

Yinelemeli Geribildirim ile 3B Yüz İmgesi Oluřturma

Abdullah Taha Arslan

DOKTORA TEZİ

Elektrik Elektronik Mühendisliđi Anabilim Dalı

Eylül 2019

Reconstructing 3D Faces with Iterative Backpropagation

Abdullah Taha Arslan

DOCTORAL DISSERTATION

Department of Electrical and Electronics Engineering

September 2019

Yinelemeli Geribildirim ile 3B Yüz İmgesi Oluřturma

Abdullah Taha Arslan

Eskiřehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmelięi Uyarınca
Elektrik Elektronik Mühendislięi Anabilim Dalı
Telekomünikasyon-Sinyal İşleme Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Erol SEKE

Eylül 2019

ONAY

Elektrik Elektronik Mühendisliđi Anabilim Dalı DOKTORA öđrencisi Abdullah Taha Arslan'ın DOKTORA tezi olarak hazırladıđı ”**Yinelemeli Geribildirim ile 3B Yüz İmgesi Oluřturma**” bařlıklı bu alıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđiyle kabul edilmiřtir.

Danıřman : Do. Dr. Erol SEKE

İkinci Danıřman : —

Doktora Tez Savunma Jürisi:

Üye : Do. Dr. Erol SEKE

Üye : Dr. Öğr. Üyesi Nihat ADAR

Üye : Dr. Öğr. Üyesi Muammer AKÇAY

Üye : Prof. Dr. Atakan DOĞAN

Üye : Prof. Dr. Rifat EDİZKAN

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve
..... sayılı kararıyla onaylanmıřtır.

Prof. Dr. Hürriyet ERŐAHAN
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Erol SEKE danışmanlığında hazırlamış olduğum “**Yinelemeli Geribildirim ile 3B Yüz İmgesi Oluşturma**” başlıklı tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 16/09/2019

Abdullah Taha Arslan

ÖZET

Üç boyutlu görüntü işleme ve üç boyutlu modellerin elde edilmesi, bilgisayarla görü alanının önemli bir alt kolunu oluşturur. Bir nesneye ait üçüncü boyut bilgisi, nesne algılama ve nesne tanıma çözümlerinde sonucu iyileştirici ipuçları sağlar. Özellikle insan yüzü ile ilgili yapılan çalışmalarda 3-boyutlu görüntüler ve modeller aranan bir bilgi türüdür. Fakat, derinlik bilgisi 2-boyutlu imgelerde yer almadığı için, bu imgelerden yola çıkarak derinliğin kestirimi araştırmacıları uzun senelerdir üzerinde çalışmaya sevk eden zorlu bir konudur. 2-boyutlu imgelerden 3-boyutlu bilgiye erişme probleminin çözümleri uzun seneler zarfında farklı dallara ayrılmıştır. Sadece tek bir imgede yer alan nesnenin derinliğinin kestirimi en temel ve en eski problemi oluşturur. Bu alanda, çok sayıda teknik, algoritma ve yaklaşım ortaya konmasına rağmen tam anlamıyla çözüme ulaşıldığı söylenemez. Her bir yöntemin kendine özgü zorlukları ve ek kısıtları söz konusudur. Bu tez çalışmasında, bu alanda ortaya konulan en temel yöntemler araştırılmış ve test edilmiştir. Ayrıca, son yıllarda hızla gelişen Yapay Zekâ ve özellikle Derin Öğrenme tekniklerine dayalı bir çözüm geliştirilmiş ve önerilmiştir. Önerilen yöntem, önceki yöntemlere kıyasla daha genel bir çözüm ortaya koymakta ve ek kısıtlar konulması gerekmemektedir. Ayrıca, ortaya konulan yöntemin diğer yaklaşımlarla nicel kıyaslanması yapılmış ve onlara göre daha başarılı sonuçlar verdiği ortaya konulmuştur.

Anahtar Kelimeler: bilgisayarla görü, 3-boyutlu görüntü işleme, derinlik kestirimi, tonlamadan-şekil, yapay zekâ, derin öğrenme.

SUMMARY

Three dimensional image processing and reconstructing three dimensional models are important branches of computer vision field. An object's third dimensional information provides cues that improve object detection and object recognition solutions. Three dimensional images and models are important types of information, especially in the researches regarding human faces. However, since 2-dimensional images do not convey depth information, researchers have been working on the difficult problem of depth estimation from these images for a long time. The solutions to the problem of recovering third dimensional information from 2-dimensional images fall into several categories for this long period. Estimating the depth of an object given in a single image is the primary and the oldest problem. Despite of the fact that a vast amount of techniques, algorithms and approaches has been proposed in this field, it is fair to say that a definitive solution has not been reached. Each method has its own difficulties, requirements, and additional constraints. Major methods proposed in this field so far are investigated and tested in this work. In addition to this, a solution based on Artificial Intelligence and especially Deep Learning techniques has been developed and proposed. The proposed method introduces a more generic solution compared to traditional methods and it does not require employment of additional constraints. The proposed method is also quantitatively compared with others and it is found to be producing more successful outputs.

Keywords: computer vision, 3-dimensional image processing, depth estimation, shape-from-shading, artificial intelligence, deep learning.

TEŐEKKÜR

Uzun süren doktora çalışmam sırasında bana her zaman destek olan çok sevgili anneme, babama ve ablama teşekkürlerimi sunarım.

Çalışmam esnasında beni yönlendiren ve yardımlarını esirgemeyen danışman hocam Doç. Dr. Erol Seke'ye teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ	xii
ÇİZELGELER DİZİNİ	xv
SİMGELER VE KISALTMALAR DİZİNİ	xvi
1. GİRİŞ VE AMAÇ	1
2. LİTERATÜR ARAŞTIRMASI	3
3. TEORİK BİLGİ	8
3.1. Görüntüden Sayısal İmgenin Oluşumu	8
3.1.1. Lambert Kanunu ve doğrusal Lambert nesneleri	8
3.1.2. Küresel harmonikler ve Lambert yansıma modeli	9
3.2. Kamera Modelleri	10
3.2.1. İğne delikli kamera modeli	11
3.3. Projeksiyon ve Dönüşümler	12
3.3.1. Projektif dönüşüm	14
3.3.2. Afin Dönüşüm	15
3.3.3. Benzerlik dönüşümü	15
3.3.4. İsoometri ya da Öklid dönüşümü	16
3.4. Öznitelik Noktaları	16
3.4.1. Öznitelik seziciler	16
3.4.2. Öznitelik çıkarma (Özütleme) ve öznitelik betimleyiciler	16
3.4.3. Eşleştirme	17
3.5. Stereo Görüş	17
3.5.1. Epipolar geometri	17
3.5.2. Temel matris	18
3.5.3. Hareketten yapı çıkarımı (<i>Structure from motion</i>)	19

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
3.6. Bir Yüz İmgesi ve Referans Yüze Ait Derinlik Haritası Kullanarak Derinlik Bilgisinin Bulunması	19
3.6.1. Aydınlatma (ışık kaynağı) vektörünün bulunması	20
3.6.2. Derinlik bilgisinin bulunması	22
3.6.3. Albedo bilgisinin bulunması	26
3.6.4. İkinci dereceden küresel harmonikler cinsinden açılım ve optimizasyon çözümü	27
3.7. Bilinen Derinlik Haritasından Albedo Bilgisinin Elde Edilmesi	32
3.8. Aydınlatma Şartlarının Kestirimi Yöntemleri	34
3.8.1. Yöntemler	35
3.8.1.1. Pentland, 1982	35
3.8.1.2. Lee ve Rosenfeld, 1985	36
3.8.1.3. Brooks ve Horn, 1985	36
3.8.1.4. Zheng ve Chellappa, 1991	37
3.8.1.5. Chojnacki, Brooks, Gibbins, 1994	37
3.8.1.6. Attick, Griffin, Redlich, 1996	38
3.8.1.7. Zhao, 1999	38
3.8.2. Yöntemlerin karşılaştırılmaları	38
3.8.2.1. Eğiklik açısı kestirimi	39
3.8.2.2. Verev açısı kestirimi	41
3.9. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme	42
3.9.1. Perseptron ve yapay sinir ağları	42
3.9.2. Aktivasyon fonksiyonları	44
3.9.3. Evrimsel ağlar	45
3.9.4. Otokodlayıcılar	45
3.9.5. Üretken ağlar	46
4. MATERYAL ve YÖNTEM	48
4.1. Uygulama Veritabanları	48
4.2. Derin Öğrenme Tekniklerine Dayanan Derinlik Kestirici Yeni Bir Yöntemin Geliştirilmesi ve Önerilmesi	53
4.2.1. Üreteç ağ yapıları	56
4.2.2. Ayırtaç ağ yapıları	58
4.2.3. Kayıp fonksiyonları	59

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
4.2.4. Eniyileştirme algoritmaları	60
4.2.5. Parametrelerin seçimi	60
4.2.6. Deneysel kurgu	61
5. BULGULAR ve TARTIŞMA	66
5.1. Aydınlatma Şartlarının Kestirimi Yöntemlerine İlişkin Bulgular	66
5.1.1. Uygulama ve deneyler	66
5.1.2. Değerlendirme	68
5.2. Kemelmacher-Shlizerman ve Basri'nin Yöntemine İlişkin Bulgular	70
5.3. Önerilen Yönteme İlişkin Bulgular ve Tartışma	77
5.3.1. Yakınsama sorunları ve öğrenme oranı optimizasyonu	79
5.3.2. Önerilen yöntemin önceki yöntemlerle karşılaştırılması	80
5.3.2.1. Önerilen yöntemin otokodlayıcı yapılarla karşılaştırılması	80
5.3.2.2. Önerilen yöntemin SFS yöntemleri ile karşılaştırılması .	81
6. SONUÇ VE ÖNERİLER	89
KAYNAKLAR DİZİNİ	91
EK AÇIKLAMALAR	104
Ek Açıklamalar - A: Doğrusal Olmayan Optimizasyon için MATLAB Betiği . . .	105
Ek Açıklamalar - B: Kod Örnekleri	110
Ek Açıklamalar - C: Derin Öğrenme Tabanlı Yönteme İlişkin Kod Örnekleri . . .	116
Özgeçmiş	122

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 3-Boyutlu Rekonstrüksiyon Yöntemlerini Sınıflayan Bir Taksonomi	4
3.1 İğne delikli kamera modeli.	12
3.2 Epipolar geometri.	18
3.3 Tek bir iki-boyutlu imgeden derinlik bilgisinin kestirimi için Kemelmacher-Shlizerman ve Basri'nin ortaya koyduğu yöntemin (Bölüm 3.6) uygulanmasının genel akışı. Bu algoritmada, test girdi imgesinin yanı sıra referans bir bireye ait derinlik ve albedo haritaları da kullanılmaktadır.	28
3.4 Bir noktasal aydınlatma kaynağı doğrultusu için azimuth ve yükselti parametreleri.	39
3.5 Bir gri seviye sayısal imge kesitinde yer alan 8 farklı değişim yönü.	40
3.6 Bir doğrusal ayırtaç fonksiyonunun sinir ağı şeklinde gösterimi. (a) 2-sınıflı durum, (b) çok sınıflı durum.	43
3.7 Otokodlayıcı yapısı.	45
4.1 Texas Veritabanı'nda yer alan örnek bireylere ait portre görüntüleri ve derinlik haritaları.	50
4.2 Bosphorus Veritabanı'nda yer alan bireylere ait örnek portre görüntüleri ve derinlik haritaları.	51
4.3 Referans bir bireye ait imgeleri kullanarak derinlik kestirimi yapan ve Bölüm 3.6'da anlatılan algoritmanın farklı adımlarında kullanılmak üzere hazırlanan referans ortalama imgeler ve derinlik haritalarından elde edilmiş modelin anlık görüntüsü. a) Texas Veritabanı'nda yer alan 116 bireye ait portre imgelerinin ortalaması alınarak oluşturulan referans gri-seviye portre imgesi, b) Bu ortalama portreye karşılık gelen ortalama derinlik haritası kullanılarak oluşturulan yüzeyin anlık bir görüntüsü, c) Bosphorus Veritabanı'nda yer alan 105 bireye ait portre imgelerinin ortalaması alınarak oluşturulan referans gri-seviye portre imgesi, d) Bu ortalama portreye karşılık gelen ortalama derinlik haritası kullanılarak oluşturulan yüzeyin anlık bir görüntüsü.	52
4.4 2-boyutlu girdi imgeleri ve derinlik haritaları ile eğitilecek Koşullu Çekişmeli Üreteç Ağ'ın genel yapısı.	55
4.5 Derinlik kestirici üreteç ağ olarak eğitilecek olan farklılaştırılmış bir U-Net yapısı. <i>Conv2D</i> iki-boyutlu evrişimsel işlev katmanlarını, <i>BN batch normalization</i> işlemini, <i>DeConv2</i> iki-boyutlu ters evrişimsel işlev katmanlarını, <i>LReLU Leaky ReLU</i> aktifleştirme işlevini göstermektedir. Girdi imgesi tensör boyutu (3,256,256)'dır.	57

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
4.6 Öğrenme oranını her iterasyonda sabit tutmak yerine belli bir algoritmaya dayalı olarak azaltmak, eğitimin daha kararlı olmasını sağlayabilir. Öğrenme oranının çizelgelenmesinde kullanılan fonksiyonlar. (a) Tek faktörlü azaltma fonksiyonu, belli iterasyon sayısından sonra öğrenme oranı yarıya düşürülmekte, (b) adımların uzunluğu değişken tutulabilir, (c) polinom denkleme göre, ve (d) kosinüs fonksiyonu.	62
4.7 Veri setinde yer alan bir bireye ait çoğaltılmış imgeler. Soldan sağa: a) Beyaz dengelenmiş, b) Saat yönünde döndürülmüş, c) Ters saat yönünde döndürülmüş, d) Gausyen bulanıklaştırılmış, e) Orijinal imge.	63
5.1 Üç boyutlu sentetik bir küre modelinin farklı açılardan aydınlatılması sonucunda elde edilen iki boyutlu imgeler: Verev açısı 0° 'den 90° 'ye kadar değişmekte, eğiklik açısı sabit ve 15° dir.	67
5.2 Üç boyutlu Mozart modelinin farklı açılardan aydınlatılması sonucunda elde edilen iki boyutlu imgeler: Eğiklik açıları 0, 90, 180, 270, verev açıları 30 ve 75 derecede.	68
5.3 Sentetik küre imgeleri için ışık kaynağı yönünün kestirimi. (a) eğiklik açısı kestirimlerini gerçek eğiklik açısı değerlerine karşı, (b) verev açısı kestirimini ($\phi = 45^\circ$) durumunda her eğiklik açısına karşı, (c) verev açısı kestirimini gerçek verev açısı değerlerine karşı, ve (d) eğiklik açısı kestirimini ($\tau = 45^\circ$) durumunda her verev açısına karşı göstermektedir.	71
5.4 Mozart imgeleri için ışık kaynağı yönünün kestirimi. (a) eğiklik açısı kestirimlerini gerçek eğiklik açısı değerlerine karşı, (b) verev açısı kestirimini ($\phi = 45^\circ$) durumunda her eğiklik açısına karşı, (c) verev açısı kestirimini gerçek verev açısı değerlerine karşı, ve (d) eğiklik açısı kestirimini ($\tau = 45^\circ$) durumunda her verev açısına karşı göstermektedir.	72
5.5 Texas Veritabanı'nda yer alan 4 bireye ait albedo hesaplama yöntemi ile elde edilmiş albedo çıktı örnekleri (Dosya numaraları: 93, 160, 161, 268). a) Girdi test imgesi (portre görüntüsü), b) girdi test imgesinin derinlik haritası, c) albedo algoritması ile elde edilen albedo haritası. s aydınlatma vektörü her birey için imgelerin altında verilmiştir.	73
5.6 Kemelmacher-Shlizerman ve Basri'nin yönteminin ikinci adımında kullanılan regülasyon katsayısı λ_1 'in algoritma sonucuna etkisini gösteren çıktı yüzey görüntüleri. λ_1 değerleri sırasıyla, 3, 5, 10, 15, 20, 25'tir. Son yüzey görüntüsü referans imgeye aittir.	75

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
5.7 Tek bir imge ve referans bireye ait derinlik haritası ile derinliğin kestirimi algoritmasının Texas Veritabanı'nda yer alan bireylere ait imgelerle çalıştırılması sonucu elde edilen örnek çıktılar. a) İki boyutlu test girdi portre imgesi, b) Gerçek derinlik haritası ile oluşturulmuş yüzeyin bir açıdan alınmış anlık görüntüsü, c) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeyin bir açıdan alınmış anlık görüntüsü, d) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeye dokunun eklemesi sonucu elde edilen anlık görüntü.	77
5.8 Tek bir imge ve referans bireye ait derinlik haritası ile derinliğin kestirimi algoritmasının test veritabanında yer almayan üç rasgele bireye ait portre imgeleri ile çalıştırılması sonucunda elde edilen çıktılar. a) İki boyutlu girdi test imgesi, b) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeyin bir açıdan alınmış anlık görüntüsü, c) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeye girdi imgesindeki dokunun eklenmesi sonucu elde edilen anlık görüntü	83
5.9 Önerilen WGAN derinlik kestirici ağ için test çıktıları. a) İki boyutlu girdi test imgesi, b) Önerilen yöntemle elde edilen derinlik haritası kullanılarak rekonstrüksiyonu yapılan yüzeyin görüntüsü, c) Referans gerçek derinlik haritası kullanılarak rekonstrüksiyonu yapılan yüzeyin görüntüsü, d) ve e) Önerilen yöntem çıktı rekonstrüksiyonlarının doku kaplamalı iki farklı görme açısından alınmış anlık görüntüleri. (Test imgeleri veritabanlarından gelmektedir, fakat eğitim aşamasında kullanılmamışlardır.)	84
5.10 İnternet üzerinden indirilen portre imgeleri için görsel inceleme amaçlı üretilmiş test çıktıları. a) İki boyutlu girdi test imgeleri, b) Önerilen yöntemle elde edilmiş çıktı derinlik haritaları, c) Derinlik haritası kullanılarak rekonstrüksiyon edilmiş yüzeyin görüntüsü, d) ve e) doku kaplamalı rekonstrüksiyon edilmiş ve iki farklı görme açısından alınmış anlık görüntüler.	85
5.11 Üreteç ağının kayıp fonksiyonu çıktısının her epok için çizilmesini gösteren şekil.	86
5.12 a) WGAN, b) SCAE ve c) SFS yöntemleri için RMS hatalarının karşılaştırılması. Ortalama RMS değerleri dikey kesikli çizgilerle belirtilmiştir.	87
5.13 a) WGAN, b) SCAE ve c) SFS yöntemleri için çıktı derinlik haritaları ve bunların gerçek değerlerle olan farkları.	88

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1 Geometri türlerinin karşılaştırılmaları: dönüşüm türleri ve dönüşümler sonucu değişikliğe uğramayan özellikler.	13
3.2 Derin öğrenmede sıkça kullanılan aktifleştirme fonksiyonları.	44
4.1 Geliştirilen ve önerilen yöntem ile eğitilecek olan ağ yapısı için kullanılan Texas ve Bosphorus veritabanlarında yer alan bireylerin sayısını, her bir veritabanı için eğitim amacıyla ayrılan imgelerin sayısını ve bu imgeler üzerinde gerçekleştirilen çoğaltma işlemleri sonucunda oluşan yeni imge sayılarını gösteren çizelge.	64
4.2 Üreteç ağ yapısı, ayırtaç ağ yapısı ve kayıp fonksiyonu için farklı seçenekler yer almaktadır. Bunlar arasından yapılan seçimlerle kurulan GAN konfigürasyonlarını gösteren çizelge.	64
5.1 Aydınlatma vektörleri ve albedo haritaları tahmin edildikten sonra 5.1 eşitliği kullanılarak elde edilen çıktı imgeleri ile orijinal portre görüntüleri arasındaki farkları gösteren çizelge. Test imgeleri Şekil 5.5'te gösterilmektedir.	71
5.2 Tek bir imge ve referans bireye ait derinlik haritası ile derinliğin kestirimi algoritması ile Texas Veritabanı'nda yer alan bireylere ilişkin çıktıların Yüzde Ortalama Mutlak Hata olarak ölçülen oranlar. Ortalama hata 6.45 ve standart sapma 3.27'dir.	76
5.3 Önerilen yöntemle derinlik kestirici olarak tasarlanan ağların yapılarını, kayıp fonksiyonlarını ve testlerin sonucunda ölçülen başarımlarını gösteren çizelge. 78	78

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

ρ	Albedo (Parlaklık Yüzdesi)
λ_1	Derinlik için regülasyon katsayısı
λ_2	Albedo için regülasyon katsayısı
τ	Azimuth açısı
ϕ	Verev (<i>slant</i>) açısı
K	Kamera kalibrasyon matrisi
R	Rotasyon matrisi
t	Öteleme vektörü

Kısaltmalar

PIE	Poz, aydınlatma, yüz ifadesi (<i>Pose, Illumination, Expression</i>)
SFS	Tonlamadan-Şekil (<i>Shape From Shading</i>)
FS	Fotometrik Stereo
GPU	Grafik İşlem Birimi (<i>Graphical Processing Unit</i>)
CNN	Evrişimsel Sinir Ağı (<i>Convolutional Neural Network</i>)
SVD	Tekil Değer Ayrışımı (<i>Singular Value Decomposition</i>)
MSE	Ortalama Kare Hata (<i>Mean Squared Error</i>)
RMSE	Kök Ortalama Kare Hata (<i>Root Mean Square Error</i>)
PSNR	Doruk Sinyal Gürültü Oranı (<i>Peak signal-to-noise ratio</i>)
PCA	Temel Bileşenler Analizi (<i>Principal Component Analysis</i>)
d.o.f.	Serbestlik derecesi (<i>degree of freedom</i>)
ML	En Büyük Olabilirlik Kestirimi (<i>Maximum Likelihood</i>)
MSER	Azami Ölçüde Kararlı Uç Bölgeler (<i>Maximally Stable Extremal Regions</i>)
SIFT	Ölçekten Bağımsız Öznitelik Dönüşümü (<i>Scale-Invariant Feature Transform</i>)
MLP	Çok Katmanlı Algılayıcı (<i>Multilayer Perceptron</i>)
GAN	Üretken Çekişmeli Ağ (<i>Generative Adversarial Network</i>)
CGAN	Koşullu Üretken Çekişmeli Ağ (<i>Conditional Generative Adversarial Network</i>)

1. GİRİŞ VE AMAÇ

Günümüzde, görüntü işleme teknikleri giderek artan oranda 3-boyutlu imgeler üzerine kaymaktadır. Bu yaklaşımın, özellikle yüz tanıma, vb. işlemlerde poz ve aydınlatma değişikliklerinden kaynaklanan zorlukları aşabilme potansiyeli dikkat çekmektedir. Bu nedenle, bu alanda yapılan araştırmaların ve yayınlanan çalışmaların sayısı giderek artmaktadır.

İnsanlar, bir 2-boyutlu görüntüyü gördüklerinde üçüncü boyut bilgisini çok kısa sürede algılayabilmektedir. Görülen imge daha önce görmedikleri bir cisme bile ait olsa, 2-boyutlu görüntü beyne ulaştınca, benzer cisimlere ait daha önce oluşturulmuş veri kullanılarak bu yeni cisme ait üçüncü boyut bilgisi beyin tarafından oluşturulabilmektedir. İnsanlar için bu, bilinçaltında gerçekleşen ve farkına varmadan yapılan bir işlem olsa da, bir bilgisayarın aynı işlemi simüle edebilmesi için cisme ait ayırt edici özelliklerin araştırılması gerekmektedir. Bu yüzden, 2-boyutlu bir görüntüden 3-boyutlu şekil bilgisinin oluşturulması işlemi kompleks ve sorunlu bir durumdur.

Herhangi bir cismin veya bir insan yüzünün derinlik bilgisini elde etmede kullanılan bazı gelişmiş algılama cihazları ve kameralar piyasada bulunmaktadır. Bu cihazlar, lazer ışınlarını, yapılandırılmış ışık projeksiyonlarını, vb. teknikleri kullanırlar. Fakat, bu aygıtların kullanımlarının çok kolay olmaması ve kolayca erişilebilir olmayışları nedeniyle çok yaygın değillerdir. Dolayısıyla, hali hazırda elde bulunan 2-boyutlu imgelerden yola çıkarak 3-boyutlu görüntülerin elde edilmesi önem taşıyan bir konudur.

Aynı görüntüye ait fakat farklı aydınlatma şartlarında alınmış birden fazla imgenin kullanıldığı fotometrik stereo (FS) yöntemi de üzerinde çokça araştırma yapılmış bir konudur. Fakat, bu tez çalışmasında tek bir imgeden yola çıkarak derinlik bilgisinin ve böylece 3-boyutlu görüntünün tekrar oluşturulması hedeflenmektedir.

Bilgisayarla görü problemlerinin çözülmesine ve ortaya konan yöntem ve algoritmaların geliştirilmesine yönelik çalışmalar çok uzun senelerdir yapılmaktadır. Son yıllarda yaşanan en önemli yaklaşım değişikliği, bu çözümlerin *Yapay Zekâ* yöntemleri ile çözülmesi arayışlarıdır. Bunun başlıca nedenleri arasında, özellikle *Derin Öğrenme* tekniklerinin gelişmesi ve bilgisayarların grafik kartları ile matematiksel işlemlerin çok daha hızlı ve verimli şekilde yapılmasının önünün açılmasıdır.

Bu tez çalışmasının amaçlarından biri, 2-boyutlu imgelerde bulunan insan yüzlerinin derinlik haritalarını kestirmeye yönelik günümüze kadar geliştirilmiş ve ortaya konmuş teknikleri ve algoritmaları incelemek, bunlar arasında tekniğin geldiği son noktadaki uygulamaları test etmek ve bu yöntemleri karşılaştırılmaktadır. Sayısal imgelerin hangi şartlar altında olduğu ve özellikle aydınlatma şartlarının neler olduğu bilgilerinin bu alanda yapılan çalışmalar için önemli olması nedeniyle, bu şartların araştırılması ve gerektiğinde bunların da kestirimlerine yönelik tekniklerin irdelenmesi de hedeflerden birini oluşturmaktadır. Bu tez çalışmasının temel amacı ise, bilgisayarla görüş problemlerinin çözülmesinde son yıllarda yaşanan yaklaşım tarzlarını uygulayan ve özellikle Derin Öğrenme tekniklerine dayanan yeni bir derinlik kestirici yöntemin geliştirilmesi, test edilmesi, geliştirilecek olan bu yöntemin önceki yöntemlerle nicel karşılaştırılmasının yapılması, önceki yöntemlere göre varsa kaydettiği ilerlemenin ve onlara göre olan dezavantajlarının araştırılması ve ortaya konulmasıdır.

Bu tez çalışmasına kaynaklık eden araştırmaların ve yayınların sunulduğu literatür araştırması 2.Bölüm'de yer almaktadır. Aynı bölümde, iki boyutlu imgelerden üçüncü boyut bilgisinin kestirimine yönelik günümüze kadar geliştirilen yöntemlere ilişkin bilgi verilmekte ve bu yöntemleri sınıflayan bir taksonomi sunulmaktadır. Bir sonraki bölümde; görüntüden sayısal imgenin oluşumu, kamera modelleri, projeksiyon ve dönüşümler, öznitelik noktaları, stereo görüş, bir yüz imgesi ve referans bir yüze ait derinlik haritası kullanarak derinlik bilgisinin bulunduğu bir yöntem, bilinen derinlik haritası kullanarak albedo bilgisinin elde edilmesi, aydınlatma şartlarının kestirimi yöntemleri, yapay zekâ, makine öğrenmesi ve derin öğrenmedir konularında teorik bilgiler verilmektedir. Bu tez araştırmasında kullanılan materyaller ve geliştirilen yöntem, kapsamlı bir şekilde 4.Bölüm olan "Materyaller ve Yöntem"de sunulmaktadır. Bütün araştırmalar ve çalışmalar neticesinde elde edilen bulgular ve bunlara ilişkin tartışmalar 5.Bölüm'dedir. Son bölümde ise, sonuçlar tartışılmakta ve öneriler ortaya konulmaktadır.

2. LİTERATÜR ARAŞTIRMASI

İnsan yüzlerine ilişkin derinlik bilgisinin elde edilmesi ve 3-boyutlu rekonstrüksiyonu bilgisayarla görü araştırmalarının önemli bir alanını oluşturur. 3-boyut bilgisi, özellikle 2-boyutlu yüz sezimi ve tanıma işlemlerinde karşılaşılan poz, aydınlatma ve yüz ifadesi değişimlerinden (*PIE*) kaynaklı zorlukların üstesinden gelmede önemli ek faydalar sağlamaktadır (Blanz vd., 2002). Poz değişiklikleri, aydınlatma koşullarının kestirimi ve imge üzerindeki örtmeler bu problemlerden bazılarıdır. Bununla birlikte, 3-boyutlu modellerin üretilmesi ya da 2-boyutlu imgelerden rekonstrüksiyon yapılması araştırmacılar için başlı başına bir zorluk oluşturmaktadır. Bu problemin çözümü için son on yıllarda çok sayıda yaklaşım geliştirilmiş ve önerilmiştir. Fakat, önerilen yöntemlerin her birinin kendine has güçlükleri olduğu da görülmektedir.

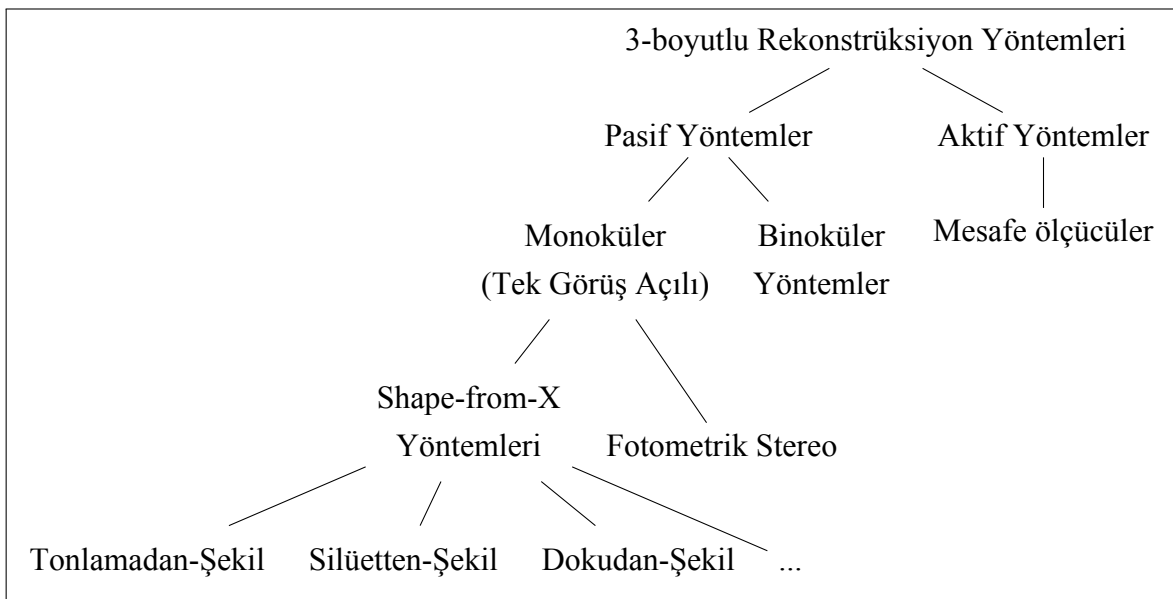
Tek imgeden 3B görüntü elde etme konusunda yapılan çalışmalar farklı yaklaşımlarla devam etmektedir. Yüz görüntüsünün elde edilmesinde bir biçimlenebilir model (*morphable model*) kullanılması ve uzun süreden beri bilinen tonlamadan-şekil (*shape-from-shading*) tekniklerinin geliştirilmeleri (Horn, 1970) bu yönlerden ikisidir. Farklı yaklaşımları gösteren bir taksonomik sınıflandırma Şekil 2.1’de gösterilmiştir.

