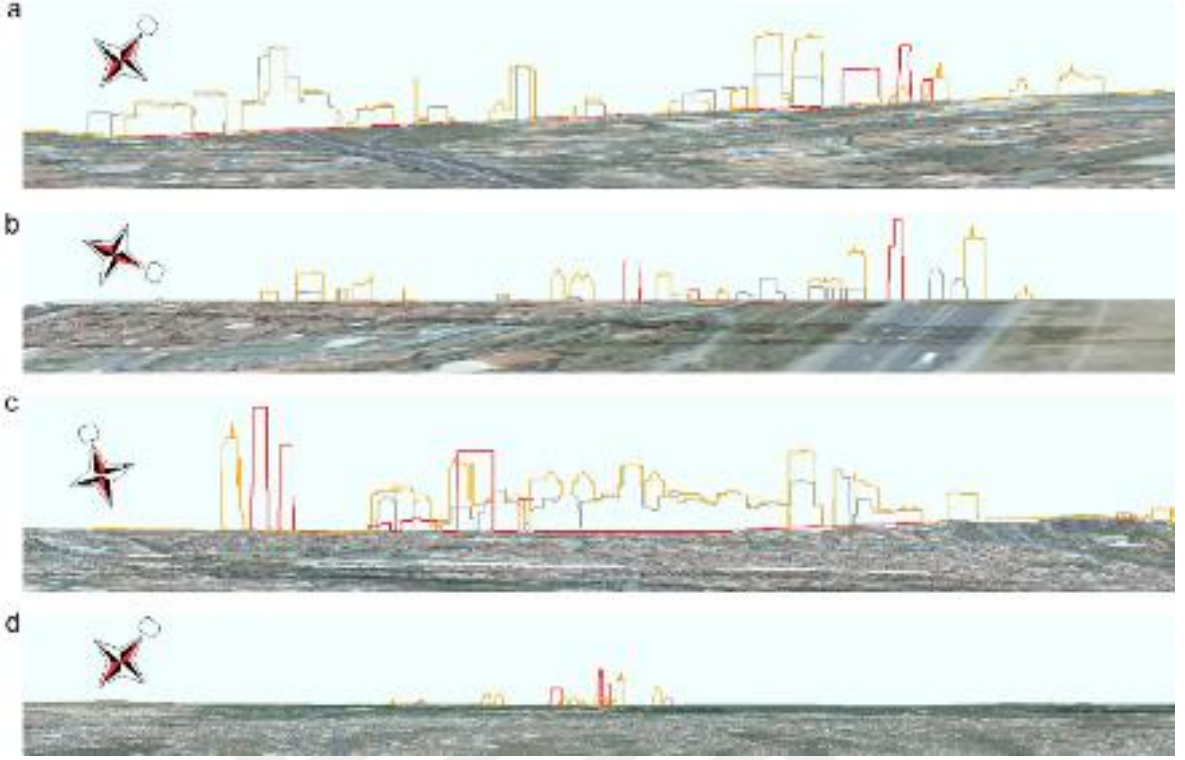
Buradan da anlaşılacağı gibi yeni yapılacak binanın, gözlem yapılan noktadan silüeti bozup bozmadığı görsel olarak belirlenebilmekte ve inşa edilecek yeni bina ile ilgili açı ve mesafe dışında herhangi bir sayısal değere ulaşamamaktadır. Ayrıca, silüetinin korunması isten bina ve yeni bina arasındaki silüet durumu da analiz edilememektedir. Başka bir ifadeyle, yeni yapılacak binanın silüeti bozmaması için inşa edilmesi gereken maksimum yükseklik veya kat adedi ile ilgili bilgilere ulaşamamaktadır.

Literatürde, ArcGIS yazılımının sunduğu görünürlük analizi fonksiyonları kullanılarak gerçekleştirilen bazı silüet çalışmaları bulunmaktadır. Örneğin; Güney vd. (2012) tarafından yapılan çalışmada, İstanbul'un Levent semtinin 3B kent modeli üretilmiş ve çalışma alanına ait hem görsel skyline analizi (visual skyline analysis) hem de zamansal skyline analizi (temporal skyline analysis) olarak belirtilen analizler gerçekleştirilmiştir. Çalışmada, Levent semtinin 1999 ve 2008 yıllarına ait 3B kent modelleri kullanılarak görsel skyline analizi ile silüet değişimi incelenmiştir. ArcGlobe programı ile gösterimi sağlanan bu analiz görüntüsü Şekil 68'de sunulmuştur.



Şekil 68. Görsel skyline analizi gösterimi; 1999 yılı, doğu-batı yönü (a), 2008 yılı ve sonrası, doğu-batı yönü (b), 1999 yılı, kuzey-güney yönü (c), 2008 yılı ve sonrası, kuzey-güney yönü (d) (Güney vd., 2012)

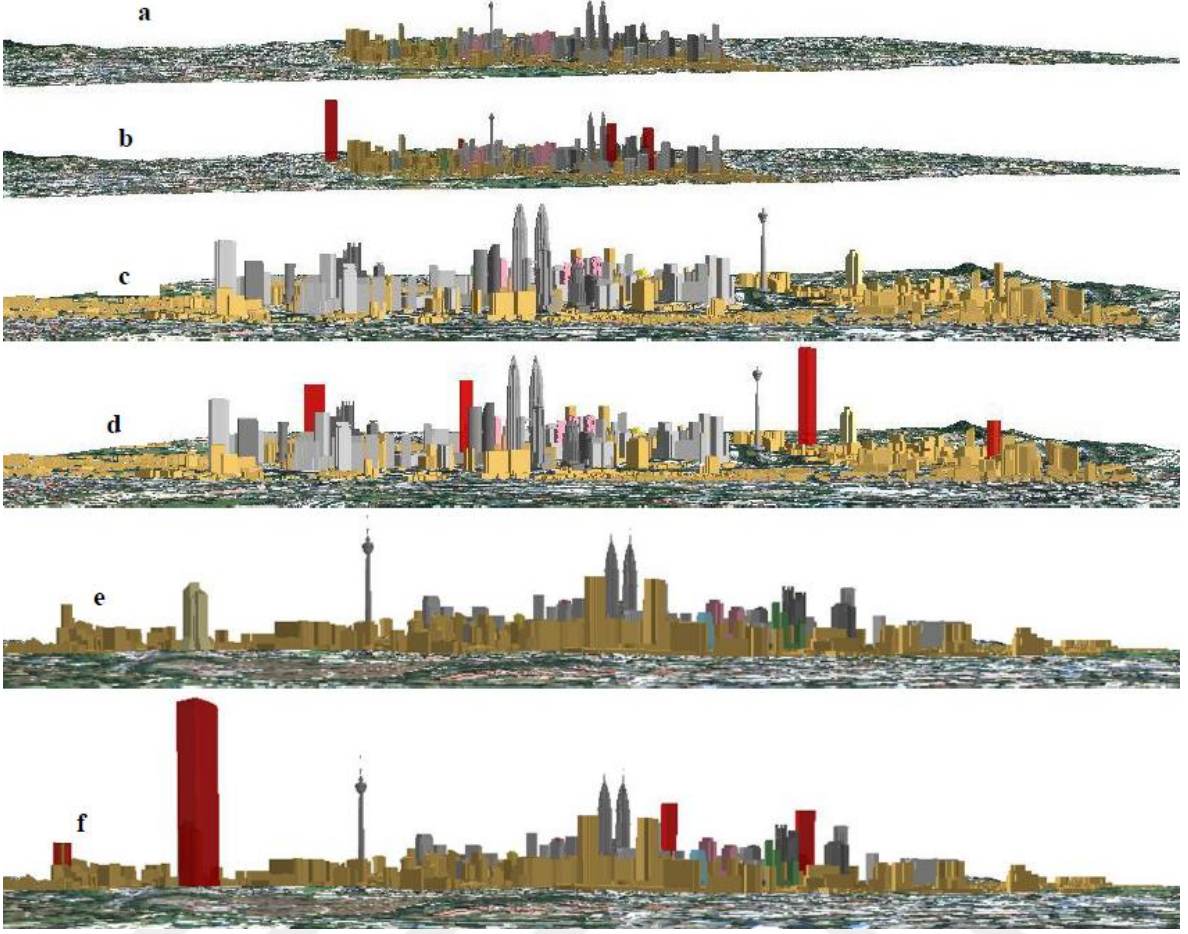
Aynı çalışmada, İstanbul'un Boğaziçi Köprüsü, Fatih Sultan Mehmet Köprüsü, Harem ve Çamlıca Tepesi gibi şehrin önemli 4 farklı noktasından, Levent semtinin zamansal skyline analizi üretilmiş ve inşa edilen yüksek yapı binaların görsel etkileri değerlendirilmiştir. ArcGlobe programı ile gösterimi sağlanan bu analiz görüntüleri ise Şekil 69'da sunulmuştur.