Model tabanlı çalışmalarda genel bir 3-boyutlu yüz modeline ya da 3-boyutlu yüzlerden oluşan bir veri setine ait şekil ve doku bilgileri ön bilgi olarak kullanılmaktadır. Bu tarz çalışmalarda, genel olarak eldeki 3-boyutlu imge veritabanından yüz sınıflarına ait bir taban oluşturulur ve verilen yeni bir görüntüdeki yüze ait şekil ve doku, bir uyarılma algoritmasıyla bu taban içinde ifade edilir. Böylece oldukça tatmin edici 3-boyutlu yüz görüntüleri oluşturulabilir. İteratif doğası gereği bu yöntemler sayısal olarak yüksek hesaplama gerektirir. Bazı durumlarda da, verilen görüntüdeki bazı noktaların 3-boyutlu model üzerinde elle işaretlenmesi istenmektedir. Bu sınırlandırmalar, model tabanlı bu yöntemleri gerçek zamanda çalışması istenen durumlar için kullanışsız kılmaktadır. Yine de, bu yöntemlerin yüz tanıma gibi uygulamalarda başarı oranını artırdığı görülmüştür (Leo ve Manimegalai, 2011).

Tonlamadan-şekil (SFS) yöntemi, "*shape-from-X*" diye adlandırılan ve verilen bir imgeden farklı ipuçları kullanarak şekil bilgisine ulaşmaya çalışan pasif yöntemlerden bir tanesidir. Bunlar arasında, SFS dışında dokudan-şekil (*shape-from-texture*), konturdan-şekil (*shape-from-contour*) yöntemleri bulunur. Ayrıca birden fazla imgenin kullanıldığı stereo

yöntemi, odaktan-şekil (*shape-from-focus*), hareketten-şekil (*shape-from-motion*) ve silüetten-şekil (*shape-from-silhouettes*) yöntemleri de vardır (Leo ve Manimegalai, 2011).

SFS yöntemi de, 4 gruba ayrılabilir: 1) Bir enerji fonksiyonunu minimize etmeye çalışan minimizasyon yöntemi, 2) Şekil bilgisini, bazı tekil yüzey noktalarından oluşturulan bir kümeden bütün bir imgeye dağıtan yayılım yöntemi (*propagation*), 3) Yüzey türü varsayımından hareketle şekil bilgisine ulaşmaya çalışan yerel yaklaşımlar, 4) Yansıma haritasının doğrusallaştırılmasına dayanan bir çözümü elde etmeye çalışan doğrusal yaklaşımlar. En basit şekliyle, SFS yönteminde, verilen tek bir görüntüdeki yüzeyin 3-boyutlu şekil bilgisi, yüzey piksellerinin yoğunluk değerleriyle, yüzey normalleri ve ışık kaynağının yönü arasındaki açılar ilişkilendirilerek elde edilir. Bunu yaparken de, yüzeyin yansıma özellikleri, ışık kaynağı yönü ve sınır noktalarındaki derinlik bilgisi gereklidir. Bunlar bilinse dahi, SFS problemine tekil bir çözüm bulmak zordur. Bazı ilave kısıtlara ihtiyaç duyulur (Zhang vd., 1999).



Şekil 2.1 3-Boyutlu Rekonstrüksiyon Yöntemlerini Sınıflayan Bir Taksonomi

Blanz ve Vettel, 200 kadar insanın yüzünü lazer taraması yöntemiyle 3-boyutlu olarak bir veritabanı oluşturmuştur. Bu veritabanında, her bir insan yüzünün 70,000 adet düğümden oluşan doku parametreleri bulunur. Biçimlenebilir yüz modeli, bu taramaların doğrusal bir kombinasyonundan oluşmaktadır. Böylece istenilen bir yüz görüntüsü parametrik olarak elde edilebilmektedir. Verilen bir 2-boyutlu hedef yüz görüntüsü, iteratif bir şekilde bu modelle eşleştirilir (Blanz ve Vetter, 1999). Cootes, vd.'nin çalışması, yüzlere

ait 2-boyutlu şekil ve beyazlık (albedo) bilgisini tanımlamak için eşleştirilmiş bir istatistiksel modelin oluşturulduğu Aktif Görünüm Modelleri'ne (AAM) dayanmaktaydı (Cootes vd., 2001). Blanz ve Vettel, bunu üç boyuta genişletmiştir.

Biçimlenebilir model tabanlı çalışmalar daha sonraki yıllarda 3-boyutlu yeniden oluşturma işlemini geliştirmeye devam etmişlerdir. Patel ve Smith, (2012) SFS yöntemiyle elde ettikleri yüzey yerleşim bilgisinin, bir 3-boyutlu biçimlenebilir modelin bir yüze ait görüntüye eşleştirilmesi işlemine yardım edebileceğini göstermiştir. Yerel biçimlenebilir modellerle SFS yöntemini birleştiren benzer bir çalışma da Liao, vd. (2012) tarafından gerçekleştirilmiştir. Bir diğer modelleme yöntemi istatistiksel modelledir. Bu modeller, 3-boyutlu yüz şeklini tahmin etmek için önsel (*a priori*) bilgiyi ve belli sayıdaki karakteristik noktayı bir örtüştürme yöntemi içinde kullanırlar. Bu yöntemlerden üzerinde durulmuş olan bir tanesi regülarizasyon yöntemleridir (Jiang vd., 2005; Maghari vd., 2013; Maghari vd., 2012; Blanz ve Vetter, 2002).

1970'lerde Berthold K.P. Horn tarafından ilk uygulaması yapılan SFS yöntemi (Horn, 1970), ilerleyen yıllarda farklı yaklaşımlarla geliştirilmiştir. İnsan yüzünün simetri özelliğini kullanan SFS çalışmalarına da rastlanılır (Shimshoni vd., 2000; Zhao ve Chellappa, 2000; Zhao ve Chellappa, 2001). Simetri gibi geometrik çözümlerinin yanında bir diğer çokça araştırılan yöntem de istatistiksel yaklaşımlardır (Atick vd., 1996; Dovgard ve Basri, 2004; Smith ve Hancock, 2006b).

Bir adet 3-boyutlu yüz biçimini (derinlik bilgisi) ve buna ait albedo bilgisini referans alan, aydınlatma şartları ve pozü bilinmeyen tek bir 2-boyutlu yüz görüntüsünden 3-boyut biçimini elde etmeye çalışan bir çalışma da Kemelmacher-Shlizerman ve Basri'ye aittir (Kemelmacher-Shlizerman ve Basri, 2011; Kemelmacher ve Basri, 2006). Genel olarak, tek bir imgeden şekil ve albedo bilgisini elde etmek, eksik tanımlanmış matematiksel bir problem olduğu için, regülarizasyon ifadeleri ve sınır kısıtlarını probleme eklemek gerekmektedir. Böylece aranan yüz ile referans yüz arasındaki fark azaltılmaya çalışılır. Burada kullanılan 3-boyutlu referans yüz, tek bir kişiye ait olabileceği gibi *generic* bir yüz görüntüsü de olabilir. Diğer çalışmaların aksine, bir 3-boyutlu yüz veritabanına ya da yüzdeki simetri özelliğinin kullanılmasına ihtiyaç kalmamaktadır. Işığın yansımaları küresel harmonikler cinsinden ifade edilir ve bundan dolayı geliştirilen algoritma bilinmeyen genel aydınlatma şartlarında çalışabilmektedir.

Grafik işlem birimlerine (*GPU*) dayalı hesaplama yöntemlerinin gelişmesiyle birlikte evrimsel sinir ağları (*CNN*) çoğu bilgisayarla görü probleminin çözümü için kullanılmaya başlanmıştır. Son yıllarda, özel bir ağ yapısına sahip ve *Üretken Çekişmeli Ağ* (*GAN*) adı verilen bir yaklaşım, kolay uygulanabilirliğinden dolayı çok sayıda

imgeden-imgeye dönüşüm probleminin çözümünde önerilmektedir. İmge bölütleme, süper çözünürlük ve arka plan maskeleyme, bu geliştirilen çözümlere bazı örneklerdir. Üretken Çekişmeli Ağlar, *üreteç* ve *ayırtaç* adı verilen iki ayrı ağdan oluşmaktadır (Goodfellow vd., 2014; Radford vd., 2015). Bu ağ yapıları, gerçeğe çok yakın imgeler üretebildiklerini ispatlamışlardır. GAN'ların bir türü de *Koşullu Üretken Çekişmeli Ağlar*dır (Mirza ve Osindero, 2014). Koşullu GAN'ların geliştirilme amacı, imgeden-imgeye dönüştürme problemlerinin çözümü için genel bir yaklaşım ortaya koymaktır. Koşullu bir GAN'ın koşulsuz bir ağ yapısından farkı, girdi imgelerin hem *üreteç* hem de *ayırtaç*a beslenmesidir. Koşullu GAN'ların uygulandığı bazı alanlar, arka plan maskeleyme, imge bölütleme (Li ve Shen, 2018) ve sınır çizgilerinden nesnelere üreten çalışmalardır (Isola vd., 2017).

Radford vd. (2015), *Derin Evrimsel Üretken Çekişmeli Ağ* (DCGAN) adı verdikleri bir CNN ağ sınıfı geliştirmişlerdir. Ayrıca, kendine özgü problemleri olan GAN eğitim süreçlerini dengelemek amacıyla birtakım kurallar ve yapısal öneriler ortaya atmışlardır. Mirza ve Osindero (2014), ağları ek bilgiyle besleyerek GAN yaklaşımını koşullu bir yapıya çevirmişlerdir. Isola, vd.nin (2017) çalışmasında, girdi bilgisi koşullayıcı değişken olarak alınarak çıktı kategorileri için bire bir eşlemler bire çok etiketlemelere genişletilmiştir. Böylece bu çalışmada imgeden-imgeye problemleri için genel bir ağ iskeleti oluşturma hedefi gözetilmektedir. Bu genelleştirilmiş ağ yapısı, hava fotoğraflarından haritaya, siyah-beyazdan renkliye, etiketlerden sokak görünümüne ve bina dış cephelerine, hatta el çizimlerinden çantalara, ayakkabılara ve kedilere gibi ilgi çekici çok farklı türlerde dönüşümler oluşturabilmektedir. Bu yapıda, imge oluşturan *üreteç* ağı, daha önce imge bölütleme amacıyla geliştirilmiş olan ve bir kodlayıcı-kodçözücüden oluşan U-Net ağının (Ronneberger vd., 2015), karşılık gelen çözünürlük bloklarını birbirine bağlamak için bazı *atlama* katmanları eklemek suretiyle değiştirilmiş bir türevidir.

Zhang vd. (2018), SAGAN adı verdikleri yapıda imgelerin ayrılmış bölgelerinden öznetelikleri bir araya getirmek amacıyla GAN'lara *özilgi* katmışlardır. Bu ağ yapısının Inception ve FID skorlarında en iyi sonucu ürettiği iddia edilmektedir. Inception skorlaması bir *üreteç* ağının başarımını ölçmek amacıyla iki ölçütü dikkate alır: 1) Üretilen imgelerin kalitesi, 2) Bu imgelerin çeşitliliği (Salimans vd., 2016; Szegedy vd., 2016). *Fréchet Inception Distance* (FID) skorlamada, Inception ağını kullanılarak ara katmanlardan öznetelikler alınmakta ve bu ölçümün imge çeşitliliği açısından daha iyi bir sonuç verdiği öne sürülmektedir (Heusel vd., 2017). SAGAN ağı eğitilirken, Heusel vd.nin (2017) önerdikleri üzere *üreteç* ve *ayırtaç* ağları için farklı öğrenme oranları ve güncelleme adımları seçilmektedir.

Karras, vd. (2017), eğitim süreci ilerlerken düşük çözünürlüklerden yeni katmanlar ekleyerek hem *üreteç* hem de *ayırtaç* ağını artan bir şekilde büyütme çalışmaları. Rölativistik

GAN (RGAN) ve Ortalama Rölativistik GAN (RaGAN) yapıları, verilen bilginin sahte bilgiye oranla daha gerçeğe yakın olma olasılığını kestirerek ayırtaç ağının davranışını değiştirirler (Jolicoeur-Martineau, 2018).

GAN'ların yapıları gereği kararsızlık gösterdikleri ve parametre seçimlerine çok duyarlı oldukları için eğitimlerinin kolay gerçekleşmediği belirtilmiştir (Arjovsky ve Bottou, 2017). Son birkaç sene içerisinde, yakınsamayı iyileştirmek için çok sayıda yaklaşım ve farklı GAN türleri önerilmiştir. Arjovsky, vd. (2017), olasılık dağılımları arasında çeşitli uzaklık ölçülerini incelemiş ve *Toprak-Taşıyıcı* uzaklığını, diğer adıyla Wasserstein uzaklığını GAN yapıları için önermişlerdir. WGAN (Wasserstein GAN) adı verilen bu ağ yapısının daha kararlı davrandığını göstermişlerdir. Algoritmalarında yeni bir kayıp metriği sunulmakta ve bunun orijinal GAN metriğine daha iyi bir alternatif olduğu iddia edilmektedir. Ayrıca, bu yapıyla birlikte ayırtaç ağının adı da *eleştirici* (critic) olarak değişmektedir. WGAN-GP ağ yapısı, WGAN'da yapıldığı gibi ağırlıkları kırmak yerine yeni bir gradyan maliyeti ekleyerek performansı iyileştirmeye çalışmaktadır (Gulrajani vd., 2017).

En az kareler GAN (LS-GAN) yaklaşımında ayırtaç kayıp fonksiyonu bir en az kareler yaklaşımı ile değiştirilir. Bu yöntemin daha kararlı bir sonuç ürettiği ve daha yüksek kalitede imgeler oluşturduğu iddia edilmektedir (Mao vd., 2017). Bir başka üretici model olan Yığılı GAN'da eğitim işlemi birden fazla katmana ayrılıp, genel yapı yukarıdan aşağıya doğru yığılan GAN'lardan oluşturulmaktadır (Huang vd., 2017). CIFAR-10 veritabanı ile yapılan Inception skorlamada (Salimans vd., 2016) bu yöntemin en iyi sonucu verdiği belirtilmektedir. BEGAN yapısında, üretic ve ayırtaçın eğitim süresince dengelenmesi önerilmekte ve Wasserstein uzaklığından türetilen yeni bir yakınsama ölçümü sunulmaktadır (Berthelot vd., 2017). Bu çalışmada ayırtaç ağ yapısı olarak daha önce Enerji-Temelli GAN (EBGAN) (Zhao vd., 2016) ile önerilen bir otokodlayıcı kullanılmaktadır. EBGAN modeli ayırtaç bir enerji fonksiyonu olarak tanımlanmaktadır.

Bahsi geçen bu modeller ve diğerleri, Google Brain'deki araştırmacılar tarafından test edilmiştir. Sonuç olarak, eğitim için daha önemli olan noktanın ağ yapısının seçiminden ziyade hiperparametrelerin optimizasyonu olduğu ve farklı modellerin benzer sonuçlar ortaya koydukları belirtilmiştir (Lucic vd., 2017).

Zhang vd. (2017) 2-boyutlu yüz imgelerinden 3-boyutlu modellerin öğrenildiği derin öğrenme temelli bir yaklaşım önermişlerdir. *Daraltan Yığın Otokodlayıcı* (SCAE) adı verilen ve otokodlayıcılara dayalı, hem girdi imgelerin hem de bunların karşılığı olan 3-boyutlu modellerin daha düşük boyuttaki özneliklerini öğrenen ve bu iki ağı bir üçüncüyle bir araya getirdikleri bir yapı geliştirmişler.

3. TEORİK BİLGİ

Bu ana bölümde, bu tez çalışmasında ihtiyaç duyulan ve kullanılan teorik bilgiler sunulmaktadır. Üzerinde durulan konular sırasıyla, görüntüden sayısal imgenin oluşumu, kamera modelleri, projeksiyon ve dönüşümler, öznitelik noktaları, stereo görüş, bir yüz imgesi ve referans bir yüze ait derinlik haritası kullanarak derinlik bilgisinin bulunduğu bir yöntem, bilinen derinlik haritası kullanarak albedo bilgisinin elde edilmesi, aydınlatma şartlarının kestirimi yöntemleri, yapay zekâ, makine öğrenmesi ve derin öğrenmedir.

3.1 Görüntüden Sayısal İmgenin Oluşumu

Bir sayısal imgenin hangi şartlar altında ve nasıl oluştuğu bilgisinin anlaşılması, görüntü işleme ve bilgisayarla görü alanlarında incelenecek ya da geliştirilecek yöntemler için elzem olan şartlardan biridir. Bir nesne yüzeyinden belli bir yansıma kuralına göre yansıyan ışık kameraya ulaşarak sensörler vasıtasıyla sayısal imgeye dönüştürülmektedir. Bu bölümde, yaygın şekilde kabul edilmiş olan Lambert Yansıma Kanunu ve bu kanundan yola çıkan Lambert Yansıma Modeli hakkında (Koppal, 2014), ayrıca küresel harmonik konularında bilgiler verilmektedir.

3.1.1 Lambert Kanunu ve doğrusal Lambert nesnelere

Nesnelerin şekillerinin yüzeyleri üzerinden yansıyan ışığı nasıl etkilediğinin anlaşılabilmesi için yansıma modellerine ihtiyaç duyulur. Çok sayıda yansıma modeli bulunmasına rağmen, SFS yöntemlerinde çoğunlukla Lambert modeli kullanılmaktadır. Çoğu yüz yeniden oluşturma uygulaması, insan yüzünün Lambert kuralına göre modellenebileceği varsayımıyla hareket eder. Gerçekten de, yapılan çalışmalarda insan derisinin yaklaşık olarak Lambert yansıma özellikleri gösterdiği ispatlanmıştır. Lambert kanunu, materyallerin ışığı bütün yönlerde eşit oranda soğurduğunu ve yansıttığını söyler. Bu modelin tek parametresi, cisim üzerinde her noktada bulunan albedo bilgisidir. Albedo, bir noktada ışığın hangi oranda yansıtıldığını tanımlar. Bir nesnenin yüzeyindeki bir nokta için Lambert yansıması,

$$I = \rho L \cdot N \quad (3.1)$$

şeklinde gösterilebilir. Burada, I bu noktadan yansıyan ışığın şiddetini, ρ bu noktanın yüzey albedosunu (yansıtma özelliğini), L ışık kaynağının şiddetini ve yönünü gösteren vektörü

ve N de bu noktadaki yüzey normalini göstermektedir. L ve N 'nin nokta çarpımı $|L| \cdot |N| \cdot \cos\theta$ 'ya eşittir. θ , L ve N arasındaki açıdır. Eğer N bir birim vektörse, Lambert yansıma denklemi $I = \rho \cdot |L| \cdot \cos\theta$ şeklinde yazılabilir. Bu görece olarak basit model, parlak olmayan materyaller için geçerlidir (Basri ve Jacobs, 2003). Işığın iki yönlü yansıtma dağılımı fonksiyonunun (BRDF), gelen ışığın açısının 30° olduğu durumlarda Lambert kuralına göre davrandığı gözlemlenmiştir. Bu durumdan uzaklaşma, daha yüksek açılarda (60° üzerinde) görülmüştür (Marschner vd., 1999; Kemelmacher-Shlizerman ve Basri, 2011). Bir yüzey materyaline ait BRDF, yansıtma oranını yani yüzey tarafından her yöne doğru yansıtılan ışığın yüzey üzerine her yönden gelen ışığa oranını tanımlayan bir fonksiyondur. Yansıtılan ışığın birimi, *birim alan/katı açı* başına düşen güçtür. Gelen ışık ise birim alana düşen güç olarak ölçülür. Bir Lambert yüzeyine ait BRDF, böyle bir yüzey ışığı her yöne eşit oranda yansıtacağı için sabit ve $1/\pi$ 'ye eşittir. Bunun tersine, parlaklık fonksiyonu, bütün ışığın dağılımına karşın bir birim yüzeyin yansıtmasını tanımlar (Basri ve Jacobs, 2003). Bir doğrusal Lambert yüzeyli cisim, şu iki özelliği aynı anda sağlayan görsel bir nesnedir: 1) Baz nesnelere tarafından doğrusal olarak kapsanır, 2) Değişken albedo alanı içeren Lambert yansıtma modelini sağlar. Bir insan yüz imgesi h , baz imgelerin doğrusal kombinasyonu ile ifade edilecek olursa:

$$h = \sum_{i=1}^m f_i h_i \quad (3.2)$$

Burada f_i 'ler karıştırma katsayılarıdır (Zhou vd., 2007).

3.1.2 Küresel harmonikler ve Lambert yansıma modeli

Küresel yüzey harmonikleri bir küre üzerinde tanımlı özel fonksiyonlardır. Özellikle kısmi diferansiyel denklemlerin çözümünde kullanılmaktadırlar. Küresel harmonikler küre üzerindeki dikgen fonksiyonların tam bir kümesidir. Böylece, sinüzoidal fonksiyonların Fourier serileri aracılığı ile bir çember üzerinde tanımlı fonksiyonları göstermelerine benzer şekilde bir küre yüzeyi üzerinde tanımlanacak fonksiyonları temsil etmede kullanılabilirler. Küresel yüzey harmonikleri şu şekilde ifade edilebilir (İdemen, 2015):

$$\{P_n^m(\cos\theta) \cos m\phi\} \quad \text{ve} \quad \{P_n^m(\cos\theta) \sin m\phi\} \quad (3.3)$$

a_{nm} ve b_{nm} katsayıları kullanılarak *küresel fonksiyonlar* da şu şekilde tanımlanabilir:

$$Y_n(\theta, \phi) = \sum_{m=0}^n [a_{nm} \cos m\phi + b_{nm} \sin m\phi] P_n^m(\cos \theta) \quad n = 0, 1, \dots \quad (3.4)$$

Ramamoorthi ve Hanrahan, (2001) ve Basri ve Jacobs, (2003), birbirlerinden bağımsız bir şekilde, Lambert yüzeylerinin alçak-geçiren bir filtre gibi davrandıklarını, yansıyan ışığın küresel harmonikler cinsinden ifadesinde enerjisinin %99.2'sinin ilk dokuz harmonik tarafından oluştuğunu göstermişlerdir. Bu değer, ikinci dereceyi de kapsayan harmoniklerin toplamı için geçerlidir. Sıfır harmoniği ve birinci dereceden harmonikler birlikte enerjinin %87.5'lük kısmını kapsarlar. $I(x, y)$ bir yüz imgesi, $z(x, y)$ yüzeyi tanımlayan fonksiyon ve yüzeye ait bir noktadaki yüzey normali

$$n(x, y) = (n_x, n_y, n_z)^T \quad (3.5)$$

şeklinde gösterilebilir. Bu yüzeye yani yüze ait *albedoyu* (bir yüzeyden yansıyan yüzdesi ya da parlaklık derecesi) ise $\rho(x, y)$ ile gösterelim. Lambert yüzeyinden yansıyan ışığı gösteren yansıma fonksiyonu (*irradiance*) $R(x, y)$ şu şekilde ifade edilebilir:

$$R(x, y) \approx \sum_{n=0}^N \sum_{m=-n}^n l_{nm} \alpha_n Y_{nm}(x, y) \quad (3.6)$$

Burada l_{nm} aydınlatmanın harmonik açılımının katsayıları, α_n sadece n 'ye bağlı ve alçak-geçiren bir filtre gibi davranan maskenin katsayıları ve $Y_{nm}(x, y)$ de yüzeyin küresel harmonik fonksiyonudur. Ortonormal bir taban oluşturan harmoniklerin ilk dokuzu şunlardır: $Y_{00}, Y_{11}^o, Y_{10}, Y_{11}^e, Y_{22}^o, Y_{21}^o, Y_{20}, Y_{21}^e, Y_{22}^e$ İmgenin her bir pikseldeki parlaklık yoğunluğu şu şekilde ifade edilebilir:

$$I(x, y) = \rho(x, y)R(x, y) \quad (3.7)$$

3.2 Kamera Modelleri

Kamera, projeksiyon işlevini yürüterek üç boyutlu dünya (nesne uzayı) ile iki boyutlu imgeler arasındaki haritalamayı gerçekleştiren bir sistemdir. Kamera modelleri, genel olarak

sonlu ve sonsuz kameralar olarak iki sınıfa ayrılabilir. Sonlu kamera modellerinin en basiti ve genel olanı *iğne delikli kamera*dır (Hartley ve Zisserman, 2003).

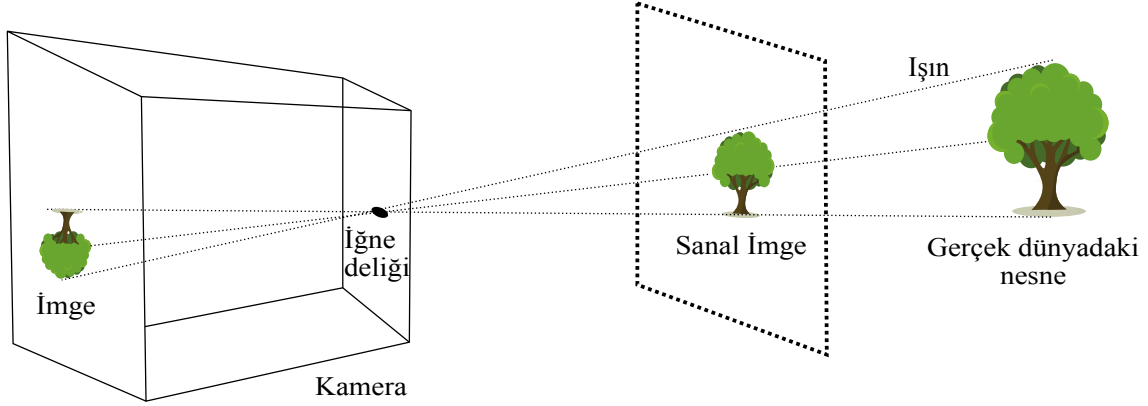
Bir sayısal görüntü oluşumunda şu adımlar gerçekleşir: Işık bir ya da birden fazla kaynaktan yayılır, ardından materyaller ile fiziksel etkileşime girer. Bu etkileşimler şunlardır: Yansıma, kırılma ve dağılma. Işığın bir kısmı kameraya erişir ve burada ölçümü yapılır. Bu model doğrudan model olarak adlandırılır ve nesnelerin geometrileri, ışık kaynakları, materyal özellikleri biliniyor ise kamera tarafından oluşturulacak görüntü bilgisayar grafik teknikleri ile simüle edilebilir. Bir görü algoritmasının nihai hedefi tam bir *rekonstrüksiyon* oluşturmaktır. Burada hedef, az önce bahsedilen doğrudan modelin tersine çevrilerek, ışık kaynaklarının, materyallerin ve geometrilerin kestirilmesidir. Fakat, daha önce de anlatıldığı üzere tam bir rekonstrüksiyon gerçekleştirmek oldukça zordur. Bunun en önemli nedenlerinden biri çözümün tekil olmayışıdır (Prince, 2012).

3.2.1 İğne delikli kamera modeli

İğne delikli ya da projektif kamera modeli, gerçek dünyadaki noktaların imge düzlemine yansıtıldığı geometrik bir modeldir. Bu kamera modeli, dünya noktaları ile imge noktaları arasındaki ilişkiyi matematiksel olarak içerir.

Şekil 3.1’de iğne delikli kamera modeli gösterilmektedir. Bu gösterimde dünyada yer alan bir nesneden yansıyan ışınlar kameranın deliğinden geçerek arka düzlemdeki görüntüyü oluşturur. Fakat bu görüntü yukarıdan aşağıya ters bir şekilde gerçekleşmektedir. Bunun yerine, kameranın önünde bulunan sanal bir düzlemde olduğu varsayılan görüntü ile çalışmak daha kolay olacaktır. Bu iki görüntü yönleri hariç birbiriyle aynıdır. Işınlardan keşiştiği iğne deliği noktası, *optik merkez* olarak adlandırılır. 3-boyutlu dünya koordinat sisteminde noktalar $\mathbf{w} = [u, v, w]^T$ ile gösterilirse, *imge düzleminde* oluşturulan sanal imge optik merkezden w eksenini (ya da *optik eksen*) boyunca uzaktadır. Optik merkezin, 3-boyutlu dünya koordinat sisteminin orijin noktası olduğu varsayılabilir. Optik eksenin imge düzlemine çarptığı nokta *asıl nokta* olarak adlandırılır. Asıl nokta ile optik merkez arasındaki mesafeye (yani imge planı ile iğne deliği arasındaki mesafeye) *odak uzunluğu* adı verilir (Prince, 2012).

Şekil 3.1 İğne delikli kamera modeli.



3.3 Projeksiyon ve Dönüşümler

Sayısal bir imgede yer alan 2-boyutlu piksel koordinatları $\mathbf{x} = (x, y) \in R^2$ şeklinde, ya da

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.8)$$

şeklinde gösterilebilir. Bu noktalar aynı zamanda *homojen koordinatlar*, $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in P^2$ olarak da ifade edilebilirler. Bu ifade ile gösterilen ve sadece ölçeklenme farkı taşıyan tüm vektörler eş sayılırlar. $P^2 = R^3 - (0, 0, 0)$ ile gösterilen uzay, *2-boyutlu projektif ya da izdüğümsel uzay* olarak adlandırılır (Szeliski, 2010).

Bir homojen vektör $\tilde{\mathbf{x}}$, homojen olmayan \mathbf{x} vektörüne, bütün elemanlarını son elemana, yani \tilde{w} 'ya bölerek dönüştürülebilir:

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\mathbf{x}} \quad (3.9)$$

Bu ifadede, $\bar{\mathbf{x}} = (x, y, 1)$ *genişletilmiş (augmented) vektörü* göstermektedir. Son elemanı $\tilde{w} = 0$ olan homojen vektörler, *ideal noktalar* ya da *sonsuzdaki noktalar* olarak adlandırılırlar ve eş birer homojen gösterimleri bulunmaz (Szeliski, 2010).

Üçüncü boyut eklendiğinde, benzer şekilde inhomojen koordinatlar, homojen koordinatlar ve genişletilmiş vektörler sırasıyla, $\mathbf{x} = (x, y, z) \in R^3$, $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w}) \in P^3$, ve $\bar{\mathbf{x}} = (x, y, z, 1)$ ile gösterilebilirler. $\tilde{\mathbf{x}} = \tilde{w}\bar{\mathbf{x}}$ ifadesi gene geçerlidir.

Paralel ilerleyen tren raylarını gösteren bir imgede, raylar sonsuzlukta birleşmektedirler. Bu birleşim noktasına *kaçış noktası* denilmektedir. Projektif geometri, Öklid geometrisinin bir uzantısı olarak bu ve benzeri fenomenleri ifade etmekte kullanılan daha geniş bir geometridir. İki geometri arasında yer alan iki farklı geometri türü de söz konusudur. Bu dört geometri türüne ilişkin karşılaştırma Çizelge 3.1'de gösterilmektedir (Faugeras vd., 2004). Çizelgede gösterilen her geometri türü, bir sonraki türün bir alt kümesidir. Daha genel dönüşümler daha zayıf değişmezlikler anlamına gelmektedir.

Çizelge 3.1 Geometri türlerinin karşılaştırılmaları: dönüşüm türleri ve dönüşümler sonucu değişikliğe uğramayan özellikler.

	Öklid	Benzerlik	Afin	Projektif
Dönüşümler				
döndürme, öteleme	✓	✓	✓	✓
yönbağımsız ölçekleme		✓	✓	✓
eksenler boyunca ölçekleme, makaslama			✓	✓
perspektif projeksiyon				✓
Değişmezler				
mesafeler	✓			
açılar, mesafelerin oranları	✓	✓		
paralellik, kütle merkezi	✓	✓	✓	
çakışım, çapraz oranlar	✓	✓	✓	✓

Projektif geometrinin görüntüleme işlemine uyarlanması esnasında, dünya bir 3-boyutlu projektif uzay ile (\mathbb{R}^3 uzayı ve sonsuzdaki noktaların toplamı: \mathbb{P}^3) ve oluşan imge de 2-boyutlu projektif düzlem ile (\mathbb{P}^2) modellenebilir. \mathbb{P}^3 uzayındaki noktalar, homojen

koordinatlar cinsinden $(X, Y, Z, T)^T$ ifade edilecek olursa, bir $P_{3 \times 4}$ kamera projeksiyon matrisi, \mathbb{P}^3 ile \mathbb{P}^2 arasında doğrusal bir haritalama tanımlar:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = P_{3 \times 4} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \quad (3.10)$$

Tanım *2-boyutta projektiflik (İzdüşüm)*: \mathbb{P}^2 uzayından kendine tersinir bir h haritalamasıdır, öyle ki $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ şeklindeki üç nokta sadece ve sadece $h(\mathbf{x}_1), h(\mathbf{x}_2)$ ve $h(\mathbf{x}_3)$ aynı doğru üzerindeyse aynı doğru üzerindedirler.

3.3.1 Projektif dönüşüm

Bir düzlemsel projektif dönüşüm, tekil olmayan bir 3×3 matris tarafından homojen 3-vektörlerin doğrusal dönüşümüdür:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad (3.11)$$

ya da daha kısa olarak $\mathbf{x}' = H\mathbf{x}$ olarak gösterilir (Hartley ve Zisserman, 2003). Blok formunda ise;

$$\mathbf{x}' = H_P \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \mathbf{x} \quad (3.12)$$

şeklinde gösterilebilir. Burada \mathbf{v} vektörü $(v_1, v_2)^T$ 'dir. Buradaki matrisin dokuz elemanı olmakla birlikte bunların sadece oranları önemlidir. Dolayısıyla, bu dönüşüm sekiz parametre ile belirtilebilir.

3.3.2 Afın Dönüşüm

Bir afın dönüşüm (ya da kısaca *afinite*), tekil olmayan bir dönüşüm ve bunu takip eden bir ötelemedir. Matris gösterimi olarak,

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.13)$$

şeklinde ya da blok formunda,

$$\mathbf{x}' = H_A \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \quad (3.14)$$

şeklinde gösterilebilir. Düzlemsel bir afın dönüşüm, matriste yer alan altı değişkene karşılık gelen altı serbestlik derecesine sahiptir.

3.3.3 Benzerlik dönüşümü

Bir benzerlik dönüşümü (ya da kısaca *benzerlik*), isotropik bir ölçekleme ile oluşturulan bir isometridir. Bir Öklid dönüşümünün bir ölçekleme ile oluşturulduğu durumda, benzerlik şu matris gösterimine sahiptir:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (3.15)$$

Blok formunda ise,

$$\mathbf{x}' = H_S \mathbf{x} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \quad (3.16)$$

şeklinde gösterilebilir. Buradaki s skalar ifadesi, ölçeklemeyi temsil etmektedir. Düzlemsel bir benzerlik dönüşümü dört serbestlik derecesine sahiptir.

3.3.4 İsoetri ya da Öklid dönüşümü

İsoetirler, Öklid uzaklığını koruyan \mathbb{P}^2 düzlemindeki dönüşümlerdir. Bir isometri şu şekilde gösterilebilir:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon \cos \theta & -\sin \theta & t_x \\ \epsilon \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (3.17)$$

Bu ifadede $\epsilon = \pm 1$ 'dir. Eğer $\epsilon = 1$ ise, isometri yön koruyucudur, böylece öteleme ve döndürmeden oluşan bir Öklid Dönüşümü'dür. Düzlemsel bir Öklid Dönüşümü, blok formunda şöyle ifade edilir:

$$\mathbf{x}' = H_E \mathbf{x} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \quad (3.18)$$

3.4 Öznitelik Noktaları

Bir imge özniteliğinin genel bir tanımı olmamakla birlikte, ayırt edici bir özellik ya da etiketlenmek istenen bir olgunun tanımı olarak algılanabilir (Gonzalez ve Woods, 2018). *Öznitelik çıkarma* işleminin iki temel adımı vardır: *Öznitelik sezme* ve *öznitelik betimleme*.

3.4.1 Öznitelik seziciler

Öznitelik sezme işlemi, bir imge, bölge ya da sınır içindeki öznitelikleri bulma eylemidir. Tüm bir imge üzerinde köşeleri sezme için geliştirilen *Harris-Stephens (HS) köşe sezici* algoritması (Harris ve Stephens, 1988), *Azami Ölçüde Kararlı Uç Bölgeler (MSER)* yöntemi (Matas vd., 2004), *Ölçek Bağımsız Öznitelik Dönüşümü (SIFT)* (Lowe, 2004) en çok kullanılan yöntemlerden bazılarıdır.

3.4.2 Öznitelik çıkarma (Özütleme) ve öznitelik betimleyiciler

Öznitelik betimleme, sezilen özniteliklere nicel niteliyiciler atama işlemidir. Bir öznitelik betimleyicinin değeri betimlenen olguya uygulanan birtakım dönüşümlere bağlı olarak eğer değişmiyorsa bu betimleyiciye *değişmezdir* denir.

3.4.3 Eşleştirme

Bir ya da daha fazla imge üzerinde öznitelik noktaları bulunup betimlendikten sonra, bir sonraki adımda bu imgeler arasındaki noktasal eşleştirmenin yapılması yer alır. Bu eşleştirme işleminde yürütülecek algoritmanın başarımı, aşağıdaki tanımlarda verilen ölçütler kullanılarak hesaplanabilir:

- Gerçek Pozitif (TP): Doğru eşleştirmelerin sayısı,
- Yanlış Negatif (FN): Doğru bir şekilde sezilmeyen eşleştirmelerin sayısı
- Yanlış Pozitif (FP): Önerilen ama yanlış olan eşleştirmelerin sayısı,
- Gerçek Negatif (TN): Eşleştirme olmayan ve doğru şekilde reddedilenlerin sayısı.

Bazı eşleştirmeler elde edildikten ve doğru oldukları varsayıldıktan sonra, geometrik hizalama ile hangi eşleştirmelerin aykırı (*outlier*) ya da normal (*inlier*) olduklarının doğrulanması işlemi gerçekleştirilebilir. Küçük bir tohum kümesinin seçildiği ve ardından daha geniş bir kümenin doğrulandığı *rasgele örnekleme* ya da RANSAC yöntemi bu aşamada uygulanabilir (Szeliski, 2010).

3.5 Stereo Görüş

İki adet imgeden 3.boyut bilgisinin hesaplanmasına *stereo görüş* denilmektedir (Kanatani vd., 2016). Stereo eşleştirmede ya da *stereo karşılıklılıkta*, aynı sahneye ilişkin iki ya da daha fazla imgenin alınarak, birbirine denk gelen piksellerin bulunması ve bunun ardından elde edilen piksellerin 2-boyutlu konumlarının 3-boyutlu derinliklere dönüştürülmesi söz konusudur. Girdi imgelerindeki piksellere göreceli olarak derinlik değerlerinin atandığı 3-boyutlu modellere *derinlik haritaları* adı verilmektedir Szeliski, 2010.

Aynı sahneye ilişkin elde edilen iki imgede yer alan ve karşılıklılığı bulunan noktalar arasında geometrik bir ilişki bulunmaktadır. Bu ilişki, sadece iki kameranın içsel (*intrinsic*) parametrelerine ve bu kameraların göreceli öteleme ve döndürmelerine bağlıdır (Prince, 2012).

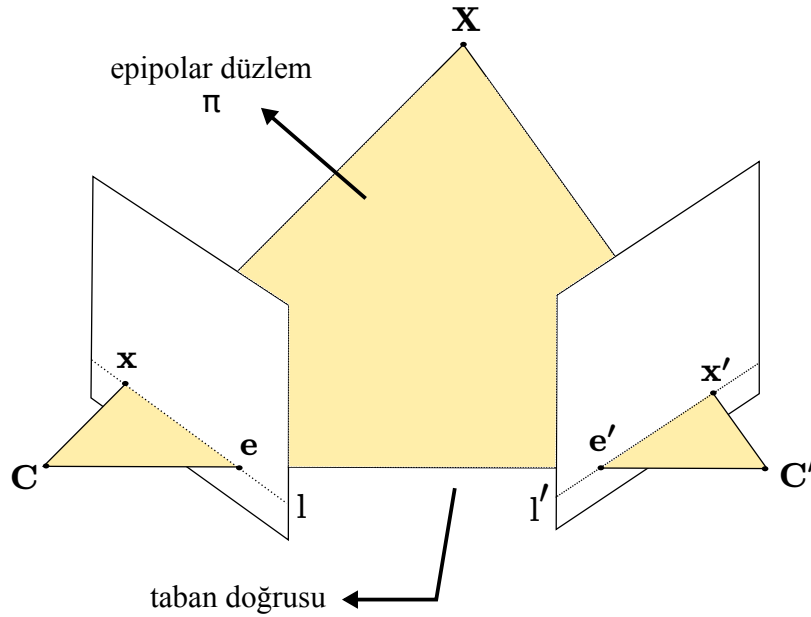
3.5.1 Epipolar geometri

Stereo imgelemede, rol alan iki kameranın merkezleri ile uzayda görüntülenmek istenen bir noktanın oluşturduğu düzleme *epipolar düzlem*; bu düzlemin bir kameranın

imge düzlemi ile kesiştiği çizgiye ise *epipolar çizgi* denir (*Türkçe Bilim Terimleri Sözlüğü* 2019). Kamera merkezlerini birleştiren doğruya *taban doğrusu*, bu doğrunun imge düzlemi ile kesiştiği noktaya ise eşkutup (*epipole*) denilmektedir. Aynı şekilde, bir imgedeki eşkutup, diğer görüntünün kamera merkezinin görüntüsüdür. Epipolar düzlem, taban doğrusunu da içerir.

Şekil 3.2’de, 3-boyutlu uzaydaki bir X noktası, bu noktanın imgeler üzerinde oluşturduğu x ve x' noktaları ve kamera merkezleri C ve C' ortak bir π epipolar düzlemde yer alırlar. Sol imge üzerinde yer alan x noktası 3-boyutlu uzayda C ve x noktaları ile tanımlı bir ışına geri izdüşüm yapar. Bu ışın, sağdaki imgede l' doğrusu üzerinde görüntülenir. x noktasına izdüşüm yapan 3-boyutlu uzaydaki X noktası bu ışın üzerinde yer almak zorundadır. Bu yüzden, X noktasının sağ imgedeki görüntüsü l' doğrusu üzerindedir (Hartley ve Zisserman, 2003).

Şekil 3.2 Epipolar geometri.



3.5.2 Temel matris

Temel ya da fundamental matris F , aynı sahneye ilişkin iki imge arasındaki epipolar geometrik özellikleri yani içsel parametreleri içerir. Temel matris, sahne yapısından bağımsızdır. Bununla beraber, görüntülenen sahnede yer alan ve karşılıkları bulunan noktalar biliniyor ise sadece bunlar kullanılarak hesaplanabilir. Böylece kameraların içsel parametrelerine ya da göreceli poza ihtiyaç duyulmaz (Hartley ve Zisserman, 2003).

3-boyutlu uzayda X noktası ilk imgede x noktası ile ve aynı nokta ikinci imgede x' noktası olarak görüntülenmişse bu noktalar şu ilişkiyi sağlarlar:

$$\mathbf{x}'^T F \mathbf{x} = 0. \quad (3.19)$$

3.5.3 Hareketten yapı çıkarımı (*Structure from motion*)

Durağan bir nesne etrafında dönen bir kamera ile alınmış imgelerden yola çıkarak, bu nesnenin 3-boyutlu modelini elde etme amacıyla, aynı anda hem kameranın parametrelerini hem de her bir imge karesine göre kamera konumunu bulma problemine *Hareketten Yapı Çıkarımı* adı verilmektedir (Prince, 2012).

3.6 Bir Yüz İmgesi ve Referans Yüze Ait Derinlik Haritası Kullanarak Derinlik Bilgisinin Bulunması

Kemelmacher-Shlizerman ve Basri, bir iki boyutlu girdi yüz imgesi ve bir başka referans yüze ait derinlik haritasını kullanarak derinlik bilgisinin kestirimine yönelik bir yöntem geliştirmişlerdir (Kemelmacher-Shlizerman ve Basri, 2011). Hobbel, Kemelmacher-Shlizerman ve Basri'nin yöntemini, C++ programlama dilinde uygulayarak yöntemin eksikliklerini ve zayıf kalan yönlerini bulmaya çalışmıştır (Hobbel, 2013). Buna göre, insan yüzünde burun ve kaşlar gibi hızlı derinlik değişiminin yaşandığı yerlerde bu yöntem, referans modeli başarılı bir şekilde değiştirememektedir. Ayrıca, referans modeldekinden farklı pozlardaki imgelerle de problemlerli sonuçlar üretmekte, dişlerin ve gözlerin bazı kısımları çıkıntılar oluşturmaktadır. Bu uygulamada, insan yüzünü içeren tek bir girdi imgesi kabaca aynı hizaya gelen bir genel referans yüz modeli ile kullanılmaktadır. 2-boyuttan 3-boyuta geçmek ve üçüncü boyutu kestirebilmek için ek varsayımlara ihtiyaç duyulur. Bu durumda, insan yüzlerinin genel seviyede birbirlerine çok benzer oldukları (kabaca aynı büyüklükte ve birbirlerinden aynı mesafede bulunan bir burun, bir ağız ve iki adet göz olduğu) bu varsayımlardan biridir. Böylece, referans model bir genel şablon gibi kullanılabilir. Bunu yaparken, girdi yüze ait imgedeki karakteristik detaylar ton farklılıkları kullanılarak elde edilip, şablon *tonlamadan şekil yöntemi (SFS)* ile yeniden şekillendirilebilir (Hobbel, 2013).

Kemelmacher-Shlizerman ve Basri'nin yöntemi üç aşamada gerçekleştirilebilir. Girdi imgesi için aranılan yüz, referans yüzle hem yüzey normaleri hem de yüzey albedosu cinsinden oldukça benzer olduğundan, ilk adımda imgenin olduğu andaki aydınlatma bilgisini küresel harmonikler cinsinden referans yüze ait veriler kullanılarak yaklaşık olarak

bulunabilir. İkinci adımda, bulunan katsayılar referans modelin albedo bilgisiyle birlikte kullanılarak her bir piksel için derinlik bilgisi SFS yöntemi ile elde edilebilir. Bu yapılırken, oluşturulan derinlik ve referans modelin derinlik bilgisi arasındaki farkın düzleştirilmesi gerekmektedir. Üçüncü ve son adımda ise, elde edilen aydınlatma katsayıları ve oluşturulan derinlik bilgisi kullanılarak aranılan yüze ait albedo oluşturulabilir. İkinci aşamada olduğu gibi, referans modelin albedosu ve elde edilen albedo arasındaki farkın düzleştirilmesi yapılmalıdır (Arslan ve Seke, 2015).

Bir 3-boyutlu yüzeyin oluşturulmasında şunlardan birisi hedeflenebilir:

- Yüze ait her bir piksel için derinlik: $Z(x, y)$
- Nesne yüzeyinin tanjant düzlemine dik olan yönelme vektörü yani yüzey normali: n_x, n_y, n_z
- x ve y yönlerinde derinlik değişimini gösteren yüzey gradyeni: $(p, q) = \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}$
- Yüzey eğimleri verev açısı Φ (*slant*) ve eğiklik açısı θ (*tilt*)

3.6.1 Aydınlatma (ışık kaynağı) vektörünün bulunması

Kemelmacher-Shlizerman ve Basri'nin yönteminin temel yaklaşımı, henüz bilinmeyen değerlerin referans yüzden alınarak *tonlamadan-şekil (SFS) algoritmasında* kullanılmasıdır. Yöntem ilerledikçe, bulunan değerler referans verinin yerine konulabilir. İlk adım için, referans yüze ait yüzeyin girdi imgesiyle yaklaşık olarak aynı yönde olduğu ve referans yüzün albedo bilgisinin aranan yüze yeterli derecede yakın olduğu varsayılır. Böylece, girdi imgenin olduğu aydınlatma şartları küresel harmonikler cinsinden tahmin edilebilir. Referans yüzün yüzey normaleri n_{ref} ve albedo bilgisi ρ_{ref} olarak gösterilsin. Aranılan değerler ise, n ve ρ ile gösterilsin. Piksel yoğunlukları için imge yansıtma fonksiyonu:

$$I(x, y) = \rho(x, y)R(x, y) \quad (3.20)$$

Yansıma haritası $R(x, y)$ küresel harmonikler cinsinden,

$$R(x, y) = l^T Y(n(x, y)) \quad (3.21)$$

şeklinde gösterilir. Aynı ifade şu şekilde de yazılabilir:

$$I(x, y) = \rho(x, y) \sum_{n=0}^N \sum_{m=-n}^n \alpha_n l_{nm} Y_{nm}(n(x, y)) \quad (3.22)$$

3.22 no.lu ifadede, N küresel harmonikler cinsinden açılımın derecesini, l_{nm} ise şu anda bilinmeyen harmonik ışık katsayılarını göstermektedir. α değerleri denklemde alçak-geçiren karakteristiği sağlayan katsayılardır. İlk üç harmonik derecesi için $\alpha_0 = \pi$, $\alpha_1 = 2\pi/\sqrt{3}$ ve $\alpha_3 = 2\pi/\sqrt{8}$ olarak verilmektedirler. $N = 1$ alındığında, $Y(n(x, y))$ ifadeleri toplam 4 adet olmak üzere,

$$Y(n(x, y)) = (c_0, c_1 n_x, c_1 n_y, c_1 n_z)^T \quad (3.23)$$

ve $N = 2$ için ikinci dereceden açılıma kadar ve toplam 9 adet olmak üzere,

$$Y(n(x, y)) = (c_0, c_1 n_x, c_1 n_y, c_1 n_z, c_2 n_x n_y, c_2 n_x n_z, c_2 n_y n_z, c_2 (n_x^2 - n_y^2), c_2 (3n_z^2 - 1))^T \quad (3.24)$$

olarak ifade edilir. c katsayıları da şu şekilde verilmektedir: $(c_0, c_1, c_2) = (1/\sqrt{4\pi}, \sqrt{3/4\pi}, 3\sqrt{5/12\pi})$.

Her bir piksel yoğunluğu için yüzeyde (Ω) bulunan bütün pikseller sayısı kadar denklem olacağından, buradaki tek bilinmeyen ifade l 'nin 4 ya da 9 adet katsayısı olacaktır. Bu değerler,

$$\sum_{x, y \in \Omega} I(x, y) - \rho_{ref} \cdot l^T \cdot Y(n_{ref}(x, y)) \quad (3.25)$$

denklemini minimize edilerek bulunabilir. Her bir piksel için bu ifade, vektörel olarak gösterilecek olursa,

$$I(x, y) = \begin{pmatrix} \rho(x, y)\alpha_0c_0, \rho(x, y)\alpha_1c_1n_x, \rho(x, y)\alpha_1c_1n_y, \rho(x, y)\alpha_1c_1n_z \end{pmatrix} \cdot \begin{pmatrix} l_{00}, l_{1-1}, l_{10}, l_{11} \end{pmatrix}^T. \quad (3.26)$$

Bu problem $\mathbf{Ax} = \mathbf{b}$ doğrusal denklem sistemi şeklindedir. \mathbf{b} vektörünün boyutu piksel sayısı kadardır ve girdi imgesinin kolonlaştırılmış halidir. \mathbf{x} vektörü bilinmeyen kısım olup, 4 elemanlı aydınlatma vektörüdür. \mathbf{A} matrisi her bir piksel için şu satırlardan oluşur:

$$\begin{pmatrix} \rho(x, y)\alpha_0c_0, \rho(x, y)\alpha_1c_1n_x, \rho(x, y)\alpha_1c_1n_y, \rho(x, y)\alpha_1c_1n_z \end{pmatrix}. \quad (3.27)$$

Böylece, \mathbf{A} matrisinin boyutu, *piksel sayısı* \times *küresel harmoniklerin sayısı* olmaktadır.

Bütün bu veriler oluşturulduktan sonra $\mathbf{Ax} = \mathbf{b}$ doğrusal sistemi farklı yöntemlerden biriyle çözülebilir: OpenCV kütüphanesinin (Bradski, 2000) SVD ayrıştırma fonksiyonu veya MATLAB uygulamasında $\mathbf{A} \setminus \mathbf{b}$ operatörü ile.

3.6.2 Derinlik bilgisinin bulunması

İkinci adımda, ilk adımda bulunan küresel harmonik aydınlatma katsayıları kullanılarak, SFS yöntemi ile derinlik bilgisine ulaşılabilir. Bu aşamada, şu fonksiyon minimize edilmek istenmektedir:

$$\int_{\Omega} (I - \rho_{ref} l^T Y(n))^2 + \lambda_1 (\Delta G * d_z)^2 dx dy \quad (3.28)$$

Bu denklemdeki ilk terim, piksel yoğunluğu şeklinde gösterilen parlaklık kısıtıdır. Girdi imgesindeki piksel ile elde edilen yüzey normali sonucunda bulunan parlaklık farkı minimize edilmeye çalışılmaktadır. İkinci terim ise regülasyon terimi olup, elde edilen derinlik bilgisi ile referans modelin derinlik bilgisinin (d_z) farkı minimize edilmektedir. Referans model ile çözüm birebir aynı olmadığı ve insan yüzü üzerindeki çeşitli yerler noktasal olarak aynı yerde olmak yerine bölgesel olarak yer alacağından, bu ifade referans yüz ile çözümlerle elde edilen değer arasındaki fark doğrudan alınmadan, bir bölgenin etrafında Gaussyan ortalama (G) kullanılmaktadır.

Data terimi ile regülasyon terimi arasındaki ilişkiyi kurmak için λ_1 değişkeni kullanılmaktadır. Yüksek bir değer, referans modele daha yakın bir yüzü, daha düşük bir değer ise girdi imgesine daha yakın bir yüzü ifade eder.

Yüzey normali şu şekilde gösterilebilir:

$$n(x, y) = \frac{1}{\sqrt{p^2 + q^2 + 1}}(p, q, -1)^T \quad (3.29)$$

3.29 no.lu ifadede yer alan $p(x, y)$ ve $q(x, y)$ terimleri sırasıyla $\partial z/\partial x$ ve $\partial z/\partial y$ 'ye eşittirler. ∂z terimi derinlik farkını göstermektedir ve aranılan değer derinlik bilgisidir. $Y(n)$ ifadesi de, birinci dereceden ise

$$Y(n) = \left(1, \frac{1}{N}p, \frac{1}{N}q, \frac{-1}{N}\right) \quad (3.30)$$

ikinci dereceden ise,

$$Y(n) = \left(1, \frac{1}{N}p, \frac{1}{N}q, \frac{-1}{N}, \frac{1^2}{N}pq, \frac{-1^2}{N}p, \frac{-1^2}{N}q, \frac{1^2}{N}(p^2 - q^2), \frac{3^2}{N} - 1\right) \quad (3.31)$$

şeklinde verilmektedir. Burada $N(x, y)$ ifadesi şu şekilde gösterilebilir:

$$N(x, y) = \sqrt{p^2 + q^2 + 1} \quad (3.32)$$

$N(x, y)$ ifadesinin yaklaşık olarak $N_{ref}(x, y)$ 'ye eşit olduğu varsayılabilir ve piksel yoğunluğu denkleminde N_{ref} kullanılabilir. Bu denklem için bütün küresel harmoniklerin ifadeleri yazılırsa, denklem birinci derece için

$$I = \rho_{ref}l_0 + \frac{\rho_{ref}}{N_{ref}}(l_1p + l_2q - l_3) \quad (3.33)$$

şekline dönüşür. İkinci derece için ise şu ifade geçerlidir:

$$I = \rho_{ref} l_0 + \frac{\rho_{ref}}{N_{ref}} (l_1 p + l_2 q - l_3) + \frac{\rho_{ref}}{N_{ref}^2} (l_4 p q - l_5 q - l_6 q + l_7 p^2 - l_7 q^2 + 3l_8) - \rho_{ref} l_8 \quad (3.34)$$

İki denklemde de, bilinmeyenler sadece p ve q 'dür. $p(x, y) = \partial z / \partial x$ ve $q(x, y) = \partial z / \partial y$ olduğu için bunlar ileri fark denklemleri ile değiştirilebilirler:

$$p = z(x + 1, y) - z(x, y) \quad (3.35a)$$

$$q = z(x, y + 1) - z(x, y) \quad (3.35b)$$

Böylece birinci dereceden küresel harmonikler için parlaklık denklemi,

$$I = \rho_{ref} l_0 + \frac{\rho_{ref}}{N_{ref}} (l_1 z(x + 1, y) - l_1 z(x, y) + l_2 z(x, y + 1) - l_2 z(x, y) - l_3) \quad (3.36)$$

şeklini alır. Bu denklem $z(x, y)$ için doğrusal olduğu ve her bir piksel için bir denklem olacağı için, toplamda bilinmeyen kadar denklem oluşacaktır. Böylece bu sistem, doğrusal en az kareler optimizasyonu ile çözülebilir. İkinci dereceden harmonikler kullanılacak olursa, oluşan denklem sistemi doğrusal olmayacaktır ve farklı bir optimizasyon yaklaşımına ihtiyaç doğacaktır.

(3.36) denklemi, bir önceki bölümde olduğu gibi bilinenler bir tarafa toplanıp vektörel açıdan gösterilecek olursa,

$$I(x, y) - \rho \left(l_0 - \frac{l_3}{N_{ref}} \right) = \frac{\rho_{ref}}{N_{ref}} \left(-(l_1 + l_2), l_1, l_2 \right) \left(z(x, y), z(x + 1, y), z(x, y + 1) \right)^T \quad (3.37)$$

şeklini alır. Bu problem gene $\mathbf{Ax} = \mathbf{b}$ doğrusal denklem sistemi şeklini almaktadır. Burada \mathbf{b} vektörü tekrar herhangi bir bilinmeyene bağlı olmayan, piksel sayısı kadar uzunlukta bir vektördür. \mathbf{x} aranılan derinlik bilgisidir ve her piksel için bir $z(x, y)$ değeri olacağı için bu vektörün boyutu da piksel sayısı kadardır. \mathbf{A} matrisi denklemin sağ tarafında verilen ve

bilinmeyenlerle çarpılan bilinen değerleri içerir. Bu matrisin boyutu, \mathbf{x} ve \mathbf{b} vektörlerine uyması için *piksel sayısı* \times *piksel sayısı* olmalıdır. Sağ taraftaki ilk parantez \mathbf{A} matrisini, ikinci parantez ise bilinmeyen \mathbf{x} vektörünü göstermektedir. \mathbf{x} vektörü sadece burada gösterilen 3 adet z ifadesini değil, bütün pikseller için geçerli olan z 'leri içerir. Bu da, \mathbf{A} matrisinin her bir piksel için oluşan satırlarında, sadece 3 tane sıfır olmayan değer olacağını gösterir. Bunlar, $z(x, y)$, $z(x + 1, y)$ ve $z(x, y + 1)$ 'e karşılık gelen yerlerdedir. Geri kalan noktalarda 0 değerleri olacaktır. Böylece, \mathbf{A} matrisi oldukça büyük ve seyrek (*sparse*) tanımlanmış bir matris olacaktır. Bunun çözümü için, SuiteSparse yazılım paketinden gelen UMFPACK kütüphanesi (Davis, 2004) ya da MATLAB operatörü kullanılabilir. Her bir denklem için çok sayıda çözüm olacağı için bu aşamada iyi bir sonuç almak olası olmayacaktır. O yüzden bir regülasyon terimine ihtiyaç vardır.