Şekil 69. Zamansal skyline analizi gösterimi; (a) Boğaziçi Köprüsü, (b) Fatih Sultan Mehmet Köprüsü, (c) Harem, (d) Çamlıca Tepesi (Güney vd., 2012)

Yusoff vd. (2014) tarafından yapılan bir diğer benzer çalışmada ise, ArcGIS yazılımı kullanılarak Kuala Lumpur şehrinin silüetinin korunmasına yönelik çalışmalar gerçekleştirilmiştir. Burada da 3B kent modeli üretiminin tamamlanmasının ardından ArcGlobe programı kullanılarak 3 farklı gözlem noktasından mevcut binalara ait silüet görüntüleri üretilmiştir. Ardından, yeni yapılacak 3 farklı binanın aynı gözlem noktası ve görüş açısından görüntüleri üretilmiştir. Böylelikle yeni yapılacak olan binaların kentin silüeti üzerindeki etkileri değerlendirilmiştir.

ArcGlobe programı ile gösterimi sağlanan bu analiz görüntüleri Şekil 70'de verilmektedir. Çalışma alanındaki 3 farklı gözlem noktasından (Ampang Tepesi, Malaysia Middle Ring Road 2 doğu sahili geçidi ve Sungai Besi Kuzey-Güney Karayolu) gözlem yapılarak, kırmızı renkte görünen 3 farklı yeni binanın Kuala Lumpur silüeti üzerindeki etkileri değerlendirilmiştir.



Şekil 70. Kuala Lumpur'un silüet analizleri gösterimi; 1 nolu gözlem noktası, mevcut durum (a), 1 nolu gözlem noktası, yeni durum (b), 2 nolu gözlem noktası, mevcut durum (c), 2 nolu gözlem noktası, yeni durum (d), 3 nolu gözlem noktası, mevcut durum (e), 3 nolu gözlem noktası, yeni durum (f) (Yusoff vd., 2014)

Yapılan çalışmalar ve kullanılan 3B analiz fonksiyonlarına ait sonuçlar değerlendirildiğinde, ArcGIS ortamında verilerin üretimi, görselleştirilmesi ve analizi işlemlerinin tek bir platformda toplanmış olmaması (ArcMap, CityEngine, ArcScene ve ArcGlobe) ve sunulan görünürlük analizlerinin çeşitliliği, yazılımın silüet çalışmalarında kullanılmasını dezavantajlı hale getirmektedir. Bu nedenle, silüet analizleri bağlamında yapılan çalışmalarla sadece modellerin görsel etkileri değerlendirilebilmiştir. Silüet analizi olarak bahsi geçen skyline analizleri ile sınırlı sayıda konumsal bilgilere ulaşılmaktadır. Bunların dışında her farklı gözlem noktasından farklı silüet oluşacağı için gözlem noktası değişikçe iş yükü artarak analizlerin tekrarı gerekecektir.

4. SONUÇLAR VE ÖNERİLER

3B kent modeli üretimi, veri temsili ile konumsal sorgulama ve analizler, CBS'deki aktif araştırma alanlarından biridir. Bu tez çalışmasında da 3B bina modelleri kullanılarak görünürlük analizlerinden biri olan silüet analizleri incelenmiştir.

Gerek kentsel alanların planlama sürecinde gerekse imar planlarının uygulanması aşamasında yeni inşa edilen binalara ruhsat verilmesi, kentsel alanların silüet görüntülerinin üretilmesi ve kent silüetindeki değişimin izlenmesine ihtiyaç duyulmaktadır. Günümüzde kullanılan mevcut CBS yazılımları, silüet analizi için doğrudan fonksiyonlar sunmamakta, fakat sahip oldukları görünürlük analizi fonksiyonları ile silüet benzeri grafiklerin üretilmesine olanak sağlamaktadır. Ayrıca, yeni yapılacak olan binaların mevcut binalar ile korunması istenen binaların silüetini nasıl etkilediğinin belirlenmesinde ve silüetin korunması için yapılması düşünülen yeni binaların maksimum yüksekliklerinin ve kat adetlerinin belirlenmesinde mevcut CBS yazılımlarının sunduğu fonksiyonlar yetersiz kalmaktadır.

Bu çalışmada, 3B bina modelleri üzerinde silüet analizleri gerçekleştirebilecek bir silüet analiz modülü geliştirilmiştir. Geliştirilen modül, kullanıcının ekrandan belirleyebileceği gözlem yapılan nokta olan bir durulan nokta ve görüş hattındaki 3B binaların silüet görüntüsünü üretmekte ve yeni yapılacak binaların silüeti bozmaması için sahip olması gereken maksimum yüksekliği ve kat adetini hesaplamaktadır. Silüet analizi modülü, özellikle belediyelerin İmar Müdürlüklerinde görevli kullanıcıların, çok fazla teknik bilgiye ihtiyaç duymadan kolaylıkla kullanabilmeleri için basit ve fonksiyonel olarak tasarlanmış ve geliştirilmiştir.

Geliştirilen silüet analizi modülü kullanılarak özellikle İstanbul gibi mega kentlerin silüet görüntüleri üretilebilir ve silüetteki zamansal değişim analiz edilebilir. Ayrıca, bu büyük kentlerdeki tarihi dokunun korunması için yeni yapılacak binaların silüete etkisi geliştirilen modül sayesinde tespit edilebilir. Örneğin, böyle bir modülün olması durumunda Zeytinburnu'ndaki 16/9 kuleleri gibi silüeti bozduğu için yıkım kararı verilen binaların inşa edilmeden önce henüz proje aşamasında iken silüet analizleri yapılarak yükseklikleri ve kat adetleri belirlenebilirdi. Bu nedenle tez kapsamında geliştirilen modülün belediyelere önemli katkı yapacağı düşünülmektedir.

5. KAYNAKLAR

- Akdag, S. G., Cagdas, G. ve Guney, C., 2010. Analyzing the Changes of Bosphoru Silhouette, in education and research in computer aided architectural design in Europe (eCAADe), September, Zürich, 28th Conference: Future Cities, 815-823.
- Ban, Y., Jakobsson, P., Kjell Dahl, L., ve Ranhagen, U., 2011. Visualization in ViSuCity, a tool for sustainable city planning, Proceedings of SIGRAD 2011. Evaluations of Graphics and Visualization - Efficiency, Usefulness, Accessibility, Usability, November 17-18, KTH, Stockholm, Sweden, 105-109.
- Cambray, B. de., 1993. Three-dimensional (3D) modelling in a geographical database, Proc. 11th International Symposium on Computer Assisted Cartography (AUTOCARTO 11), 338-347.
- Cambray, B. de. ve Yeh, T. S., 1994. A Multidimensional (2D, 2.5D, and 3D) Geographical Data Model, International Conference on Management of Data (COMAD'94), December, 317-336.
- Czerwinski, A., Sandmann, S., Stöcker-Meier, E. ve Plümer, L., 2007. Sustainable SDI for EU noise mapping in NRW – best practice for INSPIRE, International Journal of Spatial Data Infrastructures Research, 2, 90-111.
- Czyńska K., 2015. Impact of Tall Buildings on the Attractiveness of Urban Landscape – On the Example of Selected European Cities, Norway Grants, 131-144.
- Dobroja, L, 2015. Procedural 3D modeling and visualization of geotypical Bavarian rural buildings in Esri CityEngine software, Master Degree Thesis, Technische Universität München, München.
- Doğru, A. Ö., ve Seker, D. Z., 2009. Coğrafi Bilgi Sistemlerinde 3B Kent Modellerine Olanaklarının İrdelenmesi, TMMOB Coğrafi Bilgi Sistemleri Kongresi, Kasım, İzmir, 1-8.
- Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T. ve Teichmann, K., 2006, The Virtual 3D City Model of Berlin- Managing, Integrating, and Communicating Complex Urban Information, Proceedings of the 25th Urban Data Management Symposium, May, Aalborg, 15-27.
- Edvardsson, K. N., 2013. 3D GIS modeling using ESRI's CityEngine A case study from the University Jaume I in Castellón de la plana Spain, Master Thesis, The University Jaume I., Spain.
- Ekberg, F., 2007. An Approach for Representing Complex 3d Objects in GIS Applied To 3d Properties, Master Degree Thesis, University of Gavle, Gavle.

- El-Mekawy, M., 2010. Integrating BIM and GIS for 3D City Modelling: The Case of IFC and CityGML, Licentiate Thesis, Royal Institute of Technology (KTH), Stockholm.
- Emem, O., Batuk, F. ve Ayazlı, İ. E., 2008. UKVA Kapsamında UVDF-GML Tasarımı, HKM Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi, 2, 99.
- ESRI, 1998. ESRI Shapefile Technical Description, An ESRI White Paper, July.
- ESRI, 2008. The Multipatch Geometry Type, An ESRI White Paper, December.
- ESRI, 2013. CityEngine 2012.1, Yardım Dokümanı, ESRI Türkiye.
- Ford A., 2004. The visualisation of integrated 3D petroleum datasets in ArcGIS, 24th Annual Esri International User Conference, San Diego, California, 1-11.
- Ford A. ve James, P., 2005. Integration of 3D petroleum datasets in commercial GIS, 8th Conference on Geographic Information Science, Agile, 411-418.
- Ghosh, P. P. ve Rao Tvb, K., 2014. Landscape and Sunlight Analysis in 3D GIS Modeling for Smart City Planning – A Procedural Modeling Approach, Indian Cartographer, 34.
- Güney, C. Girginkaya, S. A., Çağdaş, G. ve Yavuz, S., 2012. Tailoring a Geomodel for Analyzing an Urban Skyline, Landscape and Urban Planning, 105, 160– 173.
- Jarroush, J. ve Even-Tzur, G., 2004. Constructive Solid Geometry as the Basis of 3D Future Cadastre, FIG Working week, Greece, 14.
- Karaş, İ. R., 2007. Üç Boyutlu Mekansal Nesnelerin Topolojik İlişkilerinin ve Uygulama Olanaklarının İrdelenmesi, Doktora Tezi, Y.T.Ü., Fen Bilimleri Enstitüsü, İstanbul.
- Kaya, N., 1999. Kurgu Planlama ve Modüler Aparat Tasarımında Unsur Kullanımı, Doktora Tezi, Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bursa.
- Kluijver, H. de. ve Stoter, J., 2003. Noise mapping and GIS: optimising quality and efficiency of noise effect studies, Computers, Environment and Urban Systems, 27, 1, 85-102.
- Kolbe, T. H., Gröger, G. ve Plümer, L., 2005. CityGML – Interoperable Access to 3D City Models, Proceedings of the Int. Symposium on Geo-information for Disaster Management, Delft, March, 21-23.
- Kolbe T. H., 2009. Representing and Exchanging 3D City Models with CityGML, Springer, Berlin, 15-32.

- Lee J., and Zlatanova S., 2008. A 3D data model and topological analyses for emergency response in urban areas, International Society for Photogrammetry and Remote Sensing (ISPRS), London, Geospatial information technology for emergency response, 143-168.
- Liu, L., Liqiang, Z., Chen, C. ve Hong, C., 2008. An Improved LOS Method for Implementing Visibility Analysis of 3D Complex Landscapes, International Conference on Computer Science and Software Engineering.
- Lipp, M., 2007. Interactive Computer Generated Architecture, Master Thesis, Vienna University of Technology, Viennea.
- Mao, B., 2011. Visualisation and Generalisation of 3D City Models, Doctora Thesis, K.T.H., Stockholm.
- Masry, S.E. and Lee, Y.C., 1988. An Introduction to Digital Mapping, Department of Surveying Engineering publication, UNB, Canada.
- Müller, P., Wonka. P., Haegler, S., Ulmer, A. ve Gool, L. V., 2006. Procedural modeling of buildings, ACM Transactions on Graphics, 25,3, 614 - 623.
- Nasar, J. L. ve Terzano, K., 2010. The Desirability of Views of City Skylines After Dark, Journal of Environmental Psychology, 30, 215–225.
- Ohori, K. A., 2016. Higher-dimensional modelling of geographic information, PhD thesis, Delft University of Technology, Delft
- Özdoğan, Ş. ve Başaraner, M., 2013. CityGML Standardında Ayrıntı Düzeylerinin Modellenmesi, TMMOB Coğrafi Bilgi Sistemleri Kongresi, Ankara, Kasım.
- Popelka, S., ve Vozenilek, V., 2015. Landscape Visibility Analysis and Their Visualisation, International Society for Photogrammetry and Remote Sensing conference paper ISPRS, 1-6.
- Rahman, A. A. ve Pilouk, M., 2007. Spatial Data Modeling For 3D GIS, Springer, Berlin, 25-42.
- Sadek, E. S. S. M., Ali. S. J. B. S., Rosdi, B. ve Kadzim, M. R. B. M. D, 2002. The Design and Development of a Virtual 3D City Model, 1-12.
- Schulte, C. ve Coors, V., 2008. Development of a CityGML ADE for dynamic 3D flood information, In Proceedings Joint ISCRAM-CHINA and GI4DM Conference on Information Systems for Crisis Management, Harbin, China.
- Singh, S. P., Jain, K. ve Mandla, V. R., 2014. Image based Virtual 3D Campus modeling by using CityEngine, American Journal of Engineering Science and Technology Research, 2327 – 8269, 2, 1, 1 – 10.

- Stoter, J. E. ve Ploeger, H. D., 2003. Registration of 3D objects crossing parcel boundaries, FIG Working week 2003, April, Paris.
- Stoter, J. ve Zlatanova, S., 2003. 3D GIS where are we standing?, International Society for Photogrammetry and Remote Sensing (ISPRS), Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis, October, Quebec city, 6.
- Stoter, J. E., 2004. 3D Cadastre. PhD Thesis, Delft University of Technology, Delft.
- Tafahomi, R., Hosseini, S. M. S. A., Lamit, H. ve Burshri, A., 2016. Application of GIS Method to Identify Urban Silhouette Form Case study: Mashhad city in Northeast of Iran, Planning Tech, 1-8.
- Tavernor, R. ve Gassner, G., 2010. Visual Consequences of the Plan: Managing London's Changing Skyline, City, Culture and Society, 1, 99 - 108.
- Tuan, A. N. G., 2013. Overview of Three-Dimensional GIS Data Models, International Journal of Future Computer and Communication, June, 2, 3, 271-274.
- URL-1, <https://tr.wikipedia.org/wiki/Voxel>. 21.04.2017
- URL-2, <http://johnrichie.com/V2/riche/isosurface/volume.html>. 22.04.2017.
- URL-3, <http://muallims.blogspot.com.tr/2009/04/cok-yuzlu-cisimlr-icin-eulr-formulu.html>, 20.04.2017.
- URL-4, https://en.wikipedia.org/wiki/Constructive_solid_geometry. 17.03.2017.
- URL-5, <http://citysurf.com.tr/tr/page.asp?id=3>. 3.03.2017.
- URL-7, <https://en.wikipedia.org/wiki/CityEngine>. 20.03.2017.
- URL-8, <http://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/what-is-the-3d-analyst-extension-.htm>. 20.05.2016.
- URL-9, <http://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/creating-a-line-of-sight.htm>. 18.05.2016.
- URL-10, <http://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/observer-pointss.html>. 15.02.2017.
- URL-11, <http://www.esri.com/news/arcnews/summer10articles/summer10gifs/p13p1-lg.jpg>. 5.02.2016.
- URL-12, <http://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/how-skyline-works.htm>. 2.04.2017.

- URL-13, <http://desktop.arcgis.com/en/arcmap/10.3/manage-data/editing-fundamentals/about-direction-measuring-systems-and-units.htm>. 12.04.2017.
- URL-14, <http://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/skyline-analysis-an-urban-environmental-physical-design-aid.htm>. 15.05.2016.
- URL-15, <http://programarcadegames.com/index.php?chapter=libraries&lang=tr>. 24.03.2016.
- URL-16, <http://www.istihza.com/>. 8.11.2016.
- URL-17, <http://www.pythondersleri.com/p/python-baslangc.html>. 20.02.2016.
- URL-18, <http://www.pythondersleri.com/2013/04/python-ile-uygulama-gelistirme.html>. 18.08.2016.
- Van Wees, J.D., Versseput, R.W., H.J.Simmelink, Allard, R. R. L. ve H. J. M. Pagnier., 2002. Shared Earth system models for the dutch subsurface, Netherlands Institute of Applied Geoscience TNO-National Geological Survey.
- Yang, P. P., Putra, S. Y. ve Li, W., 2007. Viewsphere: a GIS Based 3D Visibility Analysis for Urban Design Evaluation, Environment and Planning B: Planning and Design, 34, 971-992
- Yusoff, N. A. H., Noor, A. M. ve Ghazali, R., 2014. City skyline conservation: sustaining the premier image of Kuala Lumpur, 4th International Conference on Sustainable Future for Human Security, Sustain, 583 – 592
- Yücel, M. A. ve Selçuk, M., 2009. Üç Boyutlu Kent Modellerinde Ayrıntı Düzeyi (LoD) Kavramı, HKM Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi 2,101.
- Zlatanova, S., Painsil, J. ve Tempfli, K., 1998. 3D object reconstruction from aerial stereo images, Journal of WSCG. 6, 1-3
- Zlatanova, S., Rahman, A. A., ve Pilouk M., 2002. Trends in 3D GIS Development, Journal of Geospatial Engineering, 4, 2, 1-10.
- Zlatanova, S., Rahman, A. A., ve Pilouk, M., 2002. 3D GIS: Current status and perspectives, Proceedings of the Joint Conference on Geo-spatial theory, Processing and Applications, July, Ottawa, 6.