Regülasyon terimi, $d_z(x, y)$ ile d_z 'nin (x, y) çevresinde bir Gausyen fonksiyon uygulanması sonucunda elde edilen ortalaması arasındaki fark olarak alınmaktadır. Bu, aşağıdaki denklem sisteminin iki tarafı arasındaki farkı minimize etmektedir:

$$\lambda_1(z(x, y) - G * z(x, y)) = \lambda_1(z_{ref}(x, y) - G * z_{ref}(x, y)) \quad (3.38)$$

3.38 no.lu denklemlerle gösterilen sistem de $z(x, y)$ için doğrusaldır. Regülasyon ifadesinin sağ tarafında sadece bilinen değerler yani referans modelin derinlik bilgisi vardır. Böylece bu kısım her bir piksel için hesaplanıp \mathbf{b} vektöründe karşılık gelen yere eklenebilir. Denklemin sol tarafı için, \mathbf{A} matrisinin satırlarındaki uygun yerlere Gausyen katsayıları, karşılık gelen piksele uzaklıklarına bağlı olarak ve λ_1 katsayısıyla çarpılarak eklenmelidir. Her bir derinlik için data terimi ve regülasyon terimi için birer adet olmak üzere iki adet denklem olsa da denklem sistemi hala eksik tanımlıdır. Bu yüzden sınır kısıtlarını eklemek zorunludur. Referans modelin sınırları kullanılırken, Dirichlet sınır koşulları ya da referans modelin sınırlarında kullanılacak türevleri için Neumann sınır koşulları kullanılmalıdır. Fakat, bu kısıtlar aranan sonuçla uyumlu olmayabileceği için çok sınırlayıcı olabilirler.

Böylece oluşan bütün seyrek matris, LU faktörizasyonu ile ayrıştırılıp doğrusal en az kareler optimizasyonu şeklinde çözümlenebilir. \mathbf{x} ifadesi ile aranan değerler $m \times n$ boyutunda ve toplam k piksele sahip bir imge için şu şekilde gösterilebilir:

$$\mathbf{x} = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(k-1) \\ z(k) \end{bmatrix} \quad (3.39)$$

Burada z değerleri, her bir piksel noktasına karşılık gelen yüzey derinliğidir.

3.6.3 Albedo bilgisinin bulunması

Üçüncü ve son adım olarak, ikinci adımda yüze ait üçüncü boyut bilgisi elde edildikten sonra albedo oluşturulabilir. Bunun için, şu denklem minimize edilecektir:

$$\int_{\Omega} (I - \rho^T Y(n))^2 + \lambda_2 (\Delta G * d_{\rho})^2 dx dy \quad (3.40)$$

Burada, ρ ve doğrudan ilişkili olan d_{ρ} bilinmeyen değişkenlerdir. İlk ve ikinci terimler sırasıyla data ve regülasyon terimleridir. Regülasyon terimi, derinlik bilgisi adımına benzer şekilde elde edilen albedo ile referans model albedosu arasındaki farkı düzleştirmektedir. Albedo için sınır şartlarına ihtiyaç duyulmamaktadır. Sınırlara gelindiğinde işlemi durdurmak yeterli olacaktır.

Aydınlatma vektörü ve derinlik bilgisi önceki iki adımda elde edildiği için bu adımda tek bilinmeyen albedo bilgisidir. Böylece, referans imgeden herhangi bir verinin kullanılması gerekmemektedir. Önceki adımlarda olduğu üzere, bu adımda da problem bir $\mathbf{Ax} = \mathbf{b}$ doğrusal denklem sistemi şeklinde kurulabilir. Piksel sayısı kadar denklem oluşacağı için, \mathbf{b} vektörü toplam piksel adedi kadar bir boyuta sahip, bilinenlerden oluşan

bir vektördür. \mathbf{x} , denklemin bilinmeyen kısmıdır. Burada bilinmeyen albedodur (ρ). Her bir piksel için bir albedo olacağından, \mathbf{x} vektörünün boyutu da piksel adedi kadardır:

$$\mathbf{x} = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(k-1) \\ \rho(k) \end{bmatrix} \quad (3.41)$$

\mathbf{A} matrisi, bilinmeyenlerle çarpılacak olan denklemin bilinenlerinden oluşur, \mathbf{b} ve \mathbf{x} vektörleriyle uygunluk göstermesi için *piksel sayısı* \times *piksel sayısı* boyutuna sahip olmalıdır. \mathbf{b} vektörü kısaca girdi test imgesinin vektörel hale getirilmesinden oluşur. \mathbf{A} matrisinde data teriminin bilinenler kısmı,

$$l_{00}c_0a_0 + l_{1-1}c_1a_1p + l_{10}c_1a_1q - l_{11}a_1c_1 \quad (3.42)$$

olarak gösterilebilir. Regülasyon terimi olarak (3.38) denklemine benzer şekilde,

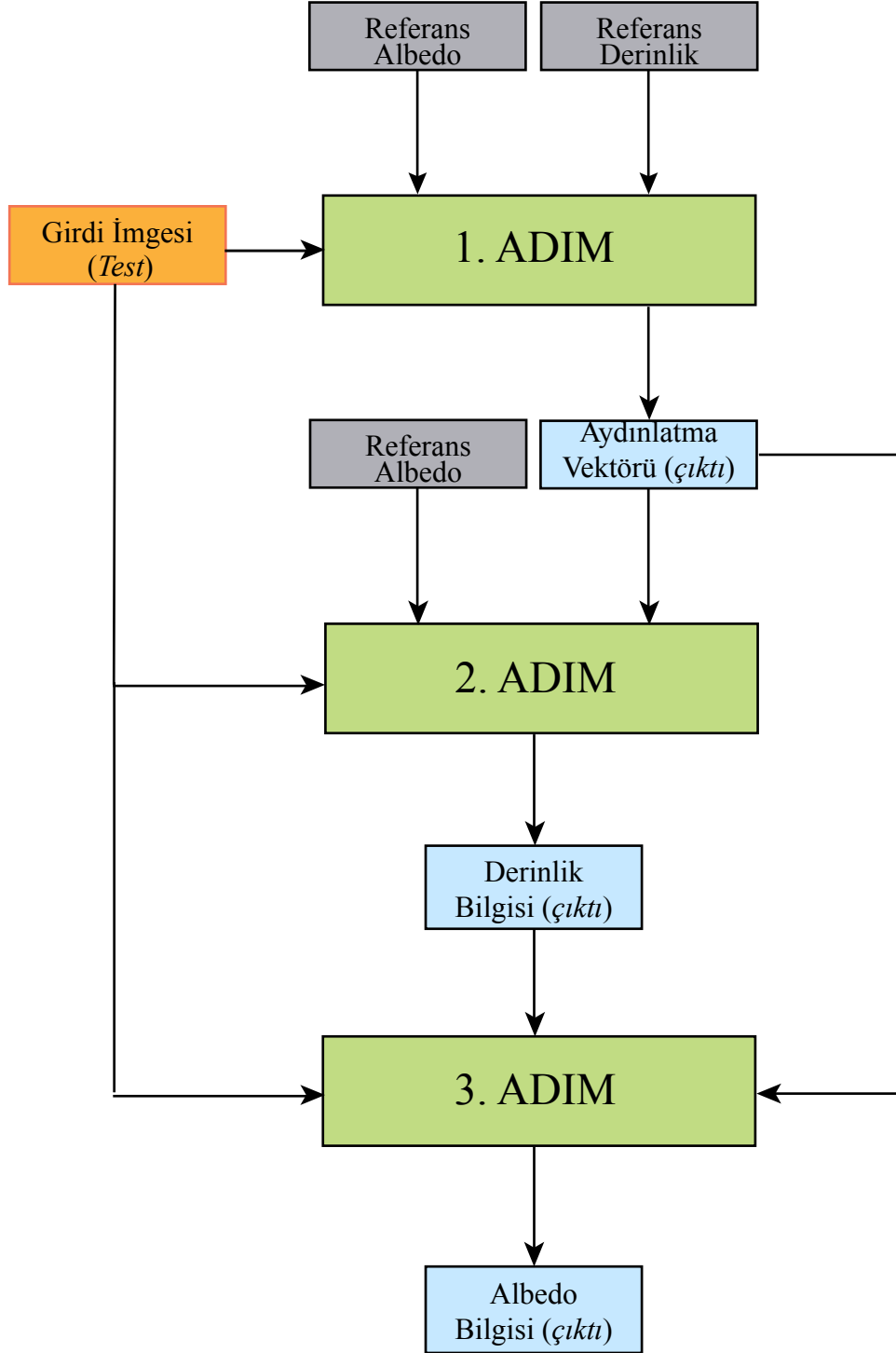
$$\lambda_2(\rho(x, y) - G * \rho(x, y)) = \lambda_2(\rho_{ref}(x, y) - G * \rho_{ref}(x, y)) \quad (3.43)$$

eşitliği söz konusudur. Bu terimler tekrar \mathbf{A} matrisine eklenmelidir. Bu aşamada, doğru bir yakınsama için sınır koşullarına ihtiyaç duyulmadığından sınırlarda algoritma durdurulur. \mathbf{A} matrisi ikinci adımdakine benzer şekilde seyrektiler. UMFPACK kütüphanesi kullanılarak LU faktörizasyonu ile veya MATLAB uygulamasında $\mathbf{A} \setminus \mathbf{b}$ operatörü ile çözüm aranabilir.

Uygulamanın üç adımını ve her adımda yer alan girdileri ve çıktıları gösteren genel bir akış diyagramı Şekil 3.6.3'te görülebilir.

3.6.4 İkinci dereceden küresel harmonikler cinsinden açılım ve optimizasyon çözümü

Bu bölümde, daha önceki üç bölümde gerçekleştirilen birinci dereceden küresel harmonikler cinsinden açılım ve doğrusal yöntemlerle çözümün aksine, ikinci dereceden küresel harmonikler cinsinden açılım ve bunun çözümü tartışılacaktır. İlk adım olarak



Şekil 3.3 Tek bir iki-boyutlu imgeden derinlik bilgisinin kestirimi için Kemelmacher-Shlizerman ve Basri'nin ortaya koyduğu yöntemin (Bölüm 3.6) uygulanmasının genel akışı. Bu algorithmada, test girdi imgesinin yanı sıra referans bir bireye ait derinlik ve albedo haritaları da kullanılmaktadır.

aydınlatma katsayıları çözümü için tekrar 3.25 no.lu denklemden hareketle, 3.26 no.lu denklem bu sefer 9 harmoniği kapsayacak şekilde gösterilirse,

$$I(x, y) = \rho(x, y) \left(\alpha_0 c_0, \alpha_1 c_1 n_x, \alpha_1 c_1 n_y, \alpha_1 c_1 n_z, \alpha_2 c_2 n_x n_y, \alpha_2 c_2 n_x n_z, \alpha_2 c_2 n_y n_z, \right. \\ \left. \alpha_2 c_2 \left(\frac{n_x^2 - n_y^2}{2} \right), \alpha_2 c_2 \left(\frac{3n_z^2 - 1}{2\sqrt{3}} \right) \right) \cdot \left(l_{00}, l_{1-1}, l_{10}, l_{11}, l_{2-2}, l_{2-1}, l_{20}, l_{21}, l_{22} \right)^T \quad (3.44)$$

ifadesi elde edilir. Bu adımda tekrar, yüzey normaleri ve albedo değerleri referans imgeden geldiği için bilinmeyenler sadece aydınlatma katsayılarıdır. \mathbf{x} , 9 adet bilinmeyen olmak üzere $\mathbf{Ax} = \mathbf{b}$ doğrusal sistemi kurulup çözüm elde edilebilir. \mathbf{x} ifadesi şu şekilde gösterilebilir:

$$\mathbf{x} = \begin{bmatrix} l_{0,0} \\ l_{1,-1} \\ l_{1,0} \\ l_{1,1} \\ l_{2,-2} \\ l_{2,-1} \\ l_{2,0} \\ l_{2,1} \\ l_{2,2} \end{bmatrix} \quad (3.45)$$

Yüzey derinlik değerlerinin bulunmaya çalışıldığı ikinci adıma geçildiğinde, küresel harmonik ifadeler yüzey türevleri cinsinden ve katsayılar da yerlerine koyularak şu şekilde ifade edilebilir:

$$\mathbf{Y}(\mathbf{n}) = \left(\alpha_0 c_0, \frac{\alpha_1 c_1}{N} p, \frac{\alpha_1 c_1}{N} q, \frac{-\alpha_1 c_1}{N}, \frac{\alpha_2 c_2}{N^2} pq, \frac{-\alpha_2 c_2}{N^2} p, \right. \\ \left. \frac{-\alpha_2 c_2}{N^2} q, \frac{\alpha_2 c_2}{N^2} (p^2 - q^2), \frac{\alpha_2 c_2}{2\sqrt{3}} \left(\frac{3}{N^2} - 1 \right) \right) \quad (3.46)$$

3.35 no.lu ileri fark denklemleri, derinlik minimizasyon denkleminin (3.28) ilk kısmı için 9 harmonikli açılım olmak üzere yerlerine yazılırsa,

$$\begin{aligned}
I(x, y) = \frac{\rho(x, y)}{N(x, y)^2} & \left(\xi_0 + \xi_1 z(x, y) + \xi_2 z(x + 1, y) + \xi_3 z(x, y + 1) + \xi_4 z(x, y)^2 + \right. \\
& \xi_5 z(x + 1, y)^2 + \xi_6 z(x, y + 1)^2 + \xi_7 z(x, y)z(x + 1, y) + \\
& \left. \xi_8 z(x, y)z(x, y + 1) + \xi_9 z(x + 1, y)z(x, y + 1) \right) \quad (3.47)
\end{aligned}$$

ifadesi elde edilir. Dikkat edilirse, 3.47 no.lu denklemde bilinmeyenler derinlik değerleri olan $z(x, y), z(x + 1, y), z(x, y + 1)$ ve bunların çarpımlarından oluşan karesel bileşenlerdir. Bu nedenle, böyle ifadelerden oluşan bir sistemin doğrusal olarak çözülemeyeceği aşikârdır. Doğrusal olmayan denklem sistemlerinin çözülebilmesi için, doğrusal olmayan eniyileştirme algoritmalarına ihtiyaç duyulur. ξ_k ifadeleri bilinen katsayılarından ve ilk adımda elde edilen aydınlatma bileşenlerinden oluşan sabitlerdir. Bu ifadeler açıkça şu şekilde gösterilebilir:

$$\xi_0 = \alpha_0 c_0 l_{0,0} N(x, y)^2 - \alpha_1 c_1 l_{1,1} N(x, y) + \alpha_2 c_2 l_{2,2} (3 - N(x, y)) \quad (3.48a)$$

$$\xi_1 = -\alpha_1 c_1 l_{1,-1} N(x, y) - \alpha_1 c_1 l_{1,0} N(x, y) + \alpha_2 c_2 l_{2,-1} + \alpha_2 c_2 l_{2,0} \quad (3.48b)$$

$$\xi_2 = \alpha_1 c_1 l_{1,-1} N(x, y) - \alpha_2 c_2 l_{2,-1} \quad (3.48c)$$

$$\xi_3 = \alpha_1 c_1 l_{1,0} N(x, y) - \alpha_2 c_2 l_{2,0} \quad (3.48d)$$

$$\xi_4 = \alpha_2 c_2 l_{2,-2} \quad (3.48e)$$

$$\xi_5 = \alpha_2 c_2 l_{2,1} \quad (3.48f)$$

$$\xi_6 = -\alpha_2 c_2 l_{2,1} \quad (3.48g)$$

$$\xi_7 = -\alpha_2 c_2 l_{2,-2} - 2\alpha_2 c_2 l_{2,1} \quad (3.48h)$$

$$\xi_8 = -\alpha_2 c_2 l_{2,-2} + 2\alpha_2 c_2 l_{2,1} \quad (3.48i)$$

$$\xi_9 = \alpha_2 c_2 l_{2,-2} \quad (3.48j)$$

Böylece, her bir piksel için oluşturulacak doğrusal olmayan denklemlerin tamamına 3.28 no.lu denklemin ikinci kısmı olan regülasyon terimleri de uygun yerlere eklenerek, MATLAB uygulamasında yer alan *Optimization Toolbox* ile sunulan araçlarla çözdürülebilir. Doğrusal olmayan denklem sistemlerinin çözümü için `fsolve` betiği kullanılmaktadır. Bu işlem için eldeki doğrusal olmayan denklem sistemine ilişkin Jacobiyen matrisinin de oluşturulması gerekmektedir. Jacobiyen matrisi, her bir fonksiyonun satırlara ve ilgili değişken cinsinden türevinin sütunlara denk geldiği bir

yapıdadır. $m \times n$ boyutundaki bir imge için k denkleminin i değişkeniyle ilişkisi şu şekilde gösterilebilir:

$$j_{k,i} = \frac{\partial f_k}{\partial x(i)} \quad (3.49)$$

Böylece Jacobiyen matrisi \mathbf{J} şu şekilde kurulur:

$$\mathbf{J} = \begin{bmatrix} j_{1,1} & j_{1,2} & \cdots & j_{1,mn} \\ j_{2,1} & j_{2,2} & \cdots & j_{2,mn} \\ \vdots & \vdots & \ddots & \vdots \\ j_{mn,1} & j_{mn,2} & \cdots & j_{mn,mn} \end{bmatrix} \quad (3.50)$$

Derinlik bilinmeyenleri vektörel indis şeklinde gösterilecek olursa şu şekilde dönüştürülebilirler:

$$z(x, y) = x(i) \quad (3.51a)$$

$$z(x + 1, y) = x(i + 1) \quad (3.51b)$$

$$z(x, y + 1) = x(i + n) \quad (3.51c)$$

Dikkat edilecek olursa her bir piksel için verilen 3.47 no.lu denklemde bilinmeyen değişken sayısı sadece üçtür. Böylece, \mathbf{J} matrisi her bir satırda sadece üç noktada değere sahip olan seyrek (*sparse*) bir matristir. Bu değerler indisler cinsinden türev alınarak gösterilecek olursa,

$$j_{k,i} = \xi_1 + 2\xi_4 x(i) + \xi_7 x(i + 1) + \xi_8 x(i + n) \quad (3.52a)$$

$$j_{k,i+1} = \xi_2 + \xi_7 x(i) + 2\xi_5 x(i + 1) + \xi_9 x(i + n) \quad (3.52b)$$

$$j_{k,i+n} = \xi_3 + \xi_8 x(i) + \xi_9 x(i + 1) + 2\xi_6 x(i + n) \quad (3.52c)$$

Böyle bir doğrusal olmayan denklem sistemi MATLAB ortamında şu şekilde çözdürülebilir:

```

1 options = optimoptions (@fsolve, 'Display', 'iter', ...
2     'Algorithm', 'trust-region-reflective', ...
3     'Jacobian', 'on');
4 xstart = zeros(m*n,1);
5 [ x, fval, exitflag, output ] = fsolve (@fun, xstart, options);

```

Bu kod bloğunda yer alan, fun tanıtıcısı, doğrusal olmayan sistem için her bir piksele ait $F(x) = 0$ fonksiyonunu ve **J** Jacobiyen matrisini üreten betiğe işaret eder. Bu betik **Ek Açıklamalar - A**'da verilmiştir.

3.7 Bilinen Derinlik Haritasından Albedo Bilgisinin Elde Edilmesi

Bir önceki bölümde tanıtılan derinlik bilgisinin kestirimi için kullanılan yöntemde, referans bir bireye ait hem derinlik hem de albedo haritalarının da elde bulunduğu varsayılmaktadır. Eğer elde referans bireye ait albedo haritası bulunmuyorsa, yani sadece iki-boyutlu portre imgesi ve derinlik haritası varsa test imgelerinin derinliklerinin kestirilmesi uygulamasını gerçekleştirmeden önce referans bireye ait eksik albedo bilgisinin elde edilmesi gerekmektedir.

Verilen bir imgeden, nesneye ait albedo bilgisinin elde edilmesi, üzerinde çokça çalışılmış bir konudur. Bu, genel olarak aynı zamanda SFS problemiyle iç içedir ya da onun bir parçasıdır (Zheng ve Chellappa, 1991; Worthington, 2005; Blake, 1985). Smith ve Hancock, (2006) yüzey normallerindeki değişiklikleri içeren bir model kurarak tek bir imgeden albedoyu elde etmeye çalışmaktadırlar. Çoğu çalışmada, önce nesnenin şekli, ardından yüzeye ait albedo elde etmeye yönelik yöntemler uygulanmaktadır (Biswas vd., 2009). Bazı yöntemlerde ise, şekil ve albedo aynı anda kestirilmektedir. Buna ek olarak, çoğu çalışmada iteratif yöntemler kullanılmaktadır. Elde edilen albedo bilgisinin doğruluğu, cisme ait şeklin yani derinlik bilgisinin ne kadar doğru elde edildiğiyle son derece ilintilidir.

Brooks ve Horn (1985), SFS problemini, verilen bir E imgesi ve R yansım haritası üzerinden $z(x, y)$ yüzeyinin bulunması olarak tanımlamaktadırlar. Burada imge parlama denklemi E 'nin Ω tanım kümesinde gerçekleşmek durumundadır:

$$E(x, y) = R \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y} \right) \quad (3.53)$$

Ayrıca, $z(x, y)$ üzerinde verilen herhangi bir sınır kısıtı da gerçekleşmek zorundadır. Bu problem, birinci dereceden kısmi diferansiyel denklem şeklini almaktadır (Horn, 1970). Yansım haritası, yüzeyin yansım özelliği ve ışık kaynaklarının yoğunluk ve dağılım bilgilerini kapsar. Eğer yüzeyin Lambert yansım özelliği gösterdiği ve tek bir noktasal ışık kaynağı tarafından aydınlatıldığı varsayılırsa, \mathbf{s} vektörü, yönü kaynağa doğru olan ışık vektörünü ve $\mathbf{n}(x, y)$ de yüzey parçasına ait birim normal vektörü tanımlamak üzere, yüzey parçasının görülen parlaklığı, yansım haritası olarak verilir:

$$\mathbf{R}_s(\mathbf{n}(x, y)) = \mathbf{n}(x, y) \cdot \mathbf{s} \quad (3.54)$$

Brooks ve Horn (1985), bunu bir en aza indirme problemi olarak ele almakta ve iteratif bir yol izlemektedirler. En aza indirilmeye çalışılan ifade:

$$\iint_{\Omega} \left(E(x, y) - \mathbf{n}(x, y) \cdot \mathbf{s} \right)^2 dx dy \quad (3.55)$$

ve regülasyon ifadesi de,

$$\iint_{\Omega} \left(\mathbf{n}_x^2(x, y) + \mathbf{n}_y^2(x, y) \right) dx dy \quad (3.56)$$

şeklinde verilmektedir. Burada vektörlerin karesi kendisiyle nokta çarpımı anlamına gelmektedir. Ayrıca yüzey normallerinin birim uzunlukta olmaları da bir diğer kısıttır. Bu üç ifadenin birleşimi olan,

$$\mathbf{I}(\mathbf{n}, \mathbf{s}) = \iint_{\Omega} \left[(E - \mathbf{n} \cdot \mathbf{s})^2 + \lambda(\mathbf{n}_x^2 + \mathbf{n}_y^2) + \mu(x, y)(\mathbf{n}^2 - 1) \right] dx dy \quad (3.57)$$

ifadesi \mathbf{n} ve \mathbf{s} 'ye göre minimize edilmeye çalışılmaktadır.

Eğer \mathbf{n} değeri verildiğinde \mathbf{s} 'ye göre minimize edilmek istenirse,

$$\mathbf{s} = \left[\iint_{\Omega} \mathbf{nn}^T dx dy \right]^{-1} \iint_{\Omega} E\mathbf{n} dx dy \quad (3.58)$$

ifadesine erişilebilir. Burada \mathbf{nn}^T , 3×3 boyutunda bir matrisdir. İteratif yöntemin son aşamasında bu formülün ayrık zamanda integral yerine toplam sembolüyle ifade edilmesinin ardından \mathbf{s} kaynak vektörü şu şekilde hesaplanabilir (Brooks ve Horn, 1985):

$$\mathbf{s}^k = \left[\sum_{i,j \in \Omega} \mathbf{n}_{ij}^k \mathbf{n}_{ij}^{kT} \right]^{-1} \sum_{i,j \in \Omega} E_{ij} \mathbf{n}_{ij}^k \quad (3.59)$$

Özetle, verilen bir yüzey şekli üzerinde yeni bir kaynak doğrultusu tek seferde imge üzerinde iterasyona gerek kalmaksızın hesaplanabilir. 3×3 boyutlu bir matris olan \mathbf{nn}^T ifadesi ve $E\mathbf{n}$ vektörü imgenin tamamı üzerinde işlenerek toplanır. Kaynak yönü, daha sonra eliminasyon yöntemiyle bulunabilir. Teorik olarak bu problemin çözümü sadece üç yüzey noktasına ilişkin parlaklık bilgisinden elde edilebileceği ve burada bütün imge kullanıldığı için oldukça artık olarak belirtilmiş bir denklem sistemidir (*over-determined*). Verilen veride gürültü yoksa ya da oldukça azsa bu yöntemle elde edilen kaynak doğrultusu oldukça doğru çıkmaktadır (Brooks ve Horn, 1985).

3.8 Aydınlatma Şartlarının Kestirimi Yöntemleri

Bir imgenin hangi aydınlatma şartı altında elde edildiği bilgisinin çok sayıda bilgisayarla görü probleminin çözülmesinde önemli bir girdi olması ve bazen de bilinmesinin elzem olması nedeniyle bu problem, üzerinde çokça durulmuş ve çalışılmış bir konudur. Bundan ötürü, bu tez çalışmasında aydınlatma şartlarının kestirimi konusu ayrıca ele alınmıştır. Bahsi geçen problemlerden önemli bir tanesi ve bu tez çalışmasının da ana konusu olan yüzey karakteristiğinin ve derinliğin kestirimi probleminin girdilerinden bir tanesi aydınlatma kaynaklarının üç boyutlu uzaydaki yeri ve şiddetidir (Arslan ve Seke, 2017).

Bu problemin çözülmesinin oldukça zor ve çoğu zaman eksik tanımlı bir problem şeklinde ifade edildiği bilinmektedir. Eksik tanımlı olmasının nedeni yeterli bilginin tek bir imgede bulunmayışıdır. Araştırmacılar bu problemi çözebilmek ya da en azından gerçek

bilgiye olabildiğince yakın bir şekilde kestirebilmek için birtakım varsayımlar, regülarizasyon ve optimizasyon yöntemleri kullanılmışlardır.

Benzer problemlerde olduğu gibi, bu uygulamada da varsayılan en temel nokta nesnelerin Lambert yansıma kanununa göre davranış sergiledikleri, yani yayınlık (*diffuse*) yüzey oldukları, ışığı ayna gibi (*specular*) yansıtmadıklarıdır. Bazı çalışmalarda ise bu iki yansıma modelinin de hibrit şekilde kullanıldığı görülmektedir. Diğer varsayımlardan birisi de nesnelerin olabildiğince dışbükmümlü olduğu, dolayısıyla gölgelemeler içermediğidir. Gölgeler de iki türlü kabul edilmektedir: döküm gölge (*cast shadows*) ve eklenik gölge (*attached shadows*). Son olarak çoğu zaman nesnelerin albedolarının da sabit ve eşdağılımlı olduğu varsayılmaktadır.

Bu uygulamada cevabı aranan sorular şunlardır: Aydınlatma kaynaklarının sayısı (genelde tek kaynak olduğu varsayılmaktadır); bu kaynakların türü, nokta kaynak (*point source*) olup olmadıkları; bu kaynakların şiddeti; ve son olarak kaynakların yerleşimleri. Araştırılan parametreler (x, y, z) koordinat sisteminde üç adet olmakla birlikte, birim vektör varsayımı altında bunlar açısız parametrelere dönüştürülmekte ve böylece sayıları ikiye düşmektedir. Bunların ilki, imge düzleminde paralel yer alan (xy düzlemi) eğikliği (*tilt*) ya da azimuth açısıdır. Diğeri ise yükseltiyi belirten verev (*slant*) açısıdır.

3.8.1 Yöntemler

Bu bölümde, aydınlatma kaynağının kestirimi konusunda şu ana kadar geliştirilmiş belli başlı yöntemler ve algoritmalar incelenmektedir. İlk kez Pentland tarafından 1982 yılında formüle edilen yaklaşımla birlikte, bu yöntemi geliştiren ya da eksik yönlerini tamamlayıcı yöntemler göze çarpmaktadır.

3.8.1.1 Pentland, 1982

1970'lerde ilk kez Horn tarafından ortaya atılan ve zamanla geliştirilen SFS yöntemi ışık kaynağının bilindiği varsayımı üzerine kuruluydu (Horn, 1970). İlk defa, Pentland 1982'de verilen bir imgenin aydınlatma şartlarını teorik olarak bulmaya yönelik bir çalışma yayınladı (Pentland, 1982). Bu çalışmada, yüzey oryantasyonunu bilmeden önce aydınlatıcı yönünün bulunmasının gerektiği söylenmekte ve insan görsel algılamasının bunu nasıl gerçekleştirdiği sorusu üzerinde durulmaktadır. Yanıt olarak da girdi bilgisi olarak yüzey parlaklıklarından ziyade bu değerlerin birinci türevlerinin etkili olduğu söylenmektedir. Buna kanıt olarak da şu iki noktaya dikkat çekilmektedir:

1. Biyolojik görme sistemleri, imgelerdeki mutlak parlaklıklara duyarsızdır fakat bu değerlerin değişimlerine duyarlıdır.
2. Bir imge parlaklığının birinci türevi, sadece bir integral sabiti hariç olmak üzere, mutlak imge yoğunluğunun bütün bilgisine sahiptir.

İkinci noktada bahsi geçen sabit, ortalama aydınlanmaya karşılık gelir ve bu değer aydınlanmanın yönünden bağımsızdır. Böylece, her iki değer de (mutlak yoğunluk ve birinci türevler) aydınlanma yönü söz konusu olduğunda eşit oranda yararlı bilgiye sahip oldukları söylenmektedir. Pentland, çalışmasını bundan hareketle türevler üzerinden gerçekleştirdiğini ifade etmektedir. Geliştirdiği algoritmayı test ederken insanlarla da çalışmış ve aydınlanma yönlerinin kestiriminde her ikisinin de (hem algoritmanın hem de insan yanıtlarının) belirgin bir korelasyon gösterdiğini, yani ikisinin de beraber doğru tahminde ya da yanlış tahminde bulunduğunu, bunun da bu yöntemin biyolojik görsel sistemin kullandığı girdileri ve ipuçlarını eşit derecede kullandığına işaret ettiğini iddia etmektedir.

Pentland, yüzey doğrultularının eşyüzeysel olarak dağıldıklarını farzettği için istatistiksel bir yaklaşımla *En Büyük Olabilirlik Kestirimi (ML)* yöntemini kullanmaktadır. Pentland'ın ön kabulleri ise, bütün yüzeylerin Lambert yansıma kanununa göre davranış sergilemeleri, bütün aydınlatmanın tek bir uzak ışık kaynağından gelmesi ve nesnelere üzerindeki yüzey normallerinin izotropik olarak dağılmakta olmasıdır.