6. EKLER

EK 1.Silüet analiz modülü kodları

```
#-*- coding: cp1254 -*-
import arcpy
import pythonaddins
import math
import time
import sys
from PyQt4 import QtGui, QtSvg

arcpy.env.overwriteOutput = True
arcpy.env.addOutputsToMap = True
arcpy.CheckExtension ('3D')
arcpy.CheckExtension ("Spatial")
arcpy.CheckOutExtension ("DataInteroperability")

class Layer (object):
    """Implementation for Kod_addin.combobox (ComboBox)"""
    def __init__(self):
        self.items = [ ]
        self.editable = True
        self.enabled = True
        self.dropdownWidth = 'WWWWWWWWWW'
        self.width = 'WWWWWWWW'

    def onSelChange (self, selection):
        tool.enabled = True
        combobox_1.value = "
        combobox_1.refresh ()
        combobox_2.value = "
        combobox_2.refresh ()

        Layer = arcpy.mapping.ListLayers (self.mxd, selection)[0]

        global Raster1
        Raster1 = "C:\\Kod\\Results\\1raster"
        arcpy.CopyRaster_management (Layer, Raster1)
        arcpy.Delete_management (Layer, "")
        arcpy.RefreshActiveView ()

    def onFocus (self, focused):
        if focused:
            self.mxd = arcpy.mapping.MapDocument ("CURRENT")
            layers = arcpy.mapping.ListLayers(self.mxd)
            self.items = [ ]
```

EK 1'in devamı

```

        for layer in layers:
            if layer.isRasterLayer:
                if (arcpy.Describe(layer).datasetType == "RasterDataset"):
                    self.items.append (layer.name)

class Observation_Point (object):
    """Implementation for Kod_addin.tool (Tool)"""
    def __init__(self):
        self.enabled = False
        self.shape = None
        self.cursor = 3

    def onMouseDownMap(self, x, y, button, shift):
        tool_1.enabled = True
        mxd = arcpy.mapping.MapDocument ("CURRENT")
        dataframe = arcpy.mapping.ListDataFrames (mxd) [0]

        global ObservationPoint1
        ObservationPoint1 = "C:\\Kod\\Results\\1ObservationPoint.shp"
        PointXY = arcpy.PointGeometry (arcpy.Point(x,y), None, True, True)
        arcpy.CopyFeatures_management (PointXY, ObservationPoint1)
        arcpy.RefreshActiveView ()

        arcpy.gp.ExtractMultiValuesToPoints_sa (ObservationPoint1, "1raster ZValue",
None)
        arcpy.RefreshActiveView ()
        arcpy.RefreshTOC ()

        rows = arcpy.SearchCursor (ObservationPoint1)
        row = rows.next ()
        while row:
            z = row.ZValue
            row = rows.next ()

        global ObservationPointZ
        ObservationPointZ = z + 1.75

        PointXYZ = arcpy.PointGeometry (arcpy.Point(x,y,ObservationPointZ), None, True,
True)
        arcpy.CopyFeatures_management (PointXYZ, ObservationPoint1)
        arcpy.AddXY_management (ObservationPoint1)
        arcpy.RefreshActiveView ()

class Silhouette_Line (object):
    """Implementation for Kod_addin.tool_1 (Tool)"""
    def __init__(self):
        self.enabled = False

```

EK 1'in devamı

```

self.shape = "Line"
self.cursor = 3
def online (self, line_geometry):
    combobox_1.enabled = True
    mxd = arcpy.mapping.MapDocument ("CURRENT")
    dataframe = arcpy.mapping.ListDataFrames(mxd)[0]

    global SilhouetteLine2
    SilhouetteLine2 = "C:\\Kod\\Results\\2SilhouetteLine.shp"
    arcpy.CopyFeatures_management (line_geometry, SilhouetteLine2)
    arcpy.RefreshActiveView ()

    global SilhouetteLineFVtoP3
    SilhouetteLineFVtoP3 = "C:\\Kod\\Results\\3SilhouetteLineFVtoP.shp"
    arcpy.FeatureVerticesToPoints_management(SilhouetteLine2, SilhouetteLineFVtoP3,
"ALL")
    arcpy.AddXY_management (SilhouetteLineFVtoP3)

    global GenerateNearTable4
    GenerateNearTable4 = "C:\\Kod\\Results\\4GenerateNearTable.dbf"
    arcpy.GenerateNearTable_analysis (ObservationPoint1, SilhouetteLineFVtoP3,
GenerateNearTable4, "", "LOCATION", "ANGLE", "ALL", "0")

    global fromAzimuth
    fromAzimuth = 0
    global toAzimuth
    toAzimuth = 0

    kayitsayisi = 0
    rows = arcpy.SearchCursor (GenerateNearTable4)
    row = rows.next()
    while row:
        kayitsayisi += 1
        if kayitsayisi == 1:
            fromAzimuth = row.NEAR_ANGLE
            row = rows.next ()
        else:
            toAzimuth = row.NEAR_ANGLE
            break
    fromAzimuth = 90 - fromAzimuth
    toAzimuth = 90 - toAzimuth

    if fromAzimuth < 0 :
        fromAzimuth = fromAzimuth + 360
    if toAzimuth < 0 :
        toAzimuth = toAzimuth + 360

```

EK 1'in devamı

```

class Building (object):
    """Implementation for Kod_addin.combobox_1 (ComboBox)"""
    def __init__(self):
        self.items = [ ]
        self.editable = True
        self.enabled = False
        self.dropdownWidth = 'WWWWWWWWWWWWWWWW'
        self.width = 'WWWWWWWWWW'

    def onSelChange (self, selection):
        button.enabled = True
        global Building
        Building = arcpy.mapping.ListLayers("", selection)[0]
        global BuildingAsShape5
        BuildingAsShape5 = "C:\\Kod\\Results\\5BuildingAsShape.shp"
        arcpy.CopyFeatures_management(Building, BuildingAsShape5)
        arcpy.RefreshActiveView()

    def onFocus (self, focused):
        if focused:
            self.mxd = arcpy.mapping.MapDocument ("CURRENT")
            layers = arcpy.mapping.ListLayers ("")
            self.items = [ ]
            for layer in layers:
                if layer.isFeatureLayer:
                    if(arcpy.Describe(layer).shapeType == "MultiPatch"):
                        self.items.append (layer.name)

class Silhouette (object):
    """Implementation for Kod_addin.button (Button)"""
    def __init__(self):
        self.enabled = False
        self.checked = False

    def onClick(self):
        combobox_2.enabled = True
        arcpy.AddField_management (ObservationPoint1, "fromA", "SHORT")

        arcpy.CalculateField_management (ObservationPoint1, "fromA", fromAzimuth,
"PYTHON")
        arcpy.AddField_management (ObservationPoint1, "toA", "SHORT")

        arcpy.CalculateField_management (ObservationPoint1, "toA", toAzimuth,
"PYTHON")

        Skyline6 = "C:\\Kod\\Results\\6Skyline.shp"

```

EK 1'in devamı

```
arcpy.Skyline_3d (ObservationPoint1, Skyline6, Raster1, "1000 Meters", "0 Meters",
BuildingAsShape5, "CONVEX_FOOTPRINT", "fromA", "toA", "1", "100 Meters",
"NO_SEGMENT_SKYLINE", "100", "VERTICAL_ANGLE",
"SKYLINE_MAXIMUM", "NO_CURVATURE", "NO_REFRACTION", "0.13", "0",
"NO_CREATE_SILHOUETTES")
```

```
SkylineIntPoint7 = "C:\\Kod\\Results\\7SkylineIntPoint.shp"
arcpy.Intersect3DLineWithMultiPatch_3d (Skyline6, BuildingAsShape5,
"IDS_ONLY", SkylineIntPoint7, "")
```

```
result = arcpy.GetCount_management (SkylineIntPoint7)
count = int (result.getOutput(0))
```

```
if count == 0:
    message1 = "Durulan noktadan binalar görünmüyor. Durulan noktayı değiştir."
    pythonaddins.MessageBox(message 1, "UYARI", 0)
```

```
else:
```

```
    global SkylineIntPointAsAscending8
    SkylineIntPointAsAscending8 =
C:\\Kod\\Results\\8SkylineIntPointAsAscending.shp"
    arcpy.Sort_management (SkylineIntPoint7, SkylineIntPointAsAscending8,
"MPATCH_OID ASCENDING", "UR")
```

```
select_text = " "
select_text_1 = " "
select_text_2 = " "
merge_text = " "
```

```
kayitsayisi = 0
```

```
fields = ["MPATCH_OID"]
```

```
result = arcpy.GetCount_management (SkylineIntPointAsAscending8)
count = int (result.getOutput(0))
```

```
rows = arcpy.SearchCursor (SkylineIntPointAsAscending8)
row = rows.next ()
```

```
with arcpy.da.SearchCursor (SkylineIntPointAsAscending8, fields) as cursor:
```

```
    for row in cursor:
        kayitsayisi += 1
```

```
    if kayitsayisi != count:
```

```
        select_text_2 = select_text_2 + "FID = " + str (row[0]) + " OR "
        select_text = "FID = " + str(row[0])
```

```
        if select_text != select_text_1:
```

EK 1'in devamı

```

arcpy.MakeFeatureLayer_management          (BuildingAsShape5,
"building_lyr")
arcpy.SelectLayerByAttribute_management    ("building_lyr",
"NEW_SELECTION", select_text)

BuildingSelected10 = "C:\\Kod\\Results\\BuildingID_" + str (row[0]) +
".shp"
arcpy.CopyFeatures_management ("building_lyr", BuildingSelected10)
arcpy.SelectLayerByAttribute_management    ("building_lyr",
"CLEAR_SELECTION")
arcpy.AddField_management    (BuildingSelected10,    "BuildingID",
"LONG")
arcpy.CalculateField_management (BuildingSelected10, "BuildingID",
str(row[0]), "PYTHON")

Skyline11 = "C:\\Kod\\Results\\11Skyline" + str (row[0]) + ".shp"
arcpy.Skyline_3d (ObservationPoint1, Skyline11, "", "0 Meters", "0
Meters", BuildingSelected10, "FULL_DETAIL", "0", "360", "1", "0 Meters",
"SEGMENT_SKYLINE", "100", "VERTICAL_ANGLE", "SKYLINE_MAXIMUM",
"NO_CURVATURE", "NO_REFRACTION", "0.13", "0",
"NO_CREATE_SILHOUETTES")

merge_text = merge_text + "11Skyline" + str(row[0]) + ";"
select_text_1 = select_text
row = rows.next ()

else:
select_text_2 = select_text_2 + "FID = " + str(row[0])
select_text = "FID = " + str(row[0])

if select_text != select_text_1:
arcpy.MakeFeatureLayer_management          (BuildingAsShape5,
"building_lyr")
arcpy.SelectLayerByAttribute_management    ("building_lyr",
"NEW_SELECTION", select_text)

BuildingSelected10 = "C:\\Kod\\Results\\BuildingID_" + str (row[0]) +
".shp"
arcpy.CopyFeatures_management ("building_lyr", BuildingSelected10)
arcpy.SelectLayerByAttribute_management("building_lyr",
"CLEAR_SELECTION")
arcpy.AddField_management    (BuildingSelected10,    "BuildingID",
"LONG")
arcpy.CalculateField_management (BuildingSelected10, "BuildingID",
str(row[0]), "PYTHON")

Skyline11 = "C:\\Kod\\Results\\11Skyline" + str (row[0]) + ".shp"

```

EK 1'in devamı

```

arcpy.Skyline_3d (ObservationPoint1, Skyline11, "", "0 Meters", "0
Meters", BuildingSelected10, "FULL_DETAIL", "0", "360", "1", "0 Meters",
"SEGMENT_SKYLINE", "100", "VERTICAL_ANGLE", "SKYLINE_MAXIMUM",
"NO_CURVATURE", "NO_REFRACTION", "0.13", "0",
"NO_CREATE_SILHOUETTES")

```

```

merge_text = merge_text + "11Skyline" + str (row[0])
select_text_1 = select_text
break

```

```

arcpy.MakeFeatureLayer_management (BuildingAsShape5, "building_lyr")
arcpy.SelectLayerByAttribute_management ("building_lyr",
"NEW_SELECTION", select_text_2)

```

```

global BuildingSelected9
BuildingSelected9 = "C:\\Kod\\Results\\9BuildingSelected.shp"
arcpy.CopyFeatures_management ("building_lyr", BuildingSelected9)
arcpy.SelectLayerByAttribute_management ("building_lyr",
"CLEAR_SELECTION")

```

```

SkylineMerge12 = "C:\\Kod\\Results\\12SkylineMerge.shp"
SkylineIntPoint13 = "C:\\Kod\\Results\\13SkylineIntPoint.shp"

```

```

arcpy.Intersect3DLineWithMultiPatch_3d (SkylineMerge12, BuildingSelected9,
"IDS_ONLY", SkylineIntPoint13, "")
arcpy.AddXY_management (SkylineIntPoint13)

```

```

global SkylineIntPointAsAscending14
SkylineIntPointAsAscending14 =
"C:\\Kod\\Results\\14SkylineIntPointAsAscending.shp"
arcpy.Sort_management (SkylineIntPoint13, SkylineIntPointAsAscending14,
"MPATCH_OID ASCENDING", "UR")

```

```

global BuildingAsShapeFtoP15
BuildingAsShapeFtoP15 = "C:\\Kod\\Results\\15BuildingAsShapeFtoP.shp"
arcpy.FeatureToPoint_management (BuildingAsShape5, BuildingAsShapeFtoP15,
"CENTROID")

```

```

global BuildingSelectedFtoP16
BuildingSelectedFtoP16 = "C:\\Kod\\Results\\16BuildingSelectedFtoP.shp"
arcpy.FeatureToPoint_management (BuildingSelected9, BuildingSelectedFtoP16,
"CENTROID")

```

```

global SpatialJoin17
SpatialJoin17 = "C:\\Kod\\Results\\17SpatialJoin.shp"
arcpy.SpatialJoin_analysis (BuildingSelectedFtoP16, BuildingAsShapeFtoP15,
SpatialJoin17, "JOIN_ONE_TO_ONE", "KEEP_ALL", "", "INTERSECT", "", "")

```


EK 1'in devamı

```
arcpy.JoinField_management (BuildingSelected9, "FID", SpatialJoin17, "FID",
"ORIG_FID_1")
```

```
arcpy.AddField_management (BuildingSelected9, "BuildingID", "LONG")
arcpy.CalculateField_management (BuildingSelected9, "BuildingID",
"!ORIG_FID_1!", "PYTHON")
```

```
SilhouetteLineFVtoPidFirst18 =
"C:\\Kod\\Results\\18SilhouetteLineFVtoPidFirst.shp"
SilhouetteLineFVtoPidSecond19 =
"C:\\Kod\\Results\\19SilhouetteLineFVtoPidSecond.shp"
arcpy.Select_analysis (SilhouetteLineFVtoP3, SilhouetteLineFVtoPidFirst18,
"\"FID\" = 0")
arcpy.Select_analysis (SilhouetteLineFVtoP3, SilhouetteLineFVtoPidSecond19,
"\"FID\" = 1")
PointDistanceTable20 = "C:\\Kod\\Results\\20PointDistanceTable.dbf"
arcpy.PointDistance_analysis (SilhouetteLineFVtoPidFirst18,
SilhouetteLineFVtoPidSecond19, PointDistanceTable20, "")
```

```
Ya = 0
Xa = 0
Yb = 0
Xb = 0
rows = arcpy.SearchCursor (PointDistanceTable20)
row = rows.next ()
while row:
    Xb = Xb + row.DISTANCE
    row = rows.next()
```

```
ya = 0
xa = 0
rows = arcpy.SearchCursor (SilhouetteLineFVtoPidFirst18)
row = rows.next ()
while row:
    ya = ya + row.POINT_Y
    xa = xa + row.POINT_X
    row = rows.next ()
```

```
yb = 0
xb = 0
rows = arcpy.SearchCursor (SilhouetteLineFVtoPidSecond19)
row = rows.next ()
while row:
    yb = yb + row.POINT_Y
    xb = xb + row.POINT_X
    row = rows.next ()
```

EK 1'in devamı

```

global a
global b
global Xo
a = ((Yb-Ya)*(xb-xa)-(Xb-Xa)*(yb-ya))/(Xb*Xb)
b = ((Xb-Xa)*(xb-xa)-(Yb-Ya)*(yb-ya))/(Xb*Xb)
Xo = Xa-(b*xa)+(a*ya)

select_text = " "
select_text_1 = " "
select_text_2 = " "
merge_text_2 = " "
merge_text_3 = " "

kayitsayisi = 0

fields = ["MPATCH_OID"]

result = arcpy.GetCount_management (SkylineIntPointAsAscending14)
count = int (result.getOutput(0))

rows = arcpy.SearchCursor (SkylineIntPointAsAscending14)
row = rows.next ()

with arcpy.da.SearchCursor (SkylineIntPointAsAscending14, fields) as cursor:
    for row in cursor:

        kayitsayisi += 1

        if kayitsayisi != count:
            select_text_2 = select_text_2 + "\"MPATCH_OID\" = " + str(row[0])+ "
OR "
            select_text = "\"MPATCH_OID\" = " + str(row[0])

            if select_text != select_text_1:
                PointSelected21 = "C:\\Kod\\Results\\21PointSelected" + str(row[0]) +
".shp"
                PointSelected21_2 = "C:\\Kod\\Results\\21PointSelected" + str(row[0]) +
"_2.dbf"
                PointSelected21_3 = "C:\\Kod\\Results\\21PointSelected" + str(row[0]) +
"_3.shp"
                PointSelected21_4 = "C:\\Kod\\Results\\21PointSelected" + str(row[0]) +
"_4.shp"
                arcpy.Select_analysis(SkylineIntPointAsAscending14, PointSelected21,
select_text)
                arcpy.AddField_management (PointSelected21, "POINTZ", "SHORT")
                arcpy.CalculateField_management (PointSelected21, "POINTZ",
"!POINT_Z!", "PYTHON")