3.8.1.2 Lee ve Rosenfeld, 1985

Lee ve Rosenfeld, Pentland'ın yöntemine oldukça benzer fakat birkaç yönüyle farklı bir algoritma ortaya koymuşlardır (Lee ve Rosenfeld, 1985). Farklılıklardan bir tanesi, koordinat sistemini bir eksenini aydınlatma vektörü doğrultusuna gelecek şekilde dönüştürmeleridir. Ayrıca, Pentland'ın aksine yüzey yapısının yüksek dereceden türevli yaklaşımlarından kaçınılmaktadırlar. Birincil türevlerin (I_x ve I_y) ne anlama geldiklerini incelemektedirler. Oluşturdukları koordinat sisteminde, iki yüzey noktasının sadece ve sadece imge üzerinde aynı gradyene sahip olduklarında aynı eğiklik açısına sahip olabileceğini kanıtlamaktadırlar. Bu kısım, SFS problemiyle ilgilidir. Aydınlatma yönünün bulunması için, Pentland'ın eğiklik açısı kestiriminin başarılı olduğunu belirtmekle birlikte veriv açısının iyileştirilmesi için yeni bir yaklaşım ortaya koymaktadırlar.

3.8.1.3 Brooks ve Horn, 1985

Bu çalışmada, yazarlar hem SFS problemini hem de aydınlatma kaynağını kestirme problemini iteratif bir şekilde çözmeye çalışmaktadırlar (Brooks ve Horn, 1985). SFS

problemini çözmeye çalışan bilinen çoğu yöntem, yüzey normallerinin türevlerini imge yoğunluğuna haritalayan yansıma haritasının (*reflectance map*) bilinmesini de gerektirmektedir. Bu çalışmada ise bu haritanın elde olmayışı durumu da göz önüne alınmaktadır. Bunu yaparken, bilinmeyen parametrelere sahip ama formu bilinen bir harita olduğu varsayılır. Örneğin, yüzeyin Lambert kuralına bağlı olduğu biliniyor fakat ışık kaynağının yönü bilinmiyorsa böyle bir durum söz konusudur. Ayrıca, yansıma haritasının gradyenler ya da stereografik koordinatlar yerine birim normaller cinsinden parametreleştirilmesi analizi kolaylaştırmaktadır. Bu iteratif algoritmada, yüzey normallerinin her bir adımdaki kestirimi ışık kaynağına doğru ya da ondan uzağa doğru evrilmektedir. Bu ayarlamaların derecesi, kestirilen imge yoğunluğu ile gözlemlenen yoğunluk arasındaki farkla orantılıdır.

İncelenen yöntemler arasında, bu problemi iki yönüyle de iteratif şekilde inceleyen tek çalışma budur. Bir yoğunluk hata terimi, bir regülarizasyon terimi ve bir Lagrange çarpımından oluşan bir fonksiyonel minimize edilmeye çalışılmaktadır. Buna karşılık gelen Euler denkleminin ikinci dereceden bir diferansiyel denklem olduğu (bilinmeyen yüzey normali fonksiyonu) görülmektedir. Ayırık domende iteratif bir algoritma ile çözülebilmektedir.

Bu yöntemin bir yararı da, eğer yüzey şekli biliniyorsa, ışık kaynağı yönünün bir kapalı form şeklinde belirlenebilir olmasıdır. İterasyonun herhangi bir adımında şu anki yüzey şekli kestiriminden kaynak yönü kestirimi elde edilebilir. Giderek daha doğru yüzey şekli kestirimi ile ışık kaynağı yönü kestirimi bir birine geçmiş durumdadır.

3.8.1.4 Zheng ve Chellappa, 1991

Bu yöntemde, aydınlatma kaynağının kestirimi yanı sıra albedo kestirimi de sunulmaktadır. Kaynağın eğiklik açısı kestirimi iki farklı yöntemle yapılabilir: lokal kestirimlerin ortalamasını kullanan yöntem ve imge konturları üzerinde yer alan bilgiyi kullanan ikinci bir yaklaşım. Verev açısı ve albedo kestirimleri de imge istatistikleri ile yapılmaktadır. Bu yöntem, Lee ve Rosenfeld yöntemindeki verev açısı kestirimine oldukça benzemekle birlikte, imgenin gölgede kalan kısmının da işleme dahil edilmesiyle farklılaştırılmış bir halidir (Zheng ve Chellappa, 1991).

3.8.1.5 Chojnacki, Brooks, Gibbins, 1994

Bu çalışmada, Pentland yönteminin hatalı olduğu gösterilmekte ve bunun düzeltilmesi için bir öneri ortaya konulmaktadır (Chojnacki vd., 1994). Buna göre, Pentland'ın yönteminde verev açısı kestirimi yanlış yapılmakta ve yeni bir kestirim tekniği önerilmektedir. Özellikle bir imgenin çözünürlüğü arttıkça verev açısı kestirimindeki hata

da artmaktadır. Bunu düzeltmek üzere, *Disk Yöntemi* adı verdikleri yeni bir yöntem sunmaktadırlar.

3.8.1.6 Attick, Griffin, Redlich, 1996

Bu çalışmada esasen SFS problemine istatistiksel bir yaklaşım sergilenmektedir. Buna göre, imgeler daha az parametreden oluşan uzaya indirgenmekte ve bu uzayda parametre kestirimi gerçekleştirilmektedir. Üç boyutlu insan yüzleriyle yürütülen bu çalışmada, PCA (*Temel Bileşenler Analizi*) uygulanmaktadır. Işık kaynağı kestirimi için sundukları teknik yöntem, çalışmanın ekler kısmında yer almaktadır (Atick vd., 1996).

3.8.1.7 Zhao, 1999

Zhao, doktora teziyle geliştirdiği simetrik SFS algoritmasını aydınlatma kaynaklarına da taşımıştır (Zhao ve Chellappa, 2001). Bunu, SFS yönteminde sunduğuna benzer şekilde bir *self-ratio* imge üzerinden uygulamaktadır.

3.8.2 Yöntemlerin karşılaştırılmaları

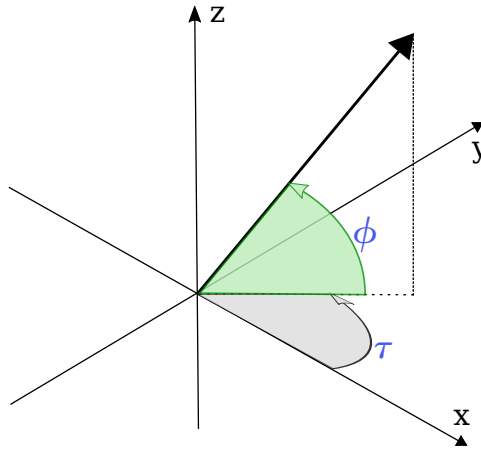
Zhao ve Chellappa, ilk üç yöntemi de uyguladıklarını, en basit ve kabul edilebilir sonuçlar veren Lee ve Rosenfeld algoritması olduğunu, bunun hem simüle edilmiş hem de gerçek yüz imgeleri için geçerli olduğunu söylemektedirler (Zhao ve Chellappa, 2000).

Yöntemleri karşılaştıran çalışmalara bakıldığında, hemen hemen bütün yöntemlerin eğiklik açısını kestirmede başarılı oldukları ve verev açısının genelde tatmin edici olmayan şekilde kestirildiği ve bunun iyileştirilmeye çalışıldığı vurgulanmaktadır.

Mingolla ve Todd, şimdiye kadar geliştirilen SFS yöntemlerinin insanın bilişsel algılamalarına uygun olmadığını söylemektedirler (Mingolla ve Todd, 1986). Gerçekleştirdikleri bir dizi testin ardından, uyumsuz gördükleri en önemli iki varsayımın nesnelerin Lambert kuralına göre davranış sergilemeleri ve başarılı bir yüzey kestirimi için ışık kaynağı yönünün bu işlemde önce kestirilmesi olduğunu söylemektedirler. Buna göre, insan görsel sistemi bu varsayımlar olmadan da derinlik bilgisini kestirebilmekte ve bunun başarılmasında bahsedilen varsayımların bir etkisi olmamaktadır. Gibbins, vd. (1991) ise bu yaklaşıma karşı çıkmaktadırlar. Onlara göre, Brooks ve Horn yönteminin başarılı bir şekilde iteratif olarak hem SFS hem de kaynak problemini çözebilmesi bunu geçersiz kılmaktadır ve Mingolla ve Todd'un iddiasının yeterli kanıtlara sahip olmadığını vurgulamaktadırlar. Işık kaynağı kestiriminin önemini dile getirirler de, bahsi geçen yöntem ve tekniklerin başarılarının, insan algılayışıyla karşılaştırıldığında oldukça sınırlı olduğunu itiraf etmekten de geri durmamaktadırlar.

Bir noktasal aydınlatma kaynağının üç boyutlu koordinat sisteminde iki parametre ile ifade edilmesi kolaylık ve karşılaştırma imkanı sağlamaktadır. Küresel parametreler cinsinden böyle bir dönüşüm Şekil 3.4'da görülebilir ve aşağıdaki denklem ifadesi ile gerçekleştirilebilir. Bu parametreler yukarıda da bahsi geçen, τ ve ϕ ile gösterilen, sırasıyla eğiklik (*tilt*) ve verev (*slant*) açılarıdır.

$$(x, y, z) = (\cos\phi \cos\tau, \cos\phi \sin\tau, \sin\phi) \quad (3.60)$$



Şekil 3.4 Bir noktasal aydınlatma kaynağı doğrultusu için azimuth ve yükselti parametreleri.

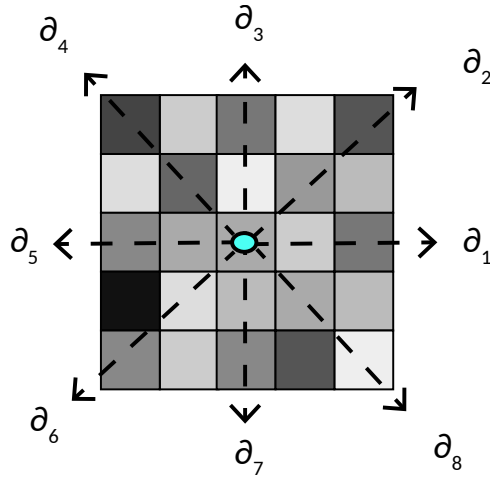
3.8.2.1 Eğiklik açısı kestirimi

Pentland ışık kaynağı eğiklik (*tilt*) açısı kestirimini birden fazla yönde aldığı ortalama parlaklık değişimleri cinsinden ifade etmiştir (Pentland, 1982). Her ne kadar çalışmasında hangi yönleri kullandığı belirtilmese de, bu tez araştırmasında kullanılan 8 farklı yön Şekil 3.5'te görülebilir. Yüzey normallerinin bir nesne için isotropik olarak dağılım gösterdiği varsayımı altında, imge üzerindeki bütün noktalarda ve yönlerde $d\mathbf{N}$ yüzey normali türevleri toplamının beklenen değeri sıfır olacaktır. Yani,

$$E \left(\sum_{x,y,\theta} d\mathbf{N} \right) = 0. \quad (3.61)$$

Bir imge yönü (dx_i, dy_i) boyunca yer alan bölgedeki parlaklık değişimleri dI ortalamasına $d\tilde{I}_i$ ve aydınlatma vektörü yönüne $\mathbf{L} = (x_L, y_L, z_L)$ denirse, regresyon modeli şu şekilde verilmektedir:

$$\begin{pmatrix} d\tilde{I}_1 \\ d\tilde{I}_2 \\ \vdots \\ d\tilde{I}_n \end{pmatrix} = \begin{pmatrix} dx_1 & dy_1 \\ dx_2 & dy_2 \\ \vdots & \vdots \\ dx_n & dy_n \end{pmatrix} \begin{pmatrix} \hat{x}_L \\ \hat{y}_L \end{pmatrix} \quad (3.62)$$



Şekil 3.5 Bir gri seviye sayısal imge kesitinde yer alan 8 farklı değişim yönü.

Yön matrisi β ile ifade edildiğinde \hat{x}_L ve \hat{y}_L 'nin en büyük olabilirlik kestirimi (ML) şu en az kareler regresyon modeli ile verilmektedir:

$$\begin{pmatrix} \hat{x}_L \\ \hat{y}_L \end{pmatrix} = (\beta^T \beta)^{-1} \beta^T \begin{pmatrix} d\tilde{I}_1 \\ d\tilde{I}_2 \\ \vdots \\ d\tilde{I}_n \end{pmatrix} \quad (3.63)$$

Son olarak, eğiklik açısı kestirimi şu şekildedir:

$$\tau = \tan^{-1} \left(\frac{\hat{y}_L}{\hat{x}_L} \right) \quad (3.64)$$

Lee ve Rosenfeld sadece iki yön kullanmaktadırlar, bunlar imge türevleri I_x I_y 'dir (Lee ve Rosenfeld, 1985). Böylece, yöntemleri Pentland'ın yönteminin özel bir durumuna dönüşmektedir (Zheng ve Chellappa, 1991). Zheng ve Chellappa'nın Yerel Oylama Yöntemi imge üzerindeki her bir nokta için parlaklığın sıfır olmadığı yerlerde eğiklik açısı için yerel kestirimleri hesaplar. Bu yöntemle azimuth şu şekilde hesaplanır:

$$\tau = \arctan \left(\frac{E_{x,y} \left\{ \frac{\hat{y}_L}{\sqrt{\hat{x}_L^2 + \hat{y}_L^2}} \right\}}{E_{x,y} \left\{ \frac{\hat{x}_L}{\sqrt{\hat{x}_L^2 + \hat{y}_L^2}} \right\}} \right) \quad (3.65)$$

3.8.2.2 Verev açısı kestirimi

Pentland'ın verev açısı kestirimi imge parlaklık değişimleri varyansını kullanmakta ve şu şekilde verilmektedir:

$$\phi = \arccos \sqrt{\left(1 - (\hat{x}_L^2 + \hat{y}_L^2) / k^2\right)} \quad (3.66)$$

Buradaki k ifadesi ise şu şekildedir:

$$k = \sqrt{\left(E\{dI^2\} - (E\{dI\})^2\right)} \quad (3.67)$$

Lee ve Rosenfeld'in yönteminin Zheng ve Chellappa tarafından değiştirilmiş hali şu şekildedir:

$$\frac{E\{I\}}{\sqrt{E\{I^2\}}} = \frac{4\sqrt{2}}{3\pi} \cdot \frac{(\pi - \phi) \cos \phi + \sin \phi}{1 + \cos \phi} \quad (3.68)$$

Yerel yöntemin verev açısı kestirimi, Lee ve Rosenfeld yöntemine benzemektedir, fakat fonksiyonlar farklıdır. Yerel yöntemin sonucu $\cos \phi$ 'nin yedinci-dereceden polinomları cinsinden yaklaşıklığıdır:

$$\begin{aligned} \frac{E\{I\}}{\sqrt{E\{I^2\}}} &= 0.5577 + 0.6240 \cos \phi + 0.1882 \cos^2 \phi \\ &\quad - 0.6514 \cos^3 \phi - 0.5350 \cos^4 \phi + 0.9282 \cos^5 \phi \\ &\quad + 0.3474 \cos^6 \phi - 0.4984 \cos^7 \phi \end{aligned}$$

Bu ifadede, $E\{I\}$ ve $\sqrt{E\{I^2\}}$ sırasıyla imge parlaklıklarının toplu ortalamaları ve imge parlaklıklarının kareleridir (Zheng ve Chellappa, 1991).

3.9 Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme

Bilgisayarın icat edilmeye başlandığı 1950’li yıllardan beri makinalara öğrenme kazandırmaya çalışan yöntemler geliştirilmeye devam etmektedir. Bu geniş alanın başlıca alt faaliyetleri, örüntü tanıma, sezgisel programlama, doğal dil işleme ve bilgisayarla görmedir (Nilsson, 2011). Kavramların, bilgisayarların ihtiyaç duyduğu biçimsel kuralların insan eliyle girilmesine gerek kalmadan, çok katmanlı bir hiyerarşi içerisinde bilgisayarlar tarafından öğrenilmesi yaklaşımına *derin öğrenme* denir. Derin öğrenme, yapay zekâ yaklaşımlarından biridir. Bilgisayarların veri ve deneyimlerden yararlanarak kendilerini geliştirdikleri bir makine öğrenmesi türüdür (Goodfellow vd., 2018). İnsan beyinde gerçekleşen öğrenme modelinin nasıl gerçekleştiğinin araştırıldığı yıllarda Rosenblatt tarafından geliştirilen perseptrona dayalı *Yapay Sinir Ağları*, derin öğrenmenin öncülüdür.

3.9.1 Perseptron ve yapay sinir ağları

İki kategorili bir sınıflandırma işleminde $y(\mathbf{x})$ ayırtaç fonksiyonu çıktısına bağlı olarak \mathbf{x} vektörü şu şartlara göre atanabilir:

$$\begin{aligned} \text{Eğer } y(\mathbf{x}) > 0 \text{ ise } C_1 \text{ sınıfına ata,} \\ \text{Eğer } y(\mathbf{x}) < 0 \text{ ise } C_2 \text{ sınıfına ata.} \end{aligned} \quad (3.69)$$

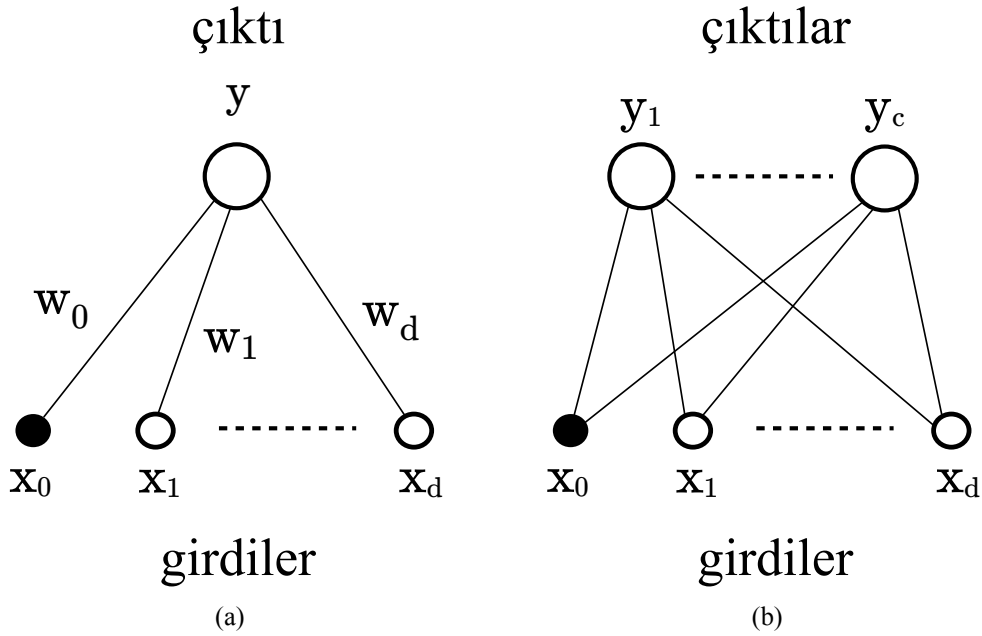
En basit ayırtaç fonksiyonu seçimi, \mathbf{x} 'in bileşenleri türünden doğrusal olan ve böylece şu şekilde yazılabilir fonksiyondur:

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \quad (3.70)$$

Bu denklemde, d boyutlu \mathbf{w} vektörü *ağırlık vektörü*, w_0 parametresi *yanlılık* olarak adlandırılır (Bishop, 1995). Şekil 3.6a'da bir doğrusal ayırtaç fonksiyonu sinir ağı olarak gösterilmiştir. Şekildeki her bir bileşen, doğrusal ayırtaç ifadesindeki bir değişkeni temsil eder. Yanlılık ifadesi w_0 , ekstra bir girdinin ağırlık parametresi olarak düşünülebilir. Böyle bir durumda, x_0 ifadesi kalıcı olarak $+1$ değerini alır. Doğrusal ayırtaçlar kolaylıkla c adet sınıf içeren bir durum için uygulanabilir. Bu durumda her bir \mathcal{C}_k sınıfı için bir $y_k(\mathbf{x})$ fonksiyonu kullanılır:

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (3.71)$$

Yeni bir \mathbf{x} noktası, eğer bütün $j \neq k$ durumları için $y_k(\mathbf{x}) > y_j(\mathbf{x})$ şartı sağlanıyorsa \mathcal{C}_k sınıfına atanır (Bkz. Şekil 3.6b).



Şekil 3.6 Bir doğrusal ayırtaç fonksiyonunun sinir ağı şeklinde gösterimi. (a) 2-sınıflı durum, (b) çok sınıflı durum.

Perseptron ya da *algılayıcı*, yapay sinir ağlarının temel ünitesidir. Her bir x_j ($j = 1, 2, \dots, d$) girişi için bir w_j katsayısı (*bağlantı ağırlığı*) tanımlıdır. y çıkışı girdilerin ağırlıklı toplamı olarak hesaplanır ve 3.70'da gösterildiği gibidir. *Öğrenme*, bütün girdiler için bu sistemin doğru çıktı üretmesi amacıyla \mathbf{w} ağırlıklarının belirlenmesi işidir (Alpaydın, 2013). Doğrusal olmayan durumları ve daha karmaşık ilişkileri öğrenmede, *saklı*

katmanlara sahip ve ileriye doğru yayılımı olan ağ yapıları kullanılmaktadır. Her bir katmanda, birden fazla perseptron bulunur ve bunların girdilerini bir önceki katmanda yer alan perseptronların çıktıları oluşturur. Çok katmanlı ağın eğitilmesinde hata geri yayma algoritması çalıştırılır.

3.9.2 Aktivasyon fonksiyonları

Derin ileri besleme (ya da çok katmanlı algılayıcı) ağlarında, öğrenilmek istenen f fonksiyonu tamamen doğrusal olursa, tüm ağın çıktısı da girdisine göre doğrusal kalacaktır. Böyle bir durum, çoğu zaman gerçek verinin kullanıldığı durumlarda istenmez. Bunun önüne geçmek için, doğrusal olmayan fonksiyonlar kullanılmalıdır. Çoğu yerde bunu gerçekleştirmek için önce öğrenilen parametrelere bağlı bir ilgin (*afin*) dönüşüm uygulanır. Ayrıca doğrusal olmayan ve sabit kalan bir aktivasyon ya da aktifleştirme fonksiyonu devreye sokulur (Goodfellow vd., 2018). Modern sinir ağlarında en çok kullanılan aktifleştirme fonksiyonları Çizelge 3.2’de gösterilmiştir.

Çizelge 3.2 Derin öğrenmede sıkça kullanılan aktifleştirme fonksiyonları.

Düzeltilmiş doğrusal birim (ReLU: Rectified linear unit)	$g(z) = \max\{0, z\}$
Lojistik sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$
Hiperbolik tanjant	$g(z) = \tanh(z) = \frac{e^{2z}-1}{e^{2z}+1}$
Softplus	$g(z) = \log(1 + e^z)$

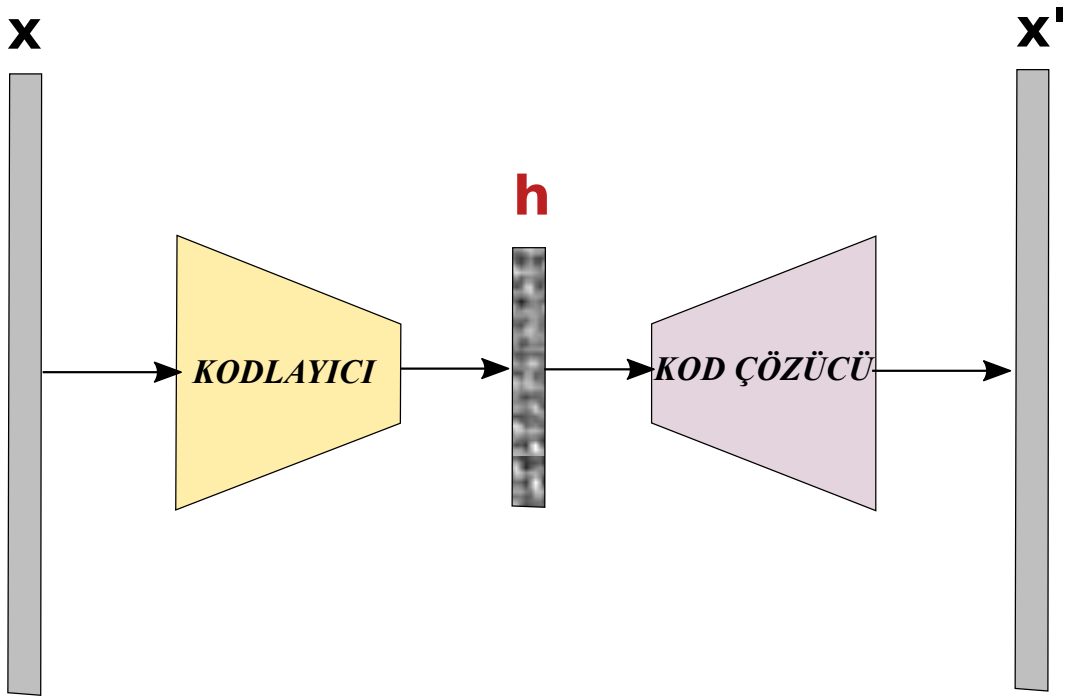
Ayrıca n farklı değere sahip ayrık bir değişken için olasılık dağılımı tanımlanmak istendiğinde *softmax* fonksiyonu kullanılabilir:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (3.72)$$

3.9.3 Evrişimsel ağlar

Evrişimsel sinir ağları veya CNN'ler (*Convolutional neural networks*), katmanlarında genel matris çarpımları yerine en az birinde evrişim operasyonu içeren filtrelerden oluşan yapay sinir ağlarına verilen addır. Doğrusal bir işlev olan evrişimle birlikte biriktirme (*pooling*) operasyonunu da kullanırlar. Bu ağ mimarileri, biyolojiden ve hayvanların görsel sistemlerinden esinlenmiş ve derin öğrenme uygulamalarında son derece başarılı olmuştur (Goodfellow vd., 2018). Derin öğrenme yarışmalarında özellikle ImageNet nesne tanıma yarışmasını kazanmasıyla ilgiyi artan bir şekilde üzerine çekmiştir (Krizhevsky vd., 2012).

3.9.4 Otokodlayıcılar



Şekil 3.7 Otokodlayıcı yapısı.

Otokodlayıcılar (*autoencoder*), kendisine girdi olarak verilen bilgiyi çıktıda tekrar oluşturmak üzere geliştirilen yapay sinir ağlarıdır. İçlerinde bir gizli katman ve bunun çıktısında kodlanmış **h** verisi bulunur. İki parçalı bir yapı olan otokodlayıcıda, ilk olarak girdi veriyi kodlayan *kodlayıcı* (encoder), ardından da bu kodlanmış veriyi çözen *kodçözücü* (decoder) yer alır. Otokodlayıcılar genellikle kodlamayı yaklaşık olarak yaparlar. Verinin yararlı özniteliklerini öğrenmek üzere geliştirilmişlerdir, ve böylece boyut düşürme görevlerinde kullanılabilirler. Genel bir otokodlayıcı yapısı Şekil 3.7'de gösterilmiştir.

En yalın halinde, bir otokodlayıcı \mathbf{x} 'i girdi olarak alır ve bir $f(\cdot)$ haritalama fonksiyonunu uygular. Eğer gizli katman gösterimi \mathbf{h} ile ifade edilecek olursa bu ifade şu şekilde gösterilebilir:

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (3.73)$$

Burada \mathbf{W} ve \mathbf{b} terimleri sırasıyla sinir ağı katmanının ağırlık matrisini ve yanlılık değerlerini göstermektedir. Kodçözücü bir başka haritalama uygulayarak \mathbf{h} 'yi girdi olarak alır ve \mathbf{x}' 'nü geri oluşturur:

$$\mathbf{x}' = f'(\mathbf{W}'\mathbf{h} + \mathbf{b}') \quad (3.74)$$

Bu temel yapı yığın hale getirilerek daha derin yapılar geliştirmek mümkündür. Otokodlayıcıların seneler içinde birden fazla türü geliştirilmiştir. Bunlar arasında, seyrek otokodlayıcılar (*sparse autoencoders*), arıtan otokodlayıcılar (*denoising autoencoders*), daraltan otokodlayıcılar (*contractive autoencoders*), ve üretken bir model olan değişimsel otokodlayıcılar (*variational autoencoders*) yer almaktadır (Goodfellow vd., 2018).

3.9.5 Üretken ağlar

Derin üretken ağlar, genel olarak olasılık dağılım fonksiyonlarını tanımlayarak bunlardan örnekler alan ve veri oluşturabilme yeteneğine sahip yapılardır. Bunlar arasında makine öğrenmesi araştırmaları tarihi boyunca geliştirilmiş olan, Boltzmann Makinesi, Kısıtlandırılmış Boltzmann Makineleri, Derin Boltzmann Makineleri, Derin İnanç Ağları (*Deep belief networks, DBN*), Evrişimsel Boltzmann Makineleri ve Değişimsel Otokodlayıcılar sayılabilir. DBN'ler evrişimsel olmayan ilk modellerden birisi olarak kabul edilir (Goodfellow vd., 2018).