```

EK 1'in devamı

```

arcpy.Statistics_analysis (PointSelected21, PointSelected21_2, "POINTZ
MIN", "")

deger_text = " "
deger_text_2 = " "
deger = 0
rows2 = arcpy.SearchCursor (PointSelected21_2)
row2 = rows2.next ()
while row2:
    deger = deger + row2.MIN_POINTZ
    row2 = rows2.next ()
deger_text = "\"POINTZ\" = " + str(deger)
deger_text_2 = "POINTZ > " + str(deger)

arcpy.Select_analysis(PointSelected21, PointSelected21_3, deger_text)
arcpy.CalculateField_management (PointSelected21_3, "Ynew",
ObservationPointZ, "PYTHON")

arcpy.Select_analysis(PointSelected21, PointSelected21_4, deger_text_2)

arcpy.AddField_management (PointSelected21_4, "Ynew", "SHORT")
arcpy.CalculateField_management (PointSelected21_4, "Ynew",
"!POINTZ!", "PYTHON")

merge_text_2 = merge_text_2 + "21PointSelected" + str(row[0]) + "_3;"
+ "21PointSelected" + str(row[0]) + "_4;"
row = rows.next ()

else:
select_text_2 = select_text_2 + "\"MPATCH_OID\" = " + str(row[0])
select_text = "\"MPATCH_OID\" = " + str(row[0])

if select_text != select_text_1:
PointSelected21 = "C:\\Kod\\Results\\21PointSelected" + str(row[0]) +
".shp"
PointSelected21_2 = "C:\\Kod\\Results\\21PointSelected" + str(row[0]) +
"_2.dbf"
PointSelected121_3 = "C:\\Kod\\Results\\21PointSelected" + str(row[0])
+ "_3.shp"
PointSelected21_4 = "C:\\Kod\\Results\\21PointSelected" + str(row[0]) +
"_4.shp"
arcpy.Select_analysis(SkylineIntPointAsAscending14, PointSelected21,
select_text)
arcpy.CalculateField_management (PointSelected21, "POINTZ",
"!POINT_Z!", "PYTHON")
arcpy.Statistics_analysis(PointSelected21, PointSelected21_2, "POINTZ
MIN", "")

```

EK 1'in devamı

```

deger_text = " "
deger_text_2 = " "
deger = 0
rows2 = arcpy.SearchCursor (PointSelected21_2)
row2 = rows2.next ()
while row2:
    deger = deger + row2.MIN_POINTZ
    row2 = rows2.next ()
deger_text = "\"POINTZ\" = " + str(deger)
deger_text_2 = "POINTZ > " + str(deger)

arcpy.Select_analysis(PointSelected21, PointSelected21_3, deger_text)

arcpy.AddField_management (PointSelected21_3, "Ynew", "SHORT")
arcpy.CalculateField_management (PointSelected21_3, "Ynew",
ObservationPointZ, "PYTHON")

arcpy.Select_analysis(PointSelected21, PointSelected21_4, deger_text_2)
arcpy.AddField_management (PointSelected21_4, "Ynew", "SHORT")
arcpy.CalculateField_management (PointSelected21_4, "Ynew",
"!POINTZ!", "PYTHON")

merge_text_2 = merge_text_2 + "21PointSelected" + str(row[0]) + "_3;"
+ "21PointSelected" + str(row[0]) + "_4;"
select_text_1 = select_text
break

PointsMerge22 = "C:\\Kod\\Results\\22PointsMerge.shp"
arcpy.Merge_management (merge_text_2, PointsMerge22, None)

arcpy.AddField_management (PointsMerge22, "a", "DOUBLE")
arcpy.CalculateField_management (PointsMerge22, "a", a, "PYTHON")

arcpy.AddField_management (PointsMerge22, "b", "DOUBLE")
arcpy.CalculateField_management (PointsMerge22, "b", b, "PYTHON")

arcpy.AddField_management (PointsMerge22, "X0", "DOUBLE")
arcpy.CalculateField_management (PointsMerge22, "X0", Xo, "PYTHON")

arcpy.AddField_management (PointsMerge22, "Xnew", "DOUBLE")
arcpy.CalculateField_management (PointsMerge22, "Xnew", "!X0!+( !b!*
!POINT_X!)-( !a! * !POINT_Y! )", "PYTHON")

global BuildingSelectedFtoP23
BuildingSelectedFtoP23 = "C:\\Kod\\Results\\23BuildingSelectedFtoP.shp"
arcpy.FeatureToPoint_management (BuildingSelected9, BuildingSelectedFtoP23,
"CENTROID")

```

EK 1'in devamı

```

    arcpy.Near_analysis (BuildingSelectedFtoP23, ObservationPoint1, "",
"LOCATION", "NO_ANGLE")

    global BuildingSelectedFtoPAsDescending24
    BuildingSelectedFtoPAsDescending24 =
"C:\\Kod\\Results\\24BuildingSelectedFtoPAsDescending.shp"
    arcpy.Sort_management (BuildingSelectedFtoP23,
BuildingSelectedFtoPAsDescending24, "NEAR_DIST DESCENDING", "UR")

    arcpy.AddField_management (BuildingSelectedFtoPAsDescending24, "List",
"SHORT")

    arcpy.CalculateField_management (BuildingSelectedFtoPAsDescending24, "List",
"!FID!", "PYTHON")

    arcpy.JoinField_management(PointsMerge22, "MPATCH_OID",
BuildingSelectedFtoPAsDescending24, "ORIG_FID", "NEAR_DIST")
    arcpy.JoinField_management(PointsMerge22, "MPATCH_OID",
BuildingSelectedFtoPAsDescending24, "ORIG_FID", "List")

    global PointsMergeAsCoordTrans25
    PointsMergeAsCoordTrans25 =
"C:\\Kod\\Results\\25PointsMergeAsCoordTrans.shp"
    arcpy.CopyFeatures_management (PointsMerge22, PointsMergeAsCoordTrans25)

    codeblock = """def XYsetVALUE ( shape, X_value, Y_value):
    point = shape.getPart(0)
    point.X = X_value
    point.Y = Y_value
    return point"""
    arcpy.CalculateField_management (PointsMergeAsCoordTrans25, "Shape",
"XYsetVALUE ( !SHAPE!, !Xnew!, !Ynew! )", "PYTHON", codeblock)
    arcpy.AddXY_management (PointsMergeAsCoordTrans25)

    arcpy.AddField_management (PointsMergeAsCoordTrans25, "Color", "SHORT")
    arcpy.CalculateField_management (PointsMergeAsCoordTrans25, "Color", "-1",
"PYTHON")

    arcpy.JoinField_management (PointsMergeAsCoordTrans25, "MPATCH_OID",
SpatialJoin17, "TARGET_FID", "ORIG_FID_1")
    arcpy.AddField_management (PointsMergeAsCoordTrans25, "BuildingID",
"LONG")
    arcpy.CalculateField_management (PointsMergeAsCoordTrans25, "BuildingID",
"!ORIG_FID_1!", "PYTHON")

    global SilhouetteWithoutColor26
    SilhouetteWithoutColor26 = "C:\\Kod\\Results\\26SilhouetteWithoutColor.shp"

```

EK 1'in devamı

```
arcpy.MinimumBoundingGeometry_management (PointsMergeAsCoordTrans25,
SilhouetteWithoutColor26, "", "LIST", "List;Color;BuildingID", "MBG_FIELDS")
```

```
arcpy.MakeFeatureLayer_management (SilhouetteWithoutColor26, "building_lyr")
arcpy.SelectLayerByAttribute_management ("building_lyr",
"NEW_SELECTION", "")
mxd = arcpy.mapping.MapDocument ("CURRENT")
df = arcpy.mapping.ListDataFrames (mxd) [0]
df.zoomToSelectedFeatures ()
arcpy.SelectLayerByAttribute_management ("building_lyr",
"CLEAR_SELECTION")
```

```
Silhouette27 = "C:\\Kod\\Results\\27Silhouette_" +
str(time.strftime("%d%m%Y_%H%M%S")) + ".lyr"
arcpy.ZRenderer_stats (SilhouetteWithoutColor26, "Color", Silhouette27)
```

```
SilhouetteAsSvg28 = "SVG,C:\\Kod\\Results\\28SilhouetteAsSvg.svg"
arcpy.QuickExport_interop (Silhouette27, SilhouetteAsSvg28)
```

```
app = QtGui.QApplication (sys.argv)
```

```
window = QtSvg.QSvgWidget ("C:\\Kod\\Results\\28SilhouetteAsSvg.svg")
window.setWindowTitle ("Silhouette View of Building")
```

```
window.setGeometry (0, 0, 800, 300)
window.move(400,200)
```

```
window.show ()
```

```
class New_Building (object):
```

```
    """Implementation for Kod_addin.combobox_2 (ComboBox)"""
```

```
    def __init__(self):
```

```
        self.items = [ ]
```

```
        self.editable = True
```

```
        self.enabled = False
```

```
        self.dropdownWidth = 'WWWWWWWWWWWWWWWWWW'
```

```
        self.width = 'WWWWWWWWW'
```

```
    def onSelChange (self, selection):
```

```
        button_1.enabled = True
```

```
        NewBuilding = arcpy.mapping.ListLayers ("", selection)[0]
```

```
        global NewBuildingAsShape29
```

```
        NewBuildingAsShape29 = "C:\\Kod\\Results\\29NewBuildingAsShape.shp"
```

```
        arcpy.CopyFeatures_management (NewBuilding, NewBuildingAsShape29)
```

```
        arcpy.RefreshActiveView ()
```

```
    def onFocus (self, focused):
```

EK 1'in devamı

```

if focused:
    self.mxd = arcpy.mapping.MapDocument ("CURRENT")
    layers = arcpy.mapping.ListLayers ("")
    self.items = [ ]
    for layer in layers:
        if layer.isFeatureLayer:
            if(arcpy.Describe(layer).shapeType == "MultiPatch"):
                self.items.append (layer.name)

class New_Silhouette(object):
    """Implementation for Kod_addin.button_1 (Button)"""
    def __init__(self):
        self.enabled = False
        self.checked = False

    def onClick (self):
        combobox_3.enabled = True

        Skyline30 = "C:\\Kod\\Results\\30Skyline.shp"
        arcpy.Skyline_3d (ObservationPoint1, Skyline30, "", "0 Meters", "0 Meters",
        NewBuildingAsShape29, "FULL_DETAIL", "fromA", "toA", "1", "0 Meters",
        "SEGMENT_SKYLINE", "100", "VERTICAL_ANGLE", "SKYLINE_MAXIMUM",
        "NO_CURVATURE", "NO_REFRACTION", "0.13", "0",
        "NO_CREATE_SILHOUETTES")

        SkylineIntPoint31 = "C:\\Kod\\Results\\31SkylineIntPoint.shp"
        arcpy.Intersect3DLineWithMultiPatch_3d (Skyline30, NewBuildingAsShape29,
        "IDS_ONLY", SkylineIntPoint31, "")

        result = arcpy.GetCount_management (SkylineIntPoint31)
        count = int (result.getOutput(0))

        if count == 0:
            message 2 = "Yeni bina silüet hattında (görüş açısı aralığında) değil. Yeni bina
            silüeti bozmuyor."
            pythonaddins.MessageBox (message 2, "UYARI", 0)

        else:
            Skyline30 = "C:\\Kod\\Results\\30Skyline.shp"
            arcpy.Skyline_3d (ObservationPoint1, Skyline30, "", "0 Meters", "0 Meters",
            NewBuildingAsShape29, "FULL_DETAIL", "0", "360", "1", "0 Meters",
            "SEGMENT_SKYLINE", "100", "VERTICAL_ANGLE", "SKYLINE_MAXIMUM",
            "NO_CURVATURE", "NO_REFRACTION", "0.13", "0",
            "NO_CREATE_SILHOUETTES")

            SkylineIntPoint31 = "C:\\Kod\\Results\\31SkylineIntPoint.shp"

```

EK 1'in devamı

```

arcpy.Intersect3DLineWithMultiPatch_3d (Skyline30, NewBuildingAsShape29,
"IDS_ONLY", SkylineIntPoint31, "")
arcpy.AddXY_management (SkylineIntPoint31)

global SkylineIntPoint31_3
SkylineIntPoint31_2 = "C:\\Kod\\Results\\31SkylineIntPoint_2.dbf"
SkylineIntPoint31_3 = "C:\\Kod\\Results\\31SkylineIntPoint_3.shp"
SkylineIntPoint31_4 = "C:\\Kod\\Results\\31SkylineIntPoint_4.shp"

arcpy.AddField_management (SkylineIntPoint31, "POINTZ", "SHORT")
arcpy.CalculateField_management (SkylineIntPoint31, "POINTZ", "!POINT_Z!",
"PYTHON")
arcpy.Statistics_analysis (SkylineIntPoint31, SkylineIntPoint31_2, "POINTZ
MIN", "")

deger_text2 = " "
deger_text2_2 = " "
deger2 = 0
rows3 = arcpy.SearchCursor (SkylineIntPoint31_2)
row3 = rows3.next ()
while row3:
    deger2 = deger2 + row3.MIN_POINTZ
    row3 = rows3.next ()

deger_text2 = "\"POINTZ\" = " + str(deger2)
deger_text2_2 = "POINTZ > " + str(deger2)

arcpy.Select_analysis (SkylineIntPoint31, SkylineIntPoint31_3, deger_text2)
arcpy.AddField_management (SkylineIntPoint31_3, "Ynew", "SHORT")
arcpy.CalculateField_management (SkylineIntPoint31_3, "Ynew",
ObservationPointZ, "PYTHON")

arcpy.Select_analysis(SkylineIntPoint31, SkylineIntPoint31_4, deger_text2_2)
arcpy.AddField_management (SkylineIntPoint31_4, "Ynew", "SHORT")
arcpy.CalculateField_management (SkylineIntPoint31_4, "Ynew", "!POINTZ!",
"PYTHON")

SkylineIntPoint31_Merge = "C:\\Kod\\Results\\31SkylineIntPoint_Merge.shp"
arcpy.Merge_management ("31SkylineIntPoint_3; 31SkylineIntPoint_4;",
SkylineIntPoint31_Merge, None)

arcpy.AddField_management (SkylineIntPoint31_Merge, "X0", "DOUBLE")
arcpy.CalculateField_management (SkylineIntPoint31_Merge, "X0", Xo,
"PYTHON")

arcpy.AddField_management (SkylineIntPoint31_Merge, "a", "DOUBLE")
arcpy.CalculateField_management (SkylineIntPoint31_Merge, "a", a, "PYTHON")

```


EK 1'in devamı

```

arcpy.AddField_management (SkylineIntPoint31_Merge, "b", "DOUBLE")
arcpy.CalculateField_management (SkylineIntPoint31_Merge, "b", b, "PYTHON")

arcpy.AddField_management (SkylineIntPoint31_Merge, "Xnew", "DOUBLE")
arcpy.CalculateField_management (SkylineIntPoint31_Merge, "Xnew", "!X0!+(
!b!* !POINT_X!)-( !a! * !POINT_Y! )", "PYTHON")

global SkylineIntPointAsCoordTrans32
SkylineIntPointAsCoordTrans32 =
"C:\\Kod\\Results\\32SkylineIntPointAsCoordTrans.shp"
arcpy.CopyFeatures_management (SkylineIntPoint31_Merge,
SkylineIntPointAsCoordTrans32)

codeblock = """def XYsetVALUE( shape, X_value, Y_value):
point = shape.getPart (0)
point.X = X_value
point.Y = Y_value
return point"""
arcpy.CalculateField_management (SkylineIntPointAsCoordTrans32, "Shape",
"XYsetVALUE ( !SHAPE!, !Xnew!, !Ynew! )", "PYTHON", codeblock)

global NewBuildingAsShapeFtoP33
NewBuildingAsShapeFtoP33 =
"C:\\Kod\\Results\\33NewBuildingAsShapeFtoP.shp"
arcpy.FeatureToPoint_management (NewBuildingAsShape29,
NewBuildingAsShapeFtoP33, "CENTROID")

PointDistanceTable34 = "C:\\Kod\\Results\\34PointDistanceTable.dbf"
arcpy.PointDistance_analysis (NewBuildingAsShapeFtoP33, ObservationPoint1,
PointDistanceTable34, "")
rows = arcpy.SearchCursor (PointDistanceTable34)
row = rows.next()

while row:
dist = row.DISTANCE
row = rows.next ()
arcpy.AddField_management (NewBuildingAsShapeFtoP33, "NEAR_DIST",
"DOUBLE")
arcpy.CalculateField_management (NewBuildingAsShapeFtoP33, "NEAR_DIST",
dist, "PYTHON")

arcpy.AddField_management (NewBuildingAsShapeFtoP33, "BuildingID",
"LONG")
arcpy.CalculateField_management (NewBuildingAsShapeFtoP33, "BuildingID", "-
1", "PYTHON")

BuildingsFtoPMerge35 = "C:\\Kod\\Results\\35BuildingsFtoPMerge.shp"

```

EK 1'in devamı

```
arcpy.Merge_management ("23BuildingSelectedFtoP;33NewBuildingAsShapeFtoP",
BuildingsFtoPMerge35, None)
```

```
global BuildingsFtoPMergeAsDescending36
BuildingsFtoPMergeAsDescending36 =
"C:\\Kod\\Results\\36BuildingsFtoPMergeAsDescending.shp"
arcpy.Sort_management (BuildingsFtoPMerge35,
BuildingsFtoPMergeAsDescending36, "NEAR_DIST DESCENDING", "UR")
arcpy.AddField_management (BuildingsFtoPMergeAsDescending36, "List",
"SHORT")
arcpy.CalculateField_management (BuildingsFtoPMergeAsDescending36, "List",
"!FID!", "PYTHON")
```

```
arcpy.JoinField_management(SkylineIntPointAsCoordTrans32, "MPATCH_OID",
NewBuildingAsShapeFtoP33, "FID", "NEAR_DIST")
arcpy.JoinField_management (SkylineIntPointAsCoordTrans32, "MPATCH_OID",
NewBuildingAsShapeFtoP33, "FID", "BuildingID")
```

```
arcpy.AddField_management (SkylineIntPointAsCoordTrans32, "Color",
"SHORT")
arcpy.CalculateField_management (SkylineIntPointAsCoordTrans32, "Color", "3",
"PYTHON")
```

```
NewSkylinePointsMerge37 = "C:\\Kod\\Results\\37NewSkylinePointsMerge.shp"
arcpy.Merge_management
("25PointsMergeAsCoordTrans;32SkylineIntPointAsCoordTrans",
NewSkylinePointsMerge37, None)
```

```
arcpy.JoinField_management (NewSkylinePointsMerge37, "BuildingID",
BuildingsFtoPMergeAsDescending36, "BuildingID", "List")
```

```
SilhouetteWithoutColor38 = "C:\\Kod\\Results\\38SilhouetteWithoutColor.shp"
arcpy.MinimumBoundingGeometry_management (NewSkylinePointsMerge37,
SilhouetteWithoutColor38, "", "LIST", "List;Color", "MBG_FIELDS")
```

```
arcpy.JoinField_management (SilhouetteWithoutColor38, "FID",
NewSkylinePointsMerge37, "List_1", "BuildingID")
```

```
arcpy.MakeFeatureLayer_management (SilhouetteWithoutColor38, "building_lyr")
```

```
arcpy.SelectLayerByAttribute_management ("building_lyr",
"NEW_SELECTION", "")
mxd = arcpy.mapping.MapDocument ("CURRENT")
df = arcpy.mapping.ListDataFrames (mxd) [0]
df.zoomToSelectedFeatures ()
arcpy.SelectLayerByAttribute_management("building_lyr",
"CLEAR_SELECTION")
```