Bu tez çalışmasında kullanılan Üretken Çekişmeli Ağ türevlenebilir üretken model tabanlı bir başka üretken model yöntemidir (Goodfellow vd., 2014). Üretken çekişmeli ağlarda iki farklı ağ bir yarışmada karşı karşıya gelen rakipler olarak tanıtılır. Buna göre, *üreteç ağ* $\mathbf{x} = g(\mathbf{z}; \theta^{(g)})$ şeklinde ifade edilen örnekler üretir. *Ayırtıcı ağ* ise, eğitim setinden gelen örneklerle üreteçten gelen örnekleri ayırt etmeye çalışır. Burada $d(\mathbf{x}; \theta^{(d)})$ olasılık değerini çıkartır. GAN'larda öğrenme sıfır toplamlı bir oyun olarak tanımlanır. Öğrenme

boyunca her iki ağ da eniyileştirmelerini maksimuma getirmeye çalışır. Bu oyunun yakınsaması durumunda:

$$g^* = \arg \min_g \max_d v(g, d). \quad (3.75)$$

olmaktadır. v için genellikle şu ifade seçilir (Goodfellow vd., 2018):

$$v(\boldsymbol{\theta}^{(g)}, \boldsymbol{\theta}^{(d)}) = \mathbb{E}_{x \sim p_{data}} \log d(\mathbf{x}) + \mathbb{E}_{x \sim p_{model}} \log [1 - d(\mathbf{x})]. \quad (3.76)$$

$p(\mathbf{x})$ olasılık dağılımı yerine $p(\mathbf{x}|\mathbf{y})$ dağılımını kullanarak koşullandırılmış GAN'lar eğitmek de mümkündür. Bunun neticesinde, üreteç ağına rasgele gürültü yerine istenilen dönüşümü sağlayacak gerçek bilgi girdi olarak sokulabilir. Koşullu GAN'ın kayıp fonksiyonu şöyle ifade edilir (Isola vd., 2017):

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log (1 - D(x, G(x, z)))]. \quad (3.77)$$

4. MATERYAL VE YÖNTEM

Bu ana bölümde, bu tez çalışmasında kullanılan materyaller, kullanılan yazılım kütüphaneleri, ayrıca geliştirilen ve önerilen yeni yöntemler üzerinde durulmaktadır. Derinlik bilgisinin kestirimi yöntemlerinin geliştirilmesi ve test edilmesinde, diğer bir çok bilgisayarla görü probleminin araştırılmasında olduğu gibi mutlaka daha önceden başka kurumlar ya da üniversiteler tarafından hazırlanmış ve derinlik bilgilerinin mesafe ölçücü aygıtlar tarafından doğruya en yakın şekilde ölçülerek derinlik haritalarının oluşturulduğu ve içine dahil edildiği veri setlerine ihtiyaç duyulmaktadır. Bu tez çalışmasında da bu şekilde üretilmiş, bireylere ait yüz imgelerini ve bunlara karşılık gelen derinlik haritalarını içeren iki adet veritabanı kullanılmıştır. Bu veritabanları hakkında bilgiler Bölüm 4.1’de verilmektedir. Bu tez çalışmasında tek bir imge kullanarak derinlik kestirimi probleminin çözümüne yönelik geliştirilen derin öğrenme tekniklerine dayalı yeni bir yöntem Bölüm 4.2’de tanıtılmaktadır.

4.1 Uygulama Veritabanları

Bu tez çalışmasında incelenen ve testleri yapılan yöntemlerin ve algoritmaların uygulanmasında, yeni yöntemlerin geliştirilmesinde ve bunların karşılaştırılmasında kullanılan insan yüz imgeleri içeren veritabanlarından ilki *Texas 3D Face Recognition Database*’dir (Gupta vd., 2010b; Gupta vd., 2010a). Bu veritabanında, 118 kişiden oluşan yetişkinlere ait 1149 adet portre ve bunlara karşılık gelen derinlik haritaları bulunmaktadır. Bireylerin yaş aralığı 22 ile 75 arasındadır. İmgeler genellikle nötr yüz ifadesine sahip olmakla birlikte, aralarında farklı yüz ifadelerine sahip olanları da bulunmaktadır. Bireylerin ait görüntülerin alınması ve işlenmesi Texas Üniversitesi, Austin öncülüğünde gerçekleştirilmiştir.

Texas Veritabanı’nın oluşturulması sırasında kullanılan kamera sistemi, 3Q Technologies Ltd. şirketi tarafından geliştirilen MU-2 stereo görüntüleme sistemidir. Görüntülerin alınması esnasında, hem derinlik bilgisi hem de yüze ait renkli imge eşzamanlı olarak kaydedildiği için portre imgeleri ve derinlik haritaları örtüşebilmektedir. Her bir modelde, burna ait uç nokta hemen hemen imgenin orta noktasında yer almaktadır. İşlendikten sonra oluşan imgelerin boyutları 751×501 , ve x, y, z eksenleri boyunca çözünürlükleri 0.32 mm’dir. Her bir z derinlik değeri, 8 bit formatta ve burna ait uç nokta yani en yüksek değer 255 değeri ile, imge arka planı da 0 değeri ile temsil edilmiştir. Renkli portre imgeleri, $751 \times 501 \times 3$ piksel boyutunda 8 bit RGB formatında sıkıştırılmadan

sunulmaktadır. Bununla birlikte, veritabanında yer alan bireylere ait albedo haritaları yer almamaktadır. Bu veritabanında derinlik haritalarına ilave olarak, her bir bireyin özellikleri de belirtilmiştir. Bu özellikler arasında cinsiyet, etnik köken, yüz ifadesi türü, ve ayrıca 25 adet el ile belirlenmiş yüze ait ayırt edici noktanın yerleşim yerleri yer almaktadır. Texas Veritabanı'nda yer alan bireylere ait örnek portre imgeleri ve bunlara karşılık gelen derinlik haritaları Şekil 4.1'de gösterilmektedir.

Bu tez çalışmasının farklı adımlarında kullanılan diğer bir 3-boyutlu yüz veritabanı da, *Bosphorus Database for 3D Face Analysis*'tir (Savran vd., 2008). Bu veritabanı içerisinde, 105 bireye ait farklı pozlarda, duygu ve yüz ifadelerinde, ayrıca bazı bölgelerin kapandığı portre imgeleri ve bunlara karşılık gelen derinlik haritaları bulunmaktadır. İmgeleri oluşturan bireylerden 60'ı erkek ve geri kalan 45'i kadındır. Bununlar birlikte, bu bireylerden 27'si profesyonel aktör ve aktrislerden oluşmaktadır. Toplam imge sayısı 4652'dir. Her bir imge modeli, yüze ait 24 adet ayırt edici nokta için el ile işaretlenmiştir. Bu noktalar arasında; burun uç noktası, kaş iç, orta ve dış noktaları, göz iç ve dış köşe noktaları, burun sol ve sağ noktaları, üst ve alt dudak ortası noktaları, kenar dudak noktaları ve çene ortası noktası bulunmaktadır.

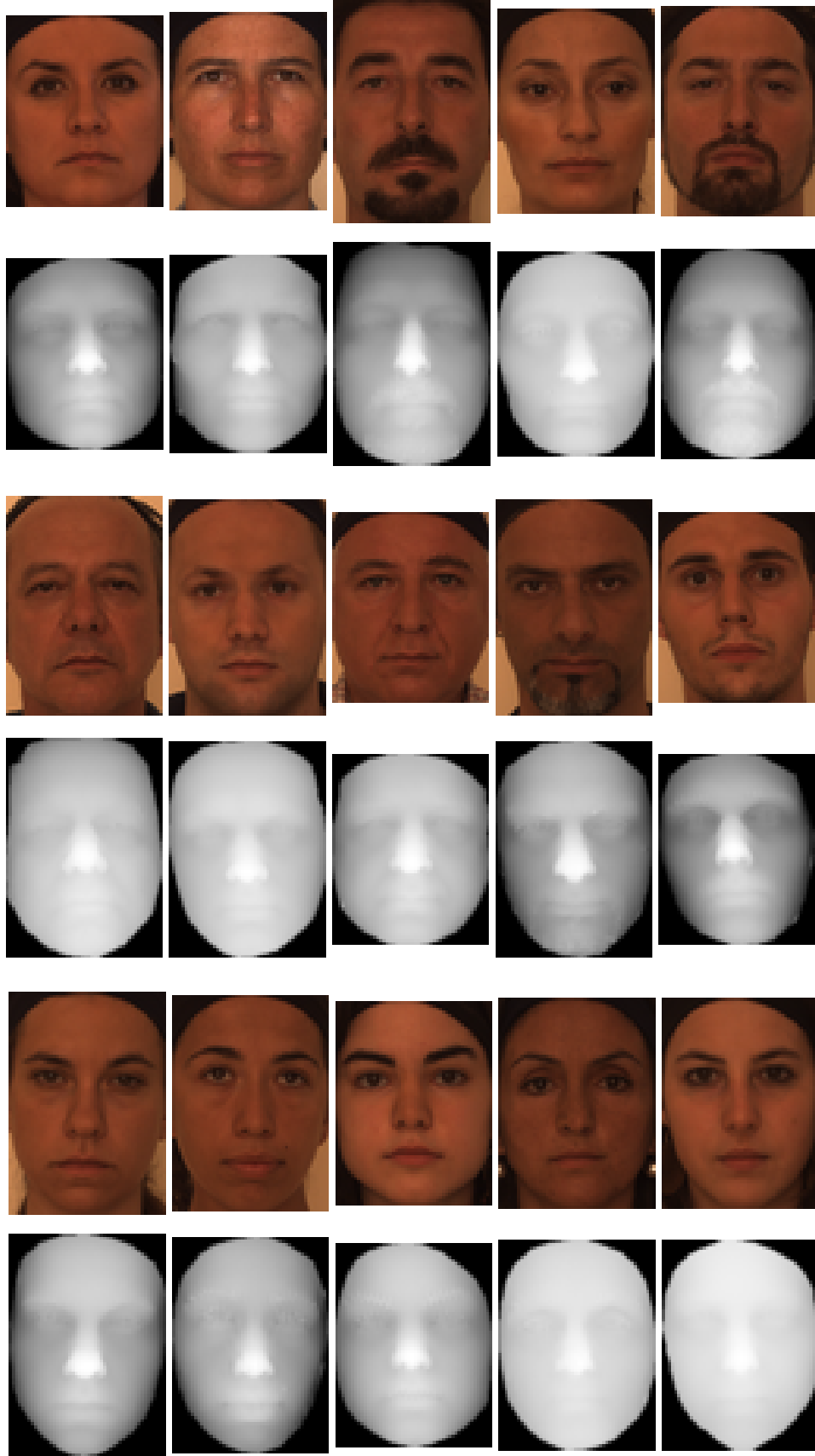
Bosphorus Veritabanı'nda yer alan bireylerin gösterdikleri farklı duygu durumları iki farklı kategoride ele alınmaktadır. Bu kategorilerden bir tanesi, *eylem birimleri* adı verilen ve yüz ifadelerini oluşturmakta yapıtaşları olarak kullanılan gruptur. İkinci grup kategoride ise; mutluluk, şaşkınlık, korku, üzüntü, kızgınlık ve tikslenme gibi duygusal ifadeleri gösteren imgeler yer almaktadır. Baş pozları, yedi farklı yalpa açısı, dört adet pala açısı ve hem yalpa hem de pala açısı içeren iki adet çapraz dönme içermektedir.

Bosphorus Veritabanı oluşturulurken kullanılan görüntüleme sistemi, yapısal ışıklandırma kullanan Inspeck Mega Capturor II 3D'dir. Sensör çözünürlüğü, x, y ve z eksenleri boyunca sırasıyla 0.30 mm., 0.30 mm. ve 0.40 mm'dir. 2-boyutlu renkli imgelerin boyutları 1600×1200 pikseldir. Taranak elde edilen ve bölütlenmiş olan 3-boyutlu yüz modeli 35 bin noktadan oluşmaktadır. Bu veritabanında yer alan bireylere ait örnek portre imgeleri ve derinlik haritaları Şekil 4.2'de gösterilmektedir.

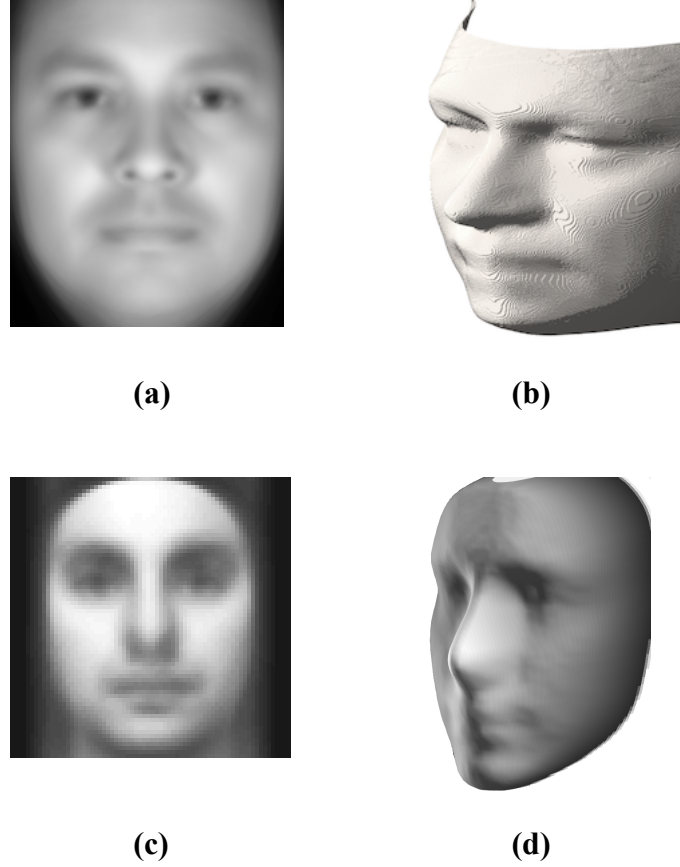
Bölüm 3.6'da anlatılan tek bir imge ve referans bir yüze ait imgeler kullanılarak derinliğin kestirimi yönteminde kullanılmak üzere ihtiyaç duyulan referans bir bireye ait imgeleri oluşturmak üzere her iki veritabanı için de bireylere ait imgelerin ortalamaları alınarak çıktılar oluşturulmuştur. Texas ve Bosphorus veritabanları için oluşturulan referans ortalama imgeler ve bunlara karşılık gelen derinlik haritalarından elde edilen modellerin anlık görüntüleri Şekil 4.3'te gösterilmektedir.



Şekil 4.1 Texas Veritabanı'nda yer alan örnek bireylere ait portre görüntüleri ve derinlik haritaları.



Şekil 4.2 Bosphorus Veritabanı'nda yer alan bireylere ait örnek portre görüntüleri ve derinlik haritaları.



Şekil 4.3 Referans bir bireye ait imgeleri kullanarak derinlik kestirimi yapan ve Bölüm 3.6’da anlatılan algoritmanın farklı adımlarında kullanılmak üzere hazırlanan referans ortalama imgeler ve derinlik haritalarından elde edilmiş modelin anlık görüntüsü. a) Texas Veritabanı’nda yer alan 116 bireye ait portre imgelerinin ortalaması alınarak oluşturulan referans gri-seviye portre imgesi, b) Bu ortalama portreye karşılık gelen ortalama derinlik haritası kullanılarak oluşturulan yüzeyin anlık bir görüntüsü, c) Bosphorus Veritabanı’nda yer alan 105 bireye ait portre imgelerinin ortalaması alınarak oluşturulan referans gri-seviye portre imgesi, d) Bu ortalama portreye karşılık gelen ortalama derinlik haritası kullanılarak oluşturulan yüzeyin anlık bir görüntüsü.

4.2 Derin Öğrenme Tekniklerine Dayanan Derinlik Kestirici Yeni Bir Yöntemin Geliştirilmesi ve Önerilmesi

Bu tez çalışmasında, tek bir imge kullanarak derinlik kestirimi probleminin çözümüne yönelik evrimsel yapay sinir ağlarına ve derin öğrenme tekniklerine dayalı yeni bir yöntem geliştirilmiştir. Literatürdeki eski yöntemlerin, yansıma, albedo ya da ışık kaynaklarının dağılımı gibi yüzey ve imge karakteristiklerini bir şekilde çözümlerine dahil etmek zorunda kaldıkları görülmektedir. Bununla beraber, derin öğrenme yaklaşımını kullanarak geliştirilecek bir yöntem, bu detayların öğrenilmesini önerilen sinir ağı içerisinde yer alan filtreler bırakacaktır. Biçimin geri kazanılması işlemi, sadece uygun bir ağ yapısının, optimizasyon tekniğinin, kayıp fonksiyonun ve veri setinin seçilmesine dönüşecektir. Ayrıca, önceki teknikler bu eksik tanımlı probleme ilave kısıtlar ekleyebilmek için çoğu zaman bir ya da daha fazla düzenleme veya eniyileştirme algoritmasını geliştirmek ya da önceden geliştirilmiş olan yöntemleri kendi akışlarına dahil etmek zorunda kalmaktadırlar. Derin sinir ağları bu adımı da ağ yapısının iç tarafında yer alan ya da kayıp fonksiyonunun bir parçası olan bir ya da daha fazla parametre aracılığı ile gerçekleştirir.

Bu tez çalışmasında, tek girdili 2-boyutlu yüz imgelerinden 3-boyutlu modellerin rekonstrüksiyonunu gerçekleştiren bir çözüm önerilmektedir. Bunun için, Çekişmeli Üreteç Ağ (GAN) yapısı kullanılmaktadır. Ayrıca, farklı ağ türleri incelenmekte ve eğitim teknikleri üzerinde de durulmaktadır. Geliştirilen yöntem, diğer bazı derin öğrenme tabanlı yöntemlerle, ayrıca başarılı bulunan geleneksel yöntemlerle de karşılaştırılmaktadır. Zhang vd. (2017) tarafından önerilen derin öğrenme tabanlı ve otokodlayıcılara dayanan bir çalışma ilk sınıf için seçilmiştir. Kemelmacher-Shlizerman ve Basri'nin (2011) geliştirdiği ve Bölüm 3.6'de tanıtılan tek bir imge ve referans bir bireye ait derinlik haritasını kullanan yöntem ise SFS tabanlı yöntemler arasında en başarılı örneklerden biri olması nedeniyle karşılaştırma yapılmak üzere ikinci sınıf için seçilmiştir.

GAN ağ yapısı, Bölüm 3.9.5'te teorik olarak anlatıldığı üzere, imge üretici olarak davranış gösteren bir *üreteç ağ* ve bu ağın etkin bir şekilde eğitilmesinde önemli bir rol oynayan, ve eğitim esnasında üreteç ağından çıktı olarak alınan imgeleri girdi olarak alan ve bunları sınıflayan bir *ayırtaç ağ* yapısından kurulur. GAN'lar, bu iki farklı ağ arasında bir sıfır toplamlı oyun kurarak, üreteç ağını eğitmek için etkili bir yol sunarlar. Üreteç ağının karşısında yer alan ayırtaç ağın hedefi gerçek imgeler ile üretilen yani sahte imgeleri ayırt etmektir. Bu eğitim esnasında, bu sıfır toplamlı oyun bir *maxmin* çözümüne yani teorik Nash Dengesi'ne doğru evrişir. Artık bu denge noktasına gelen bir sistemde, her iki ağ da daha fazla

öğrenemez ve adapte olamaz. Böyle bir yolla eğitilen üretçi ağı, artık imge üretici olarak kullanılabilir. Eğer GAN kararlı bir şekilde eğitilecek olursa, üretçi ağı çıktısı olan imgeler gerçek imgelere oldukça yaklaşıacaktır.

Bu tez çalışmasında araştırılan problem, tek bir 2-boyutlu imge girdisinden derinlik haritasının kestirimi olduğu ve bu haritaların da 2-boyutlu birer imge olarak değerlendirilebileceği için, bir GAN ağı yapısının imge üretici kısmı bu kestirim işi için önerilmektedir. Bu şekilde kurgulanacak bir yapıda, eğitim esnasında 2-boyutlu portre girdi imgeleri üretçi ağının girdisini, üretçi ağının ürettiği derinlik haritaları da ayırtaç ağının sahte olarak etiketlenmiş girdilerini oluşturmalıdır. Portre imgelerine karşılık gelen ve veritabanlarında yer alan gerçek derinlik haritaları da ayırtaç ağının gerçek imgelerle eğitiminde kullanılmalıdır. Bu eğitimin sonucunda, üretçi ağı ayırtaç ağı olabildiğince yanıtlanacak ve herhangi bir girdi test imgesi verildiğinde buna karşılık gelen derinlik haritasını üretebilecektir.

Orijinal Üretken Çekişmeli Ağ (GAN) (Goodfellow vd., 2014), bir gürültü değişkenini girdi olarak kullanırken, Koşullu Üretken Çekişmeli Ağlar (CGAN), girdi verisini bir koşullayıcı değişken olarak ağı yapısına sokarlar. Bu koşullama daha önceden birçok uygulamada kullanılmıştır, örneğin etiketlerde (Mirza ve Osindero, 2014), metinlerde (Reed vd., 2016), imgelerde ve videolarda (Pathak vd., 2016; Wang ve Gupta, 2016; Zhu vd., 2016; Mathieu vd., 2015; Li ve Wand, 2016), ayrıca herhangi bir uygulamaya bağlı olmayan genel geçer bir yapı içerisinde (Isola vd., 2017). Üretçi ağı girdisi olarak bir veritabanı imgesi kullanılmak istenirse koşullanmış GAN'lar kullanılmalıdır.

Derinlik kestirimi amacıyla geliştirilecek Koşullu GAN yapısı şu şekilde tanımlanabilir: G ve D , sırasıyla üretçi ve ayırtaç ağları gösterebilir. G rasgele bir Gaussyen dağılımlı gürültü z 'yi gözlemlenen x imgesinin koşulu altında d derinlik haritasına haritalar. Yani, $G : \{x, z\} \rightarrow d$ şeklinde gösterilebilir.

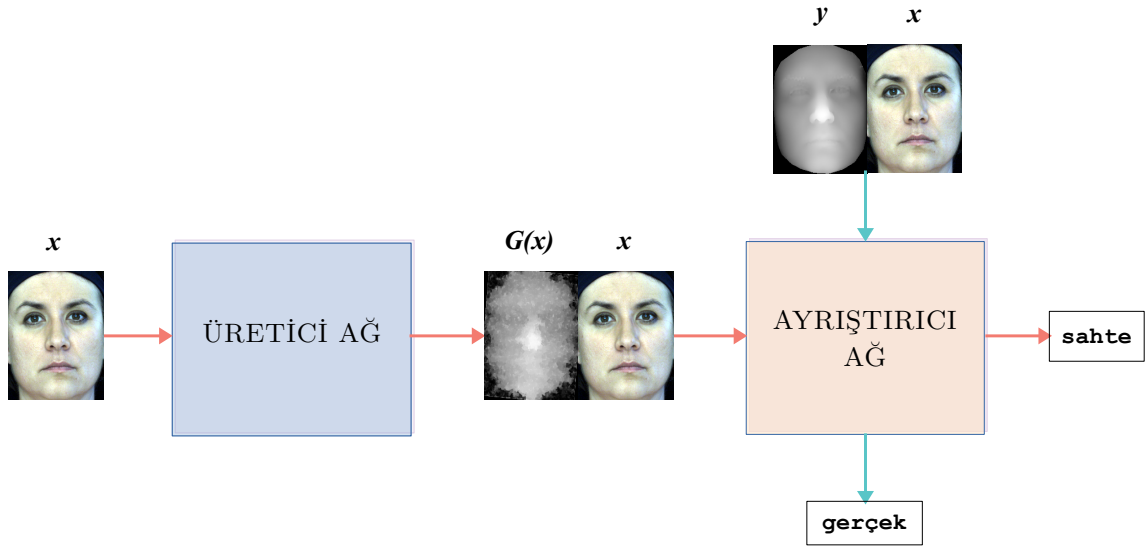
Üretçi ağı eğitilirken amaç fonksiyonunun maksimumu bulunmaya çalışılır:

$$L_G(G, D) = \sum \log D(x, G(x, z)) \quad (4.1)$$

Bu ifadeye G ağı, D ağını üretilmiş derinlik haritalarını gerçek çıktılar olarak kabul etmeye zorlar. Aynı zamanda D ağı da sahte haritaları gerçeklerinden ayırtırmak üzere eğitilir. Bu eğitim sırasında maksimumu aranan amaç fonksiyonu şudur:

$$L_D(G, D) = \sum \log D(x, d) + \log(1 - D(x, G(x, z))) \quad (4.2)$$

Son denklemin ilk bölümü, gerçek imgelerin gerçek derinlik haritalarına eğitimini gösterirken, ikinci bölümü üreteç ağından çıkan ve sahte olarak etiketlenmiş haritaları içermektedir. Eğitim esnasında üreteç ağının gerçek referans değerlerden çok fazla uzaklaşmasını engellemek için 4.1 denkleminde fazladan bir uzaklık kaybı terimi eklenebilir. Bu terim, L2 uzaklık kaybı ya da Isola, vd.nin (2017) önerdiği gibi L1 uzaklık kaybı olabilir.



Şekil 4.4 2-boyutlu girdi imgeleri ve derinlik haritaları ile eğitilecek Koşullu Çekişmeli Üreteç Ağ'ın genel yapısı.

Önerilen yaklaşımın genel yapısı Şekil 4.4’da gösterilmektedir (Arslan ve Seke, 2019). Üreteç ağ için nihai amaç fonksiyonu şu şekilde yazılabilir:

$$G^* = \arg \min_G \max_D L_G(G, D) + \lambda d_{L1|L2}(G). \quad (4.3)$$

Bu denklemde geçen L_G ifadesi, 4.1 denkleminde verilen kayıp fonksiyonudur. Sondaki terim bir $L1$ -norm ya da $L2$ -norm uzaklığı fonksiyonu olarak seçilebilir.

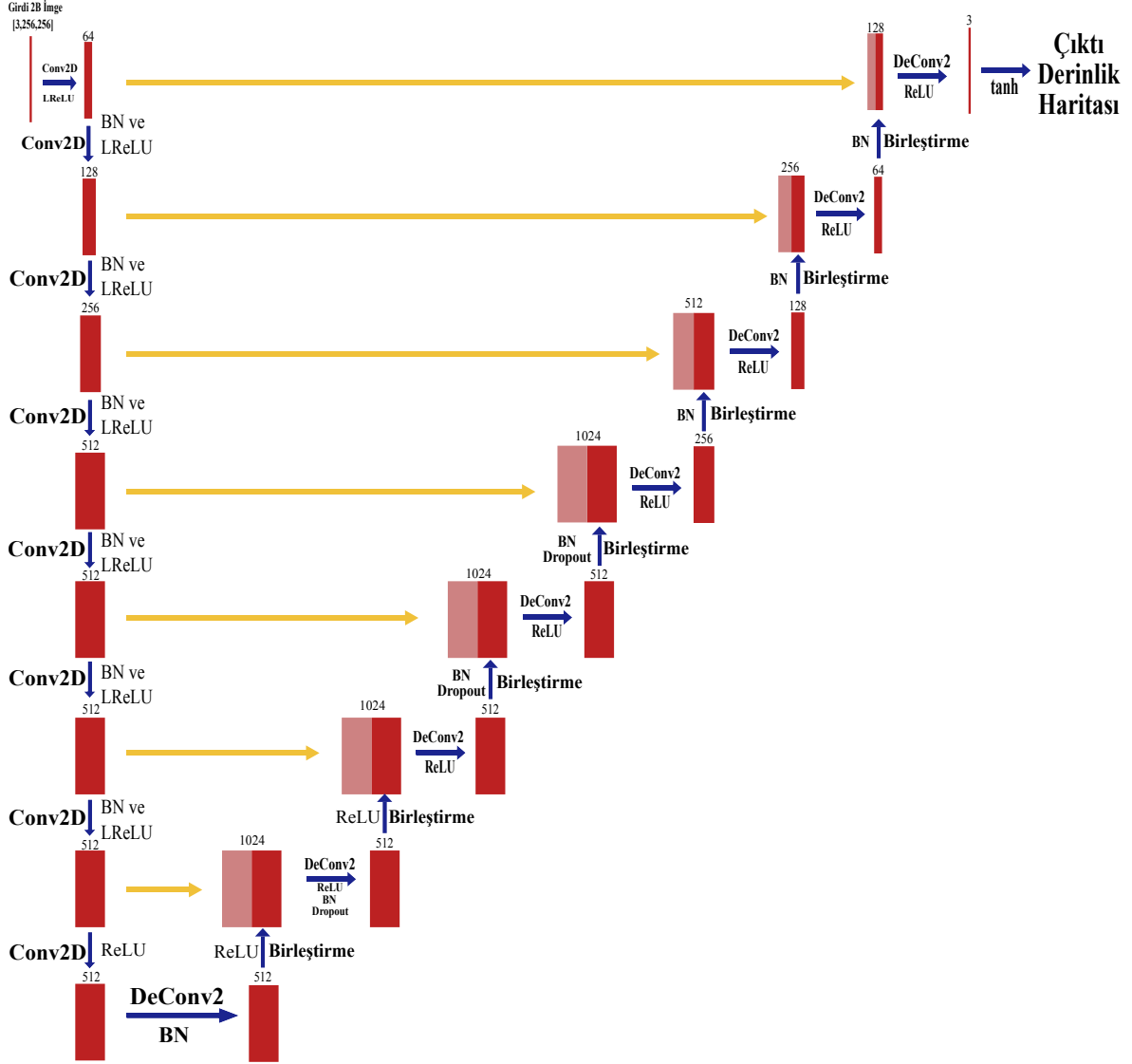
Derinlik kestirimi için oluşturulacak Koşullu Üretken Çekişmeli Ağ’ın kurgulanması sırasında belirlenmesi gereken yapıtaşları şu şekilde listelenebilir:

1. Üreteç ağın yapısı,
2. Ayırtaç ağın yapısı,
3. Kayıp fonksiyonu,
4. Eniyileştirme (optimizasyon) algoritması,
5. Hiperparametreler.

Aşağıda yer alan alt bölümlerde sırasıyla bu listede yer alan unsurların üstünde durulmaktadır.

4.2.1 Üreteç ağ yapıları

Üreteç-ayırtaç ikili yapısından oluşan bütüncül modelin en önemli parçasını oluşturan ve eğitim sonrası kullanılmak üzere derinlik kestirici görevini yerine getirecek olan üreteç ağıdır. Bu ağı oluşturmak için çeşitli alternatifler söz konusudur. Bunlardan bir tanesi, daha önce biyomedikal imgelerin bölütlenmesi işlevi için Freiburg Üniversitesi, Almanya’da geliştirilmiş olan ve bir kodlayıcı-kodçözücüden oluşan U-Net ağıdır (Ronneberger vd., 2015). Bu model, girdi imgeleri öncelikle yüksek çözünürlüklere adım adım dönüştürür. Yapının ikinci kısmında ise, tipik evrişimsel ağlarda olduğu gibi birbirini takip eden evrişim katmanları, düzeltilmiş doğrusal birim (ReLU) ve maksimum havuzlama (*max pooling*) operatörleri yer almaktadır. Bu büzülme aşamasında, uzamsal bilgi azalırken öz nitelik bilgisi artmaktadır. Genişletici ve büzücü kısımlarda karşılık gelen çözünürlük bloklarını birbirlerine bağlamak için bazı atlama katmanları (*skip layers*) eklenmesi, elde edilen bilginin karşı tarafa taşınmasına olanak tanımaktadır. Derinlik kestirici olarak kullanılan, farklılaştırılmış bir U-Net modeli Şekil 4.5’te gösterilmektedir.



Şekil 4.5 Derinlik kestirici üretici ağ olarak eğitilecek olan farklılaştırılmış bir U-Net yapısı. *Conv2D* iki-boyutlu evrişimsel işlev katmanlarını, *BN* *batch normalization* işlemini, *DeConv2* iki-boyutlu ters evrişimsel işlev katmanlarını, *LReLU* *Leaky ReLU* aktifleştirme işlevini göstermektedir. Girdi imgesi tensör boyutu (3,256,256)'dır.

Üreteç ağ yapısı için kullanılabilir alternatif modellerden bir tanesi de *Yığılı Kumsaati* adı verilen (*Stacked Hourglass*) (SHG) yapıdır (Newell vd., 2016). Bu yapı, vücut pozü kestirimi amacıyla geliştirilmiş olmakla birlikte, yukarı örnekleme ile aşağıdan-yukarıya ve havuzlama ile yukarıdan-aşağıya işleme birimlerinden oluşan tek bir modül yan yana eklenerek imge çıktısı oluşturulabilmektedir. Bu yapı, otokodlayıcı yapılara ya da *conv-deconv* yapılara benzemekle birlikte, bunlardan iki noktada ayrılmaktadır. Bu farklardan ilki, SHG yapısında geri-havuzlama ve geri-evrimsel katmanlar kullanılmaz. Diğer bir fark da, SHG’de modüller yan yana birleştirilmek suretiyle daha büyük bir ağ kurulur.

4.2.2 Ayırtaç ağ yapıları

Kendisine girdi olarak verilen imgelerin gerçek imgeler mi yoksa üreteç ağı tarafından üretilmiş ve dolayısıyla sahte olarak etiketlenmesi gerektiğine karar veren, böylece de üreteç ağını daha doğru sonuçlar üretmeye zorlayan ayırtaç ağ için kullanılabilir yapılardan biri DCGAN’dır (Radford vd., 2015). Hem üreteç hem de ayırtaç olarak kullanılabilen DCGAN ağ yapısı, son evrim katmanı düzleştirilip bir sigmoid çıktısına bağlanırsa sınıflandırıcı bir hal almaktadır. Bu tez çalışmasında önerilen yöntemde kullanılan DCGAN ağ yapısında yer alan katmanlar sırasıyla şu şekildedir:

- 1 Conv2D (64 kanal)
- 2 BatchNorm
- 3 LeakyReLU
- 4 Conv2D (256 kanal)
- 5 BatchNorm
- 6 LeakyReLU
- 7 Conv2D (512 kanal)
- 8 BatchNorm
- 9 LeakyReLU
- 10 Conv2D (1024 kanal)
- 11 BatchNorm
- 12 LeakyReLU
- 13 Conv2D (2048 kanal)
- 14 BatchNorm
- 15 LeakyReLU

Ayrıtaç ağ olarak test edilebilecek bir diğer yapı da PatchGAN'dir (Isola vd., 2017). L1 ve L2 kayıp fonksiyonlarının, bulanık imgeler ürettiği belirtilmiştir. Her ne kadar yüksek frekanslarda bir keskinlik sunmasalar da, düşük frekanslı verileri yakalayabilmektedirler. Böylece, GAN ayrıştırıcısı sadece yüksek frekansları modellemek üzere ve düşük frekanslar için de bir L1 terimi ekleyerek kurgulanabilir. Bunu gerçekleştirmek üzere, yerel imge bölgelerinde yer alan yapıları kontrol eden ve $N \times N$ bir piksel alanında yer alan bölgelerin gerçek mi yoksa sahte mi olduğunu sınıflayan bir ayrıştırıcı yapı ortaya çıkmaktadır. Bu ayrıştırıcı, tüm imge üzerinde evrimsel olarak gezdirilir, bütün sonuçların ortalaması alınarak D ağının nihai çıktısı üretilir.

4.2.3 Kayıp fonksiyonları

GAN'ların eğitim süreçlerinin, birden fazla nedenden dolayı sorunlu olduğu vurgulanmıştır (Metz vd., 2016; Heusel vd., 2017; Berthelot vd., 2017). Bu nedenler arasında, modelin salınım yapması ya da hiç yakınsamaması gibi yakınsama problemleri, ayrıtaç ağın sıfır toplamlı oyunda üreteç ağına baskın gelmesi sonucu gradyanların *yok olması* ya da *patlaması*, üreteç ağının öğrenememesi ve az sayıda çıktı üretmesi sonucunda yaşanan modların çökmesi problemi, ve aşırı uyumlanma sorunları gösterilmiştir. Modellerin genel olarak hiperparametrelere son derece duyarlı oldukları görülmektedir. Daha kararlı ve sağlam ağların eğitilebilmesi amacıyla bu problemin çözümüne yönelik çok çaba sarf edilmektedir (Salimans vd., 2016; Arjovsky ve Bottou, 2017; Arjovsky vd., 2017; Gulrajani vd., 2017; Mao vd., 2017). Gradyan İnişi ve bununla ilintili diğer algoritmalar bir minimum noktasına yakınsamaya çalıştıkları için, minimum noktadan ziyade bir Nash dengesinin aranan çözüm olduğu GAN'ların eğitimi esnasında kolaylıkla çökebilmektedirler.

Wasserstein GAN yapısının, GAN'ların eğitilmesi sırasında baş gösteren bu zorluklardan bir kısmını gidermede başarılı olduğu belirtilmiştir (Arjovsky vd., 2017). *Toprak-Taşıyıcı* (EM) ya da Wasserstein-1 uzaklığı, bir bölge üzerinde iki olasılık dağılımı arasındaki uzaklıktır. μ ve ν ile gösterilecek iki dağılım arasında, bu uzaklık şu şekilde ifade edilebilir:

$$W_p(\mu, \nu) = \inf \mathbf{E}[d(X, Y)] \quad (4.4)$$

Bu denklemde \mathbf{E} , X ve Y rasgele değişkenlerinin bütün bileşik olasılık dağılımları üzerinde alınan beklenen değerdir. $d(\cdot)$ ifadesi de, mutlak değer uzaklık fonksiyonudur. Arjovsky, vd. (2017) 4.4 denkleminde yer alan en büyük alt sınırı (*infimum*) belirtebilmek

için *K-Lipschitz süreklilik kısıtını* önermişler ve bu şartı yaklaşık olarak gerçekleyebilmek için ağırlık kırpma işlemini uygulamışlardır.

4.2.4 Eniyileştirme algoritmaları

Bir derin öğrenme temelli ya da geleneksel yapay sinir ağının eğitilmesinde, yani her bir öğrenim adımında ağırlık katsayılarının girdi ve çıktılara göre güncellenmesinde, böylece eğitimi yapılan ağın veri setine uygun şekilde yakınsamasında etkili olan en önemli aşama, bu güncellemeleri gerçekleştiren eniyileştirme fonksiyonudur. Stokastik Gradyen İnişi (*Stochastic gradient descent*) (SGD), bu algoritmalar arasında en eskilerden bir tanesidir ve geri yayılım (*backpropagation*) algoritması ile birleştirildiğine yapay sinir ağlarının eğitiminde kullanılan standart yöntemdir. SGD, rasgele seçilen örnekleri gradyen değerlerinin hesaplanmasında kullanan iteratif bir yaklaşımdır.

AdaGrad (Duchi vd., 2011), her bir parametre için farklı bir öğrenme oranı kullanan adaptif bir algoritmadır. Bu yaklaşımda, seyrek parametreler için öğrenme oranları artırılırken, daha az seyrek olanlar için öğrenme oranları azaltılır. RMSProp (Tieleman ve Hinton, 2012), her bir parametre için öğrenme oranının adapte edildiği bir diğer algoritmadır. Bu yaklaşımda, öğrenme oranı, bir ağırlık katsayısı için, o ağırlık için daha önce hesaplanan gradyenlerin ortalamasına bölünmektedir. Adam algoritması, RMSProp'un güncellenmiş bir halidir (Kingma ve Ba, 2014). Bu optimizasyonda, hem gradyenlerin hem de gradyenlere ilişkin ikinci momentlerin ortalamaları kullanılmaktadır.

4.2.5 Parametrelerin seçimi

Geliştirilen bütün derin öğrenme modellerinin eğitimi öncesinde ve sırasında belirlenmesi gereken parametreler söz konusudur. Bu parametreler, eğitimin ve eğitim çıktısının kararlı ve etkin olmasında son derece etkili bir rol oynarlar. Eğitim süreci başlamadan önce belirlenen ve eğitim esnasında da sabit kalan parametrelere *hiperparametre* adı verilmektedir. Bunlara örnek olarak, öğrenme oranı, epok sayısı, *batch* sayısı, kullanılan ağ yapılarında yer alan evrişimsel katmanlardaki *kernel* ve *stride* değerleri gösterilebilir.

Öğrenme oranı, eğitim sürecini en çok etkileyen parametredir. Her bir iterasyonda, eniyileştirme fonksiyonu bulunan gradyenleri bu oranla çarparak işleme alınır. Öğrenme oranını eğitim süreci boyunca sabit tutmak yerine iterasyonlar ilerledikçe belli bir fonksiyona bağlı olarak azaltmanın eğitimi daha kararlı duruma getirdiği ve yakınsamayı kolaylaştırdığı görülmüştür. Öğrenme oranını her bir iterasyonda güncellemeye, öğrenme oranının çizelgelenmesi (*learning rate scheduling*) ve yeni oluşacak öğrenme oranını

belirleyen fonksiyonlara da çizelgeleme fonksiyonları (*scheduling functions*) adı verilmektedir. Bir başlangıç değeri lr_{ilk} ve bir son değer lr_{son} arasındaki çizelgelemeyi gösteren fonksiyonlara örnekler Şekil 4.6'da görülebilir. Bu fonksiyonlar arasında, belli bir adım sayısından sonra belli bir faktör oranınca azaltan, adım sayılarının farklılık gösterebildiği, üstel bir fonksiyona ve katsayıya bağlı olarak azaltma işlemi yapan, kosinüs fonksiyonuna bağlı olarak güncelleme yapan fonksiyonlar yer almaktadır. Çoğu uygulamada, bu fonksiyonlar $lr_{son} = 0$ alınarak kullanılmaktadır. Çizelgeleme fonksiyonları şu şekilde listelenebilir:

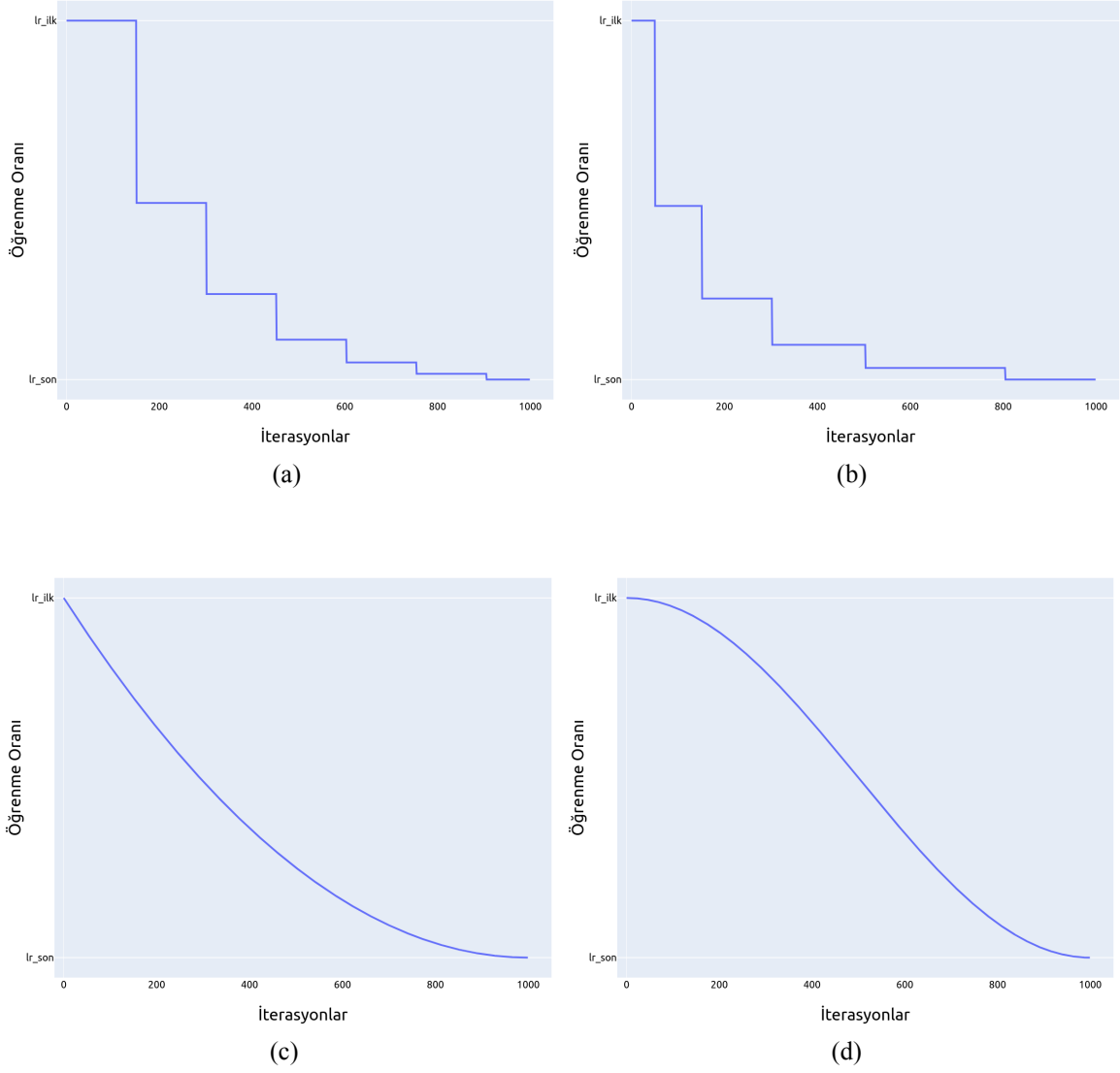
1. Tek faktör çizelgeleme fonksiyonu
2. Çok faktörlü çizelgeleme fonksiyonu
3. Üstel çizelgeleme fonksiyonu
4. Kosinüs çizelgeleme fonksiyonu

Epok sayısı, eğitim süresinin uzunluğunu etkiler ve veri setinde yer alan bütün örneklerin işlenmesini kapsayan kaç adet *çağ* olacağını belirtir. Her bir epoktaki güncellemede kullanılan örnek sayısı *batch* sayısıdır. Böylece bu sayıyla güncelleme adımları sayısı ters orantılı oluşur. Örneğin, 10.000 adet imgeden oluşan bir veri setinde, *batch* sayısı 50 ise, her bir epokta gerçekleşecek güncelleme sayısı 200 olacaktır.

4.2.6 Deneysel kurgu

Bu bölümde, eğitim ve test işlemleri için kurgulanan yapı sunulmaktadır. Bu süreçte kullanılan veri setleri hakkında genel bilgiler Bölüm 4.1'de verilmiştir. Herhangi bir derinlik kestirici algoritmanın geliştirilmesinde lazer tarayıcılar ya da diğer derinlik ölçücü aygıtlar ile elde edilmiş gerçek derinlik haritalarına ihtiyaç duyulmaktadır. Kullanılan veritabanlarında yer alan imgeler, üç gruba ayrılır: eğitim alt kümesi, geçiş alt kümesi ve test alt kümesi. Test alt kümesinde yer alan imgeler, eğitim aşaması tamamlandıktan sonra gerçekleştirilecek olan test adımında kullanılacak olan imgeler ve bunlara karşılık gelen derinlik haritalarıdır.

Bu tez çalışması kapsamında, iki ayrı 3-boyutlu yüz veritabanı kullanılmıştır. Bunlar, *The Texas 3D Face Recognition Database* ve *Bosphorus Database for 3D Face Analysis* veritabanlarıdır. Her iki veritabanında yer alan bireylerin hem portre görüntüleri hem de bu görüntülere birebir karşılık gelen derinlik haritaları verilmektedir. Texas Veritabanı'nda 116 bireye ait imgeler yer alırken, Bosphorus Veritabanı'nda 105 bireyin farklı pozlarda ve duygu ifadelerinde alınmış imgeler bulunmaktadır. Bosphorus



Şekil 4.6 Öğrenme oranını her iterasyonda sabit tutmak yerine belli bir algoritmaya dayalı olarak azaltmak, eğitimin daha kararlı olmasını sağlayabilir. Öğrenme oranının çizelgelenmesinde kullanılan fonksiyonlar. (a) Tek faktörlü azaltma fonksiyonu, belli iterasyon sayısından sonra öğrenme oranı yarıya düşürülmekte, (b) adımların uzunluğu değişken tutulabilir, (c) polinom denkleme göre, ve (d) kosinüs fonksiyonu.

Veritabanı'ndan seçilen nötr ve nötre yakın pozlardaki imgeler Texas Veritabanı'ndan gelen imgelerle birleştirilmiştir. Veritabanlarında yer alan her bir 2-boyutlu portre imgesi için, bu imgedeki her bir piksele denk gelen derinlik bilgisini gösteren bir derinlik haritası bulunmaktadır. Derinlik haritaları, $[0, 255]$ aralığına normalize edilmiştir. Buradaki 255 değeri kameraya en yakın düzlemi, 0 değeri ise imge arka planını temsil eder. Algoritmaların çalıştırılması ya da eğitimler öncesi, bütün 2-boyutlu girdi imgeleri ve çıktı derinlik haritaları 256×256 çözünürlüğüne ölçeklenmiştir. Texas Veritabanı'nda yer alan yüz portre imgeleri ve bunlara karşılık gelen derinlik haritaları, orijinal imgelerde geniş arka plan bölgeleri bulunduğu için görüntü karesinin çoğunu doldurabilmek amacıyla yakınlaştırılmışlardır. Her veritabanından 10 birey test amacıyla rasgele seçilmiş ve eğitim veri seti geriye kalan portre-derinlik çiftleriyle oluşturulmuştur.

Oluşturulan veri setindeki varyansı artırmak ve eğitilen ağların sağlamlığını sağlamak amacıyla eğitim veri setine çoğaltma (*augmentation*) işlemi uygulanmıştır. Orijinal veri setinden rasgele seçilen ve tüm veri setinin üçte birini oluşturan imgelere şu dönüşümler yapılmıştır: Gaussyen bulanıklaştırma, saat yönünden az döndürme, ters saat yönünde az döndürme, beyaz dengeleme. Son dönüşüm, orijinal veritabanı imgelerinin düşük kontrast ve sönük aydınlatma şartları altında alınması nedeniyle seçilmiştir. Böylece, veri setinde yer alan imgelerin sayısı 1901'den 4442'ye yükselmiştir. Bir bireye ait çoğaltılmış imgeler kümesi Şekil 4.7'de gösterilmiştir. Veritabanları ile ilgili bilgiler Çizelge 4.1'de özetlenmiştir.



Şekil 4.7 Veri setinde yer alan bir bireye ait çoğaltılmış imgeler. Soldan sağa: a) Beyaz dengelenmiş, b) Saat yönünde döndürülmüş, c) Ters saat yönünde döndürülmüş, d) Gaussyen bulanıklaştırılmış, e) Orijinal imge.

Üreteç ağ yapısı, ayırtaç ağ yapısı ve kayıp fonksiyonu birden fazla seçenek arasından seçmek suretiyle, çok sayıda farklı GAN yapısı ortaya konulabilir. Bu farklı yapılardan testleri yapılan üç tanesi Çizelge 4.2'de gösterilmiştir. I nolu konfigürasyon, Isola, vd. (2017)'nin önerdiği ve *pix2pix* adı verilen yapıdır. Üreteç ağı olarak değiştirilmiş

Çizelge 4.1 Geliştirilen ve önerilen yöntem ile eğitilecek olan ağ yapısı için kullanılan Texas ve Bosphorus veritabanlarında yer alan bireylerin sayısını, her bir veritabanı için eğitim amacıyla ayrılan imgelerin sayısını ve bu imgeler üzerinde gerçekleştirilen çoğaltma işlemleri sonucunda oluşan yeni imge sayılarını gösteren çizelge.

Veritabanı	Birey sayısı	Eğitim için ayrılan imgelerin sayısı	Çoğaltılmış imgeler
Texas	116	843	1979
Bosphorus	105	1058	2463

bir U-Net (Ronneberger vd., 2015) versiyonunu kullanmaktadır. II nolu konfigürasyonda koşullandırılmış Wasserstein GAN kayıp fonksiyonu ile ayırtaç ağ yapısı olarak DCGAN (Radford vd., 2015) kullanılmıştır. III nolu konfigürasyonda, üreteç ağ yapısı için daha önceden vücut pozü kestirimi için geliştirilmiş olan *Yıgılı Kum Saati* adında farklı bir ağ yapısı kullanılmıştır (Newell vd., 2016). Bu tez çalışmasında, girdi imge boyutunu koruyabilmek için bu yapının ilk katmanı kaldırılarak bir miktar değiştirilmiştir. Bunun haricinde, çalışmada önerildiği üzere 8-kumsaati 1-kalıntılı (*residual*) bir ağ yapısı uygulanmıştır.

Çizelge 4.2 Üreteç ağ yapısı, ayırtaç ağ yapısı ve kayıp fonksiyonu için farklı seçenekler yer almaktadır. Bunlar arasından yapılan seçimlerle kurulan GAN konfigürasyonlarını gösteren çizelge.

Konfigürasyon	Üreteç	Ayırtaç	Kayıp Fonksiyonu
I	U-Net	PatchGAN	CGAN + L1 (L2)
II	U-Net	DCGAN	WGAN + L1 (L2)
III	SHG	DCGAN	WGAN + L1 (L2)

Bütün kaynak kodu geliştirme işlemleri, Apache MxNet, Tensorflow ve PyTorch derin öğrenme kütüphaneleri kullanılarak Python programlama dilinde yapılmıştır. Ubuntu işletim sistemi çalıştıran ve üzerinde tek bir GPU kartı (*NVIDIA Tesla V100*) bulunan bir bilgisayar ortamında eğitimler ve testler gerçekleştirilmiştir. Eğitim süreçlerinin eniyileştirilmesi ve her bir konfigürasyon için en iyi sonucun elde edilebilmesi için, hiperparametreler için farklı değerlerde denemeler gerçekleştirilmiştir. Örneğin; epok sayıları için 250, 500, 1000 ve 2500 değerleri, *batch* sayıları için 10, 20, 32, 64; öğrenme oranı için

ise 0.0001 ya da 0.0002 deęerleri ile eęitimler yapılmıřtır. Derin öęrenme temmelli yaklařıma iliřkin modellerin kurulmasını gsteren ve eęitim srelerini gerekleřtiren kod rneklere *Ek Aıklamalar-C* blmnde sunulmaktadır.

Rekonstrksiyon ıktılarının bařarım oranlarını deęerlendirebilmek ve dięer yntemlerle de karřılařtırmalarını yapabilmek iin, test veri setinde yer alan her bir imge iin *Yzde Ortalama Mutlak Hata* hesaplanmıřtır. Bunun ardından, btn test imgeleri iin ortalamalar, Kemelmacher-Shlizerman ve Basri'nin (2011) uyguladıęı hesaplamaya benzer řekilde elde edilmiřtir. $h_{out}(x, y)$ ve $h_{gt}(x, y)$ sırasıyla bir girdi imgesinden kestirilen piksel derinlik deęeri ve bu pikselin gerek referans derinlik deęerleri olsun. Bylece, Yzde Ortalama Mutlak Hata řu řekilde hesaplanabilir:

$$e(x, y) = \left| \frac{h_{out}(x, y) - h_{gt}(x, y)}{255.0} \right| * 100. \quad (4.5)$$

Bir imge iin toplam ortalama hata oranı ise řyle gsterilebilir:

$$E_i = \left(\sum_{x, y \in \Omega} e(x, y) \right) \times \frac{1}{m \times n}. \quad (4.6)$$

Burada imge boyutu $m \times n$ 'dir.

5. BULGULAR VE TARTIŞMA

Bu tez çalışmasında yapılan deneyler, testler, karşılaştırmalar ve geliştirilen yöntemlere ilişkin çıktılar bu ana bölüm altında sunulmaktadır. Bölüm 3.8’de tartışılan aydınlatma şartlarının kestirimi yöntemlerine ilişkin çıktılar ve değerlendirmeler Bölüm 5.1’de verilmektedir. Bir yüz imgesi ve referans yüze ait derinlik haritası kullanarak derinlik bilgisinin bulunması tekniğiyle ilgili bulgular Bölüm 5.2’de tartışılmaktadır. Bu tez çalışması kapsamında geliştirilen ve önerilen yöntemle ilişkin bulgular ve değerlendirmeler Bölüm 5.3’te sunulmaktadır.

5.1 Aydınlatma Şartlarının Kestirimi Yöntemlerine İlişkin Bulgular

Bölüm 3.8’de sunulan aydınlatma şartlarının kestirimi yöntemlerinden öne çıkan bazı algoritmalar seçilerek, bunlar MATLAB ortamında uygulanmış, girdi test nesnelere ile çıktılar alınmış ve bu tekniklerin karşılaştırılmaları yapılmıştır. Ayrıca, Bölüm 5.1.2’de bu çalışmalar neticesinde elde edilen sonuçlar yorumlanmakta ve değerlendirilmektedir.

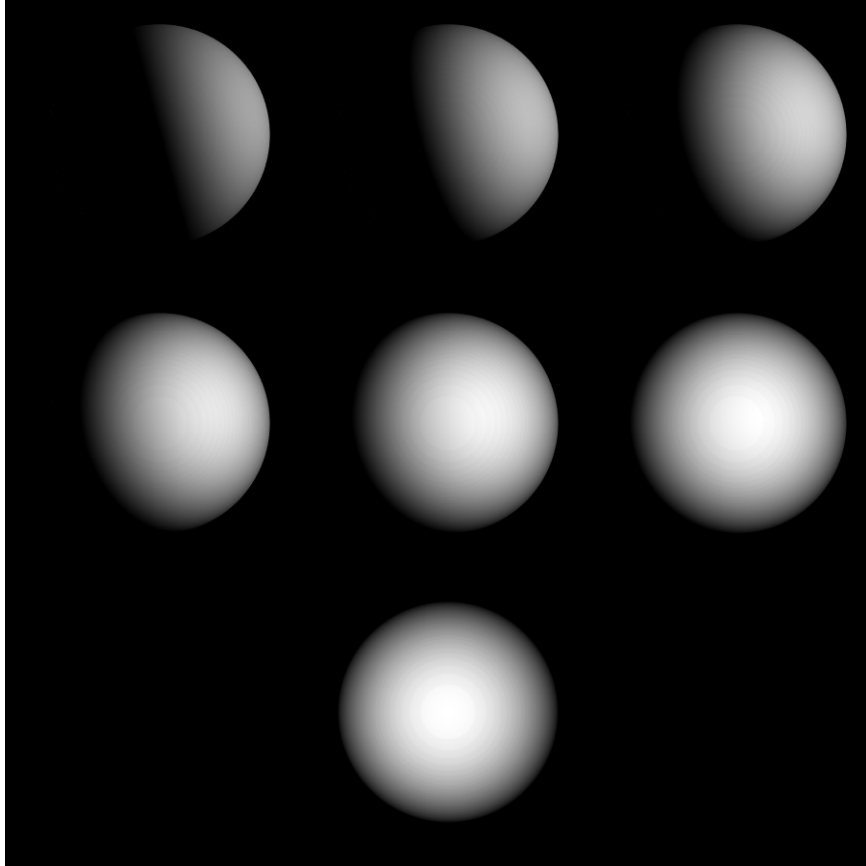
5.1.1 Uygulama ve deneyler

Sentetik olarak oluşturulmuş bir küreden ve Mozart yüzey verisi kullanılarak oluşturulmuş 3-boyutlu bir modelden kurulan imgeler, böylece iki farklı sınıfta yer alacak şekilde değişen eğiklik açılarında (*tilt*) yer alan uzak noktasal bir ışık kaynağı ile aydınlatılmıştır. Bütün ambiyans ve ayna benzeri yansıma ışıkları göz ardı edilmiştir, yani yansıyan ışığın sadece yayınık bileşeni araştırılmıştır. Aydınlatma yönü açıları, eğiklik açısı için 15° ’lik artışlarla 0 ve 360 derece arasında, ve verev açısı için 0 ve 90 derece arasındadır. Şekil 5.1 ve Şekil 5.2, bu şekilde oluşturulan test veritabanından iki örnek imge setini göstermektedir.

Karşılaştırma yapmak üzere seçilen teknikler şunlardır: Pentland’ın yöntemi, Lee ve Rosenfeld’in yöntemi, Zheng ve Chellappa’nın yerel yöntemi. Pentland yönteminde β matrisi oluşturulurken sekiz adet yön kullanılmıştır. Ayrıca, Lee ve Rosenfeld yöntemi, Zheng ve Chellappa (1991) tarafından ifade edildiği şekliyle uygulanmıştır.

Tekniklere ilişkin algoritmalar geliştirildikten sonra, her bir nesne için dört farklı deney seti hazırlanmış ve bu teknikler bu setler üzerinde test edilmiştir. Gerçekleştirilen bu deneyler şunlardır:

1. eğiklik açıları gerçek eğiklik açısı değerlerine karşı, 0 dereceden 360 dereceye 15 derecelik artışlarla (verev açısı 45° 'de iken),
2. verev açıları gerçek eğiklik açısı değerlerine karşı, kestirilmesi gereken verev açısı 45° iken,
3. verev açıları gerçek verev açısı değerlerine karşı, 0 dereceden 90 dereceye 15 derecelik artışlarla (eğiklik açısı 45° 'de iken),
4. eğiklik açıları gerçek verev açısı değerlerine karşı, kestirilmesi gereken eğiklik açısı 45° iken.



Şekil 5.1 Üç boyutlu sentetik bir küre modelinin farklı açılardan aydınlatılması sonucunda elde edilen iki boyutlu imgeler: Verrev açısı 0° 'den 90° 'ye kadar değişmekte, eğiklik açısı sabit ve 15° 'dir.



Şekil 5.2 Üç boyutlu Mozart modelinin farklı açılardan aydınlatılması sonucunda elde edilen iki boyutlu imgeler: Eğiklik açıları 0, 90, 180, 270, verev açıları 30 ve 75 derecede.

5.1.2 Değerlendirme

Şekil 5.3'te ve Şekil 5.4'te, sırasıyla sentetik küre ve Mozart modellerinden elde edilen imgeler için oluşturulmuş olan test setlerine ait sonuçlar gösterilmektedirler. Her bir grupta, gerçek eğiklik ve gerçek verev açılarına karşı kestirilen eğiklik ve verev açıları olmak üzere 4 farklı grafik bulunmaktadır. Bu grafikler incelendiğinde şu sonuçlar çıkartılabilir:

- Karşılaştırılması yapılan üç yöntemin de sentetik küre imgeleri için eğiklik açısı kestirimini neredeyse mükemmel derecede yaptıkları gözlemlenmektedir. Bunun nedeni, sentetik kürenin, bu algoritmaların bir ön kabulü olan yüzey normallerinin isotropik olarak dağılım gösterdiği varsayımına uyan en uygun nesne olmasıdır.
- Zheng ve Chellappa'nın yerel yöntemini, gerçek eğiklik açıları karşısında verev açısı kestiriminde diğer iki yönteme göre çok daha iyi sonuç verdiği görülmektedir. Pentland'ın verev açısı kestiricisinin, 0° – 90° aralığında bulunan bütün eğiklik açılarına karşı kötü performans gösterdiği de gözlemlenmektedir.