EK 1'in devamı

```

    Silhouette39 = "C:\\Kod\\Results\\39Silhouette_" +
str(time.strftime("%d%m%Y_%H%M%S")) + ".lyr"
    arcpy.ZRenderer_stats (SilhouetteWithoutColor38, "Color", Silhouette39)

    SilhouetteAsSvg40 = "SVG,C:\\Kod\\Results\\40SilhouetteAsSvg.svg"
    arcpy.QuickExport_interop (Silhouette39, SilhouetteAsSvg40)

app = QtGui.QApplication (sys.argv)

window = QtSvg.QSvgWidget ("C:\\Kod\\Results\\40SilhouetteAsSvg.svg")
window.setWindowTitle ("Silhouette View of New Building")

window.setGeometry (0, 0, 800, 300)
window.move (400,200)

window.show ()

class Protected_Building (object):
    """Implementation for Kod_addin.combobox_3 (ComboBox)"""
    def __init__(self):
        self.items = [ ]
        self.editable = True
        self.enabled = False
        self.dropdownWidth = 'WWWWWWWWWWWWWWWWWW'
        self.width = 'WWWWWWWWWW'

    def onSelChange (self, selection):
        button_2.enabled = True
        global ProtectedBuilding41
        ProtectedBuilding41 = arcpy.mapping.ListLayers("", selection)[0]
        global ID
        rows = arcpy.SearchCursor (ProtectedBuilding41)
        row = rows.next ()
        while row:
            ID = row ()
            row = rows.next ()
        arcpy.RefreshActiveView ()

    def onFocus (self, focused):
        if focused:
            self.mxd = arcpy.mapping.MapDocument ("CURRENT")
            layers = arcpy.mapping.ListLayers ("", "BuildingID_*")
            self.items = [ ]
            for layer in layers:

                if layer.isFeatureLayer:

```

EK 1'in devamı

```
if (arcpy.Describe(layer).shapeType == "MultiPatch"):
    self.items.append (layer.name)
```

class Analysis (object):

```
    """Implementation for Kod_addin.button_2 (Button)"""
    def __init__(self):
        self.enabled = False
        self.checked = False

    def onClick (self):
        list = [ ]
        fields = ["Xnew"]
        with arcpy.da.SearchCursor (SkylineIntPointAsCoordTrans32, fields) as cursor:
            for row in cursor:
                list.append (row [0])
            list.sort ()
            Xymin = list [0]

            SkylineIntPointAsCoordTransAsDescending32 =
"C:\\Kod\\Results\\32SkylineIntPointAsCoordTransAsDescending.shp"
            arcpy.Sort_management
(SkylineIntPointAsCoordTrans32,SkylineIntPointAsCoordTransAsDescending32, "Xnew
DESCENDING", "UR")

            list= [ ]
            fields = ["Xnew"]
            with arcpy.da.SearchCursor (SkylineIntPointAsCoordTransAsDescending32, fields)
as cursor:
                for row in cursor:
                    list.append (row [0])
                Xymax = list [0]

            global ProtectedBuildingAsCoordTrans42
            ProtectedBuildingAsCoordTrans42 =
"C:\\Kod\\Results\\42ProtectedBuildingAsCoordTrans.shp"
            text = "\"BuildingID\" = " + str(ID)
            arcpy.Select_analysis
(PointsMergeAsCoordTrans25,
ProtectedBuildingAsCoordTrans42, text)

            list= [ ]
            fields = ["Xnew"]
            with arcpy.da.SearchCursor (ProtectedBuildingAsCoordTrans42, fields) as cursor:
                for row in cursor:
                    list.append (row [0])

            list.sort ()
            Xkmin = list [0]
```

EK 1'in devamı

```

ProtectedBuildingAsCoordTransAsDescending43                                     =
"C:\\Kod\\Results\\43ProtectedBuildingAsCoordTransAsDescending.shp"
    arcpy.Sort_management
(ProtectedBuildingAsCoordTrans42,ProtectedBuildingAsCoordTransAsDescending43,
"Xnew DESCENDING", "UR")

list = [ ]
fields = ["Xnew"]
with arcpy.da.SearchCursor (ProtectedBuildingAsCoordTransAsDescending43,
fields) as cursor:
    for row in cursor:
        list.append (row [0])
Xkmax = list [0]

parameter = 0

if Xkmin < Xymin < Xkmax:
    parameter = 0
else:
    if Xkmin < Xymax < Xkmax:
        parameter = 0
    else:
        if Xymin < Xkmin < Xymax:
            parameter = 0
        else:
            if Xymin < Xkmax < Xymax:
                parameter = 0
            else:
                parameter = 0 + 1

if parameter == 0:
    Skyline44 = "C:\\Kod\\Results\\44Skyline.shp"
    arcpy.Skyline_3d (ObservationPoint1, Skyline44, "", "1000 Meters", "0 Meters",
[ProtectedBuilding41], "CONVEX_FOOTPRINT", "fromA", "toA", "1", "100 Meters",
"NO_SEGMENT_SKYLINE", "100", "VERTICAL_ANGLE",
"SKYLINE_MAXIMUM", "NO_CURVATURE", "NO_REFRACTION", "0.13", "0",
"NO_CREATE_SILHOUETTES")

    SkylineIntPoint45 = "C:\\Kod\\Results\\45SkylineIntPoint.shp"
    arcpy.Intersect3DLineWithMultiPatch_3d (Skyline44, ProtectedBuilding41,
"IDS_ONLY", SkylineIntPoint45, "")
    arcpy.AddXY_management (SkylineIntPoint45)

    SkylineIntPointAsDescending46                                             =
"C:\\Kod\\Results\\46SkylineIntPointAsDescending.shp"
    arcpy.Sort_management (SkylineIntPoint45,SkylineIntPointAsDescending46,
"POINT_Z DESCENDING", "UR")

```

EK 1'in devamı

```

list = [ ]
fields = ["POINT_Z"]
with arcpy.da.SearchCursor (SkylineIntPointAsDescending46, fields) as cursor:
    for row in cursor:
        list.append (row [0])
maxKot = list [0]

global NewBuildingAsShapeFVtoP47
NewBuildingAsShapeFVtoP47 =
"C:\\Kod\\Results\\47NewBuildingAsShapeFVtoP.shp"
arcpy.FeatureVerticesToPoints_management (NewBuildingAsShape29,
NewBuildingAsShapeFVtoP47, "ALL")
arcpy.AddXY_management (NewBuildingAsShapeFVtoP47)

list = [ ]
fields = ["POINT_Z"]
with arcpy.da.SearchCursor (NewBuildingAsShapeFVtoP47, fields) as cursor:
    for row in cursor:
        list.append (row [0])
list.sort ()
minKot = list [0]

binaYuksekligi = math.ceil (maxKot - minKot)
binaKatSayisi = math.ceil (binaYuksekligi/3-0.5)

pythonaddins.MessageBox (
'Yeni binanın korunan bina silüetini bozmaması için maximum yüksekliği = ' +
str(binaYuksekligi) + "m" + "\n\n'+
'Yapılabilecek maximum bina katsayısı = ' + str(binaKatSayisi) + "\n\n',
"Silüet Analiz Raporu", 0)

else:
message 3 = "Yeni bina korunan binanın silüetini bozmuyor."
pythonaddins.MessageBox (message 3, "UYARI", 0)

```

ÖZGEÇMİŞ

1988 yılında Rize ili Ardeşen ilçesinde doğdu. İlköğrenim eğitimini Yavuz Selim İlköğretim Okulu'nda ortaöğretim eğitimini ise Ardeşen Lisesi'nde tamamladı. 2007 yılında Karadeniz Teknik Üniversitesi, Gümüşhane Mühendislik Fakültesi, Jeodezi ve Fotogrametri Mühendisliği Bölümü'nü kazandı. 2011 yılı Haziran ayında bölümden mezun oldu. 2011-2012 eğitim-öğretim yılında Karadeniz Teknik Üniversitesi, Yabancı Diller Yüksek Okulu'nda eğitime başladı ve 2012 yılı Haziran ayında dil eğitimini tamamladı. 2012-2013 eğitim-öğretim yılında Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Harita Mühendisliği Anabilim dalında Yüksek Lisans programına başladı. 2013 yılı Aralık ayında Artvin Çoruh Üniversitesi, Harita Mühendisliği Bölümüne Araştırma Görevlisi olarak atandı ve halen bu görevine devam etmektedir. İngilizce bilmektedir.