- Zheng ve Chellappa'nın yerel yönteminin verev açısı kestiricisinin, gerçek verev açıları karşısında gözle görülür bir şekilde daha iyi sonuç verdiği görülmektedir. Lee ve Rosenfeld yönteminin, 90 dereceye yakın açılarda ani bir bozulma gösterdiği görülmekte ve bunun aynı zamanda Zheng ve Chellappa'nın yerel yönteminin verev açısı kestiricisinin diğer yöntemlere karşı sağlamlığını ispat ettiği söylenebilir.
- Zheng ve Chellappa'nın yerel yönteminin eğiklik açısı kestiricisinin Şekil 5.3d'de görüldüğü gibi değişen verev açıları karşısında daha sağlam olduğu anlaşılmaktadır. 80 derece ve üzerinde bozulmalar başlarken, Pentland ve Lee-Rosenfeld kestiricilerinin de gerçek verev açısı aralığı boyunca aynı sonuçları ürettikleri gözlenmektedir.
- Bütün sonuçlar ele alındığında, sentetik küre modelinden elde edilen imgelerde bile verev açısı kestiricilerin eğiklik açısı kestiricileri ile karşılaştırıldıklarında çok daha kötü performans gösterdikleri açıktır. Bu üç yöntem arasında en iyi sonuç verenin Zheng ve Chellappa'nın yerel yöntemi olduğu ve hata oranının bazı noktalarda 10 dereceye yaklaştığı söylenebilir.
- Sentetik küreden Mozart modeli imgelerine geçildiğinde, eğiklik açısı kestiricileri de dahil olmak üzere bütün kestiricilerin performanslarında bozulmalar gözlemlenmektedir.
- Mozart modeli imgelerinde, Zheng ve Chellappa'nın yerel yönteminin verev açısı kestiricisi dramatik şekilde bozulmakla birlikte Lee-Rosenfeld kestiricisinin çok daha iyi sonuç verdiği görülmektedir.
- Pentland'ın verev açısı kestiricisinin Mozart modeli imgelerinde de en kötü sonuç veren yöntem olduğu ve Mozart modeli imgelerinde hata oranının oldukça arttığı görülmektedir.
- Pentland'ın yöntemi haricindeki iki yöntemin eğiklik açısı kestiricilerinin, verev açısının 60 derecenin üstüne çıktığı durumlarda hızla kötü sonuçlar vermeye başladıkları gözlemlenmektedir.

Elde edilen sonuçlara genel olarak bakıldığında, bu sonuçların Zheng ve Chellappa (1991), Chojnacki vd., (1994) ve Gibbins vd., (1991) yayınlarında belirtilen bulgularla örtüştüğü görülmektedir. İnsan yüzleri gibi daha kompleks nesnelere oluşan sahnelerde aydınlatma yönlerinin kestirimi için daha tutarlı ve sağlam algoritmalara ihtiyaç olduğu kolaylıkla görülmektedir. Özellikle verev açısı kestiricilerinin performansları büyük oranda artırılabilir. Birden fazla kaynağın bulunduğu ve gölge etkilerinin ya da ambiyans ve

aynasal aydınlatma bileşenlerinin de sürece dahil edildiği aydınlatma şartları, daha gerçekçi uygulamalarda kullanılmak üzere araştırılmalıdır.

Bütün yüzey türlerinde geçerli olabilecek sağlam ve otomatik bir tekniğin oluşturulmasının son derece zor hatta belki de imkansız olduğu görülmektedir (Gibbins vd., 1991). Verev açısı kestirimlerindeki hata oranlarının imge çözünürlüğü ile korelasyonu bulunduğu gösterilmiştir (Chojnacki vd., 1994). Bundan dolayı, ele alınan yeni yaklaşımların farklı imge çözünürlükleri için geliştirilmesi gerekmektedir. Ayrıca, tamamen Lambert yansıma kanununa göre davranmayan yüzeyleri de içeren daha kapsayıcı yansıma modelleri, albedo kestirimi ile birlikte kurgulandığında nihai sonuçları iyileştirebilir.

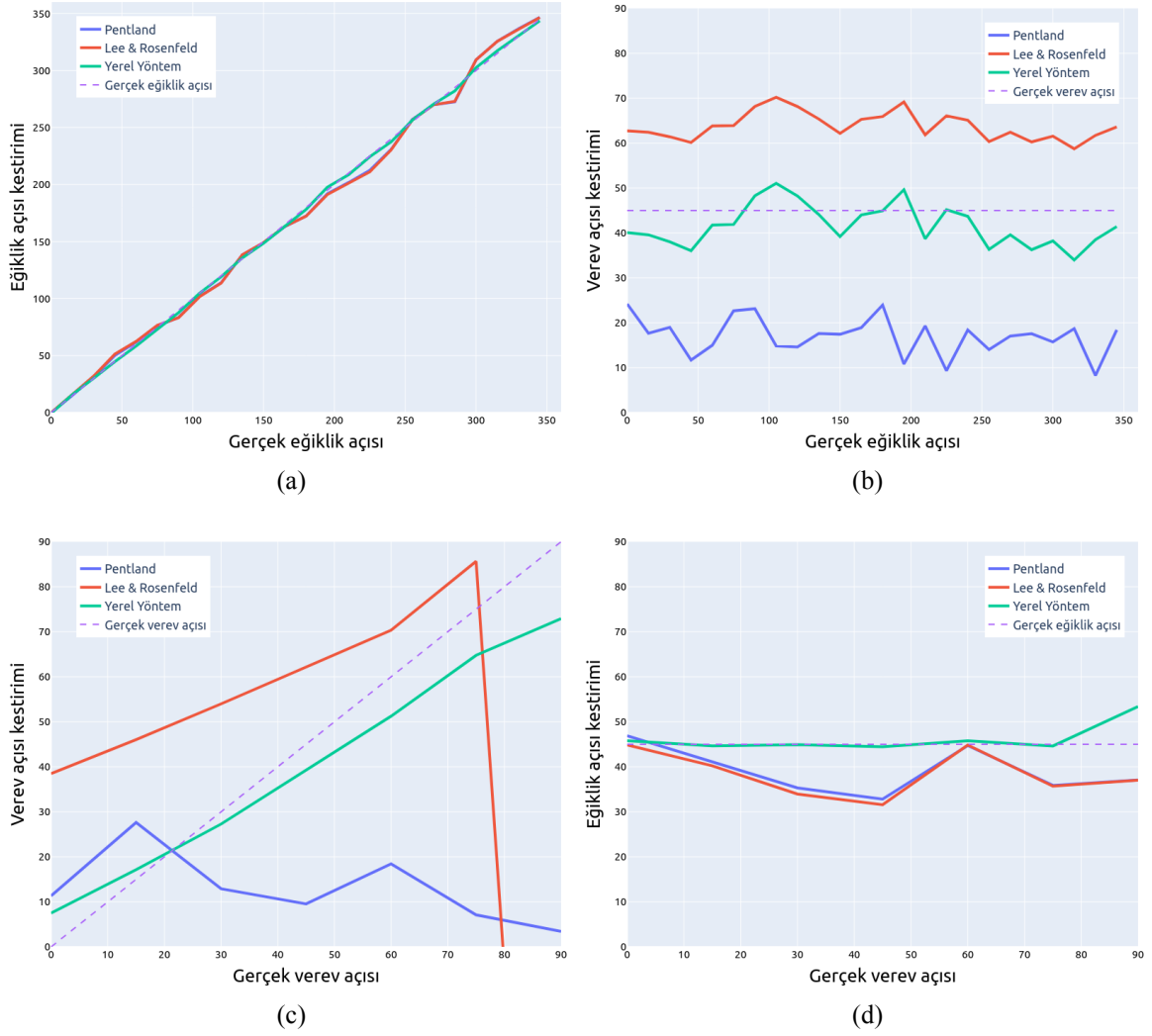
5.2 Kemelmacher-Shlizerman ve Basri'nin Yöntemine İlişkin Bulgular

Bölüm 3.6'da tanıtılan tek bir imge ve referans bir bireye ait derinlik haritasının kullanıldığı yöntemin uygulanmasında ve test edilmesinde kullanılan Texas ve Bosphorus veritabanlarında yer alan portre imgelerine karşılık gelen derinlik haritaları bulunmakla beraber bunlara karşılık gelen albedo haritaları yer almamaktadır. Bu yöntemin uygulanması sırasında ilk iki adımda, referans imgeye ait albedo haritası da girdi olarak algoritmada gerekmektedir. Bu yüzden, eldeki veritabanında yer alana her bir bireye ve imgeye karşılık gelen albedo haritaları Bölüm 3.7'de anlatılan yöntem ile hesaplanmıştır. Bu yöntemde gösterildiği üzere test veritabanından gelen herhangi bir imge için, aydınlatma kaynağı vektörünü bulduktan ve albedoyu tahmin ettikten sonra, 3.1 eşitliğiyle verilen Lambert yansıması ve görüntü denklemine göre; yüzey normali, albedo ve ışık vektörünü çarparak her bir piksel için görüntü tersten elde edilebilir:

$$I(x, y) = \rho(x, y) (l \cdot N(x, y)) \quad (5.1)$$

Burada, $\rho(x, y)$ pikselin sahip olduğu albedoyu, l aydınlatma vektörünü ve parantez içinde verilen ifade de bu aydınlatma vektörünün ilgili pikselin yüzey normali ile yapılan nokta çarpımını göstermektedir. Bu ifade ile hesaplanmış, Şekil 5.5'te verilen ve Texas Veritabanı'ndan alınan örnek imgelere ilişkin çıktı görüntüleri ile orijinal portre görüntüleri arasındaki hata oranları Çizelge-5.1'de gösterilmiştir.

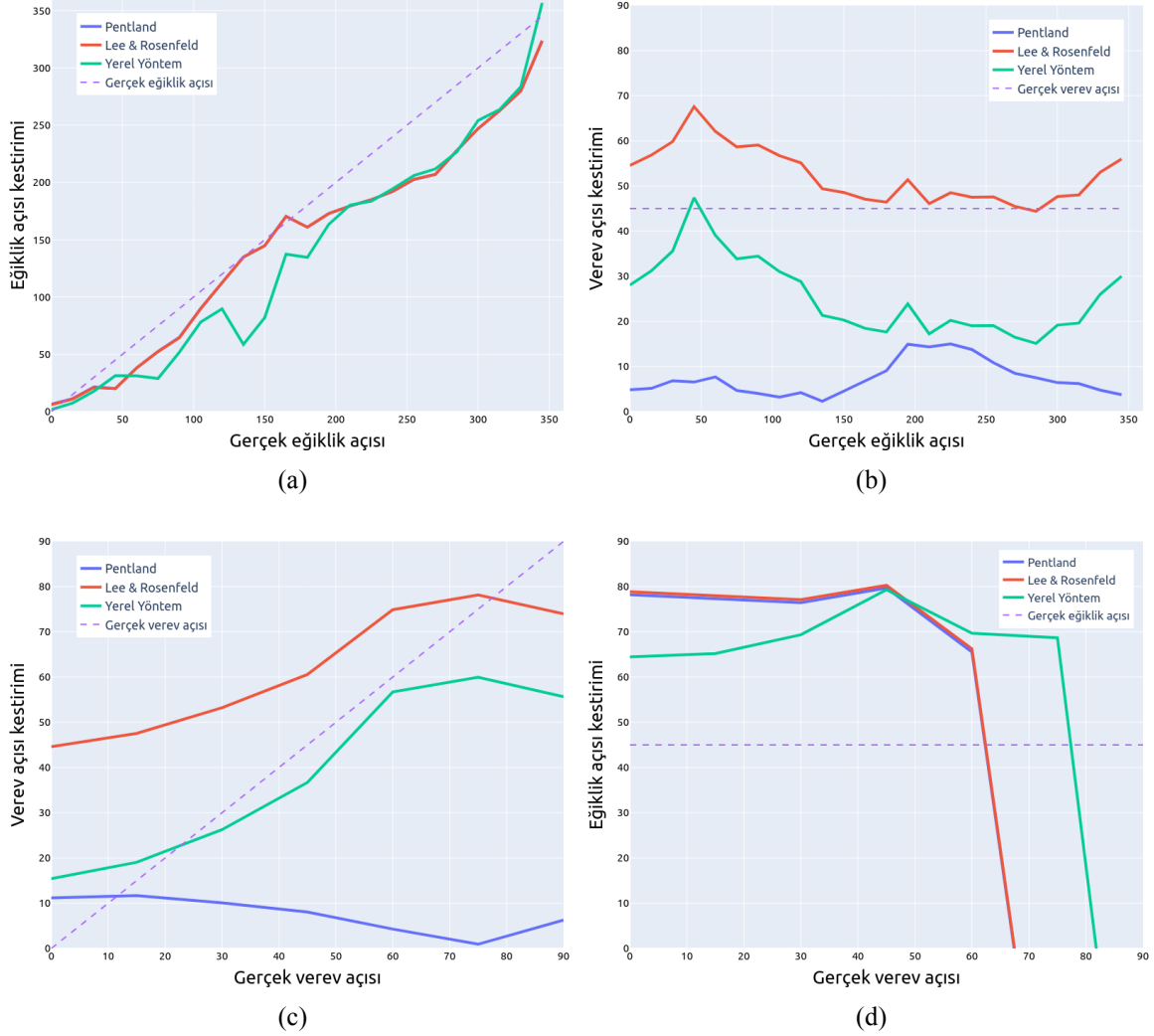
Çizelge-5.1'de görüleceği üzere, ortalama kare hatalar (MSE) oldukça düşüktür. Çıktı imgeleri orijinal portre imgelerine son derece yakındır. Böylece, bu uygulama ile elde edilen albedo haritalarının bu tez çalışmasının bu bölümünde ihtiyaç duyulan noktalarda



Şekil 5.3 Sentetik küre imgeleri için ışık kaynağı yönünün kestirimi. (a) eğiklik açısı kestirimlerini gerçek eğiklik açısı değerlerine karşı, (b) verev açısı kestirimini ($\phi = 45^\circ$) durumunda her eğiklik açısına karşı, (c) verev açısı kestirimini gerçek verev açısı değerlerine karşı, ve (d) eğiklik açısı kestirimini ($\tau = 45^\circ$) durumunda her verev açısına karşı göstermektedir.

Çizelge 5.1 Aydınlatma vektörleri ve albedo haritaları tahmin edildikten sonra 5.1 eşitliği kullanılarak elde edilen çıktı imgeleri ile orijinal portre görüntüleri arasındaki farkları gösteren çizelge. Test imgeleri Şekil 5.5'te gösterilmektedir.

Dosya no	MSE	PSNR (dB)
093	0.4×10^{-3}	23.98
160	2.4×10^{-4}	36.19
161	1.5×10^{-3}	28.38
268	1.2×10^{-3}	29.21



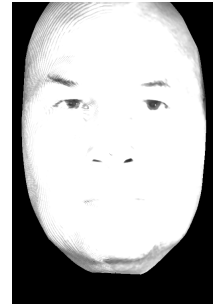
Şekil 5.4 Mozart imgeleri için ışık kaynağı yönünün kestirimi. (a) eğiklik açısı kestirimlerini gerçek eğiklik açısı değerlerine karşı, (b) verev açısı kestirimini ($\phi = 45^\circ$) durumunda her eğiklik açısına karşı, (c) verev açısı kestirimini gerçek verev açısı değerlerine karşı, ve (d) eğiklik açısı kestirimini ($\tau = 45^\circ$) durumunda her verev açısına karşı göstermektedir.



$$\mathbf{s} = [1.90199, 0.124466, -0.0360782]$$



$$\mathbf{s} = [3.11591, -0.170805, -0.995476]$$



$$\mathbf{s} = [1.66392, 0.13163, -0.0392276]$$



$$\mathbf{s} = [2.84757, 0.135103, -0.261519]$$

(a)

(b)

(c)

Şekil 5.5 Texas Veritabanı'nda yer alan 4 bireye ait albedo hesaplama yöntemi ile elde edilmiş albedo çıktı örnekleri (Dosya numaraları: 93, 160, 161, 268). a) Girdi test imgesi (portre görüntüsü), b) girdi test imgesinin derinlik haritası, c) albedo algoritması ile elde edilen albedo haritası. \mathbf{s} aydınlatma vektörü her birey için imgelerin altında verilmiştir.

kullanılabilir oldukları söylenebilir. Ortalama kare hata (MSE) ve doruk sinyal gürültü oranı (PSNR) değerleri,

$$MSE(I_2, I_1) = \frac{1}{jk} \sum_{j=1}^N \sum_{k=1}^M (I_2(j, k) - I_1(j, k))^2 \quad (5.2)$$

$$PSNR(I_2, I_1) = 10 \log_{10} \left(\frac{I_{max}^2}{MSE(I_2, I_1)^2} \right) \quad (5.3)$$

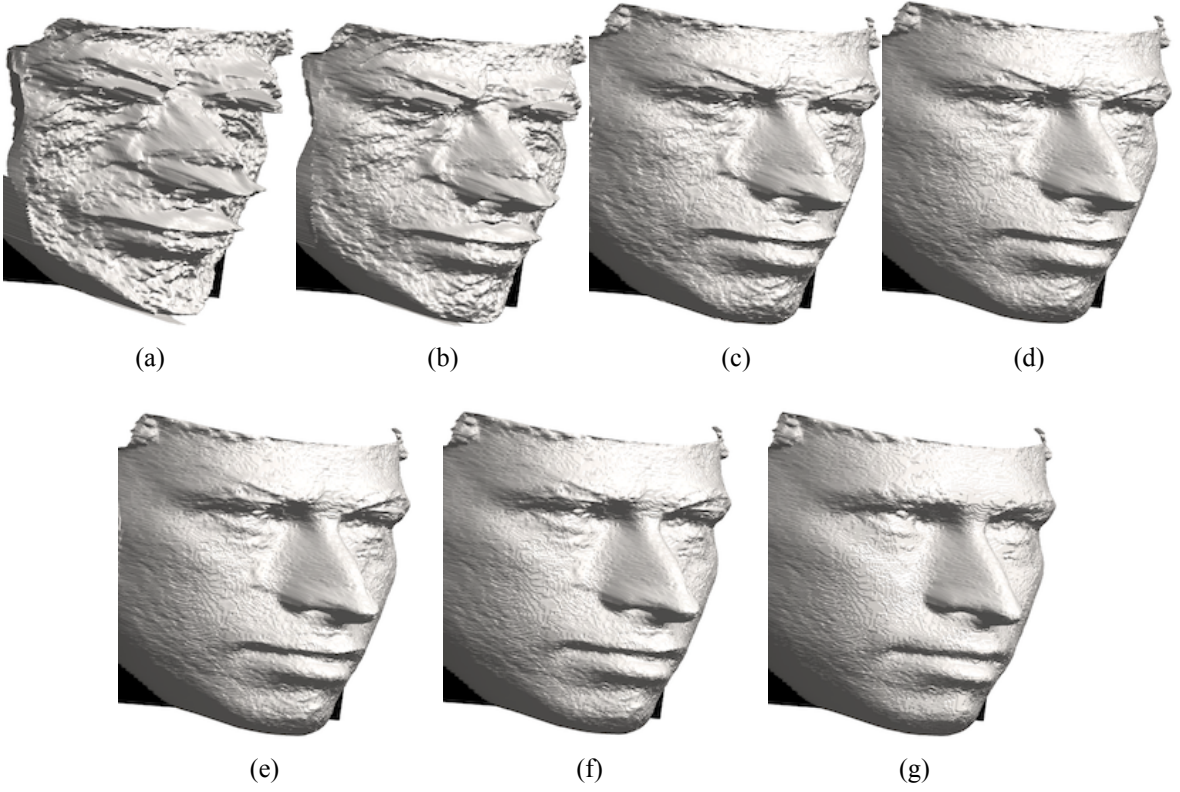
formülleriyle hesaplanmıştır. Burada, $I_{max} = 1.0$ alınmıştır.

Şekil-5.5'te Texas Veritabanı'nda yer alan bireylere ait görüntülere ilişkin hesaplanan kaynak vektörlerine ve bunlarla hesaplanan albedo çıktılarına örnekler verilmiştir.

Kemelmacher-Shlizerman ve Basri'nin yönteminin ikinci adımında yani derinliğin kestiriminde geçerli olan ve (3.28) eşitliğinde yer alan λ_1 regülasyon katsayısının derinlik bilgisinde etkisini göstermek üzere farklı değerlere ilişkin çıktılar Şekil 5.6'da gösterilmektedir. Bu katsayının algoritmanın çıktısını oldukça etkilediği görülmektedir. Bu parametrenin yüksek değerler alması, elde edilen yüzeyi referans yüzeye yaklaştırırken, daha düşük değerler girdi imgesine ait bireysel karakteristik özellikleri açığa çıkarmakta, bununla birlikte oluşacak şekilde bozulmaları da beraberinde getirmektedir. Böylece burada bir ödünleşim söz konusudur. Uygulamaya sokulan girdi test imgesine göre en uygun sonucu veren değer seçilmesi gerekmektedir.

Diğer önemli bir katsayı da Gausyen maskenin büyüklüğüdür. Bu tez çalışmasının bu aşamasında, maske büyüklüğü olarak çok sayıda farklı değer ile testler yapılmıştır ve 3×3 büyüklüğündeki bir kernel boyutunun hem performans hem de sonuca etkisi bakımından en iyi sonucu verdiği gözlemlenmiştir. Daha büyük boyutlardaki kernellerin sonucu gözle görülür bir şekilde iyileştirmediği kaydedilmemiştir.

Algoritmanın üç adımı da MATLAB ortamında geliştirilmiştir. Texas Veritabanı'nda yer alan 116 ait bireye ait birer portre imgesi seçilmiş ve algoritma çalıştırılarak bu imgelere ilişkin çıktılar elde edilmiştir. Elde, bu bireylere ait gerçek derinlik bilgisi bulunduğu için algoritmanın başarısı ölçülebilir. Bunu elde etmek için her bir piksel için gerçek derinlik değeri ile algoritma çıktısı sonucu kestirilmiş derinlik değerinin yüzde cinsinden farkı alınıp bütün bir imge için piksel ortalamaları elde edilmektedir. Böylece bir bireye ait hata oranı hesaplanmaktadır. Bütün test imgelerini içeren kümenin ortalaması ise algoritmanın genel hata oranını ve böylece başarımleri ölçüsünü



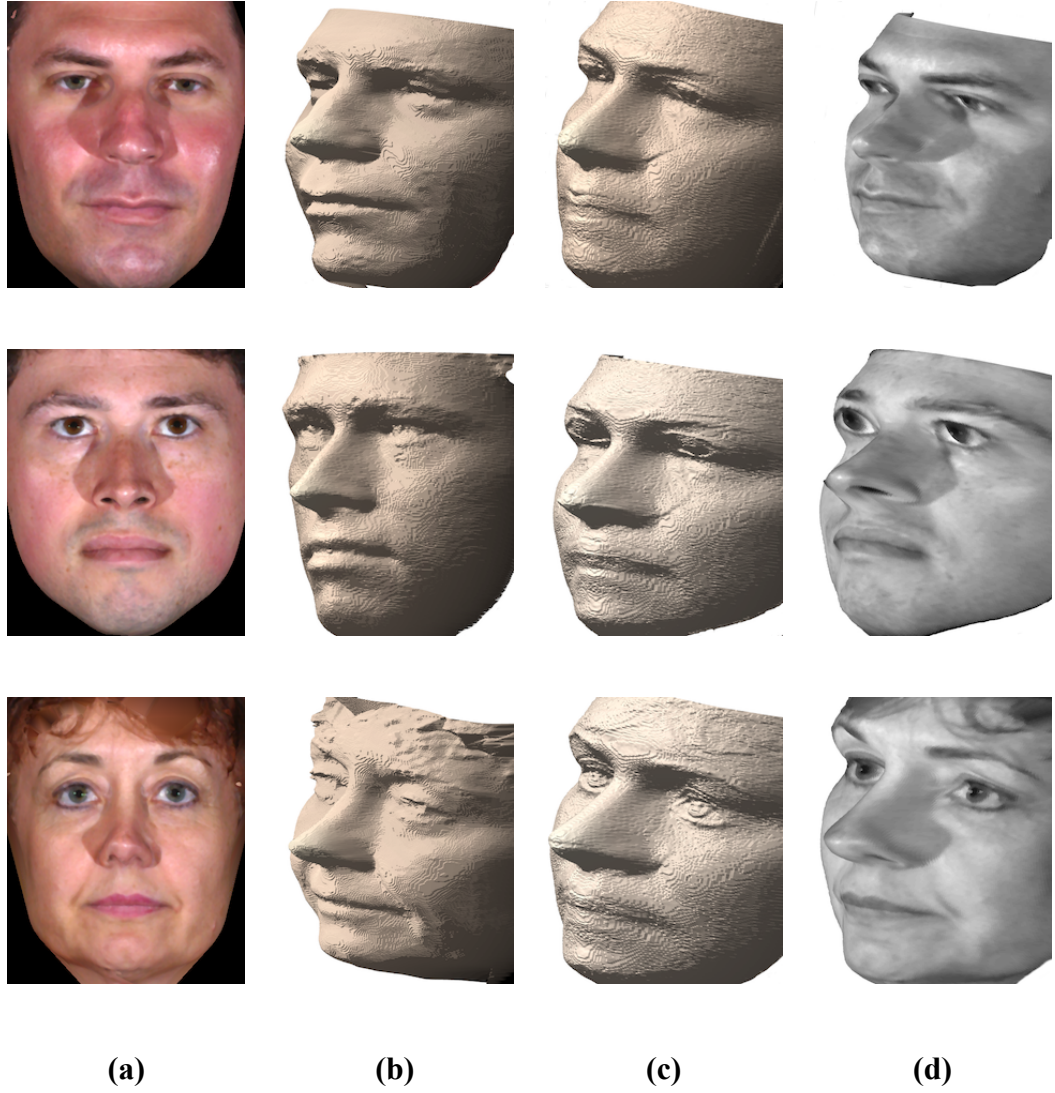
Şekil 5.6 Kemelmacher-Shlizerman ve Basri'nin yönteminin ikinci adımında kullanılan regülasyon katsayısı λ_1 'in algoritma sonucuna etkisini gösteren çıktı yüzey görüntüleri. λ_1 değerleri sırasıyla, 3, 5, 10, 15, 20, 25'tir. Son yüzey görüntüsü referans imgeye aittir.

verecektir. Buna göre bütün test imgelerini içeren küme için elde edilen hata ortalaması 6.45 ve standart sapma değeri de 3.27 olarak bulunmuştur.

Algoritmanın bütün adımları çalıştırdıktan sonra elde edilen çıktılara ilişkin örnekler Şekil 5.7'de görülebilir. Burada Texas Veritabanı'nda yer alan örnek bireylere ilişkin çıktılar verilmektedir. Sırasıyla, bireye ait portre görüntüsü, gerçek yüzey görüntüsü, tekrar oluşturulan yüzey ve üzerine bireye ait dokunun eklendiği yüzey görüntüsü yer almaktadır. Rasgele bireylere ait örnek çıktılar ise Şekil 5.8'de görülebilir. Benzer şekilde, sırasıyla girdi test imgesi, oluşturulan yüzey görüntüsü ve bireye ait yüzün eklendiği yüzey görüntüsü yer almaktadır.

Çizelge 5.2 Tek bir imge ve referans bireye ait derinlik haritası ile derinliğin kestirimi algoritması ile Texas Veritabanı'nda yer alan bireylere ilişkin çıktuların Yüzde Ortalama Mutlak Hata olarak ölçülen oranlar. Ortalama hata 6.45 ve standart sapma 3.27'dir.

Birey	Hata (%)	Birey	Hata (%)	Birey	Hata (%)
001	6.50	040	4.87	079	5.79
002	4.34	041	7.22	080	5.30
003	6.35	042	4.61	081	4.21
004	9.80	043	7.02	082	8.19
005	6.49	044	3.61	083	2.22
006	6.83	045	6.12	084	3.54
007	3.69	046	5.99	085	8.41
008	4.09	047	4.09	086	9.01
009	9.78	048	6.54	087	3.04
010	3.59	049	9.64	088	4.35
011	2.97	050	14.38	089	3.43
012	17.03	051	3.68	090	16.56
013	2.06	052	4.01	091	3.84
014	3.63	053	6.86	092	2.80
015	4.46	054	4.08	093	6.80
016	6.51	055	3.98	094	3.08
017	13.24	056	10.85	095	5.01
018	4.74	057	7.26	096	10.48
019	4.74	058	7.98	097	8.92
020	4.77	059	3.99	098	5.42
021	3.63	060	2.76	099	3.96
022	3.44	061	4.96	100	5.34
023	2.90	062	5.46	101	4.62
024	4.59	063	8.34	102	4.49
025	5.68	064	4.35	103	9.94
026	4.01	065	11.32	104	14.03
027	13.13	066	2.34	107	5.27
028	6.12	067	13.37	108	3.99
029	6.69	068	5.36	109	4.11
030	3.35	069	6.18	110	3.88
031	11.27	070	12.59	111	4.93
032	11.21	071	3.79	112	11.48
033	5.33	072	12.35	113	8.01
034	7.08	073	4.63	114	8.23
035	5.46	074	3.65	115	12.64
036	7.78	075	6.75	116	7.19
037	3.77	076	5.78	117	6.85
038	4.42	077	6.71	118	10.59
039	4.03	078	11.80		



Şekil 5.7 Tek bir imge ve referans bireye ait derinlik haritası ile derinliğin kestirimi algoritmasının Texas Veritabanı'nda yer alan bireylere ait imgelerle çalıştırılması sonucu elde edilen örnek çıktılar. a) İki boyutlu test girdi portre imgesi, b) Gerçek derinlik haritası ile oluşturulmuş yüzeyin bir açıdan alınmış anlık görüntüsü, c) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeyin bir açıdan alınmış anlık görüntüsü, d) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeye dokunun eklemesi sonucu elde edilen anlık görüntü.

5.3 Önerilen Yönteme İlişkin Bulgular ve Tartışma

Önerilen yönteme ilişkin ve Bölüm 4.2.6'da anlatılan deneylere ve farklı ağ yapılarına ait nicel karşılaştırmalar Çizelge 5.3'te verilmektedir. Başarım oranı, 100—*yüzde hata oranı* olarak hesaplanmıştır. Buna göre, en başarılı ağ yapısı olarak U-Net temelli üreteç, DCGAN'dan oluşan bir ayrıştırıcı, Wasserstein metriği ve L1 uzaklık teriminin

oluşturduğu kombinasyon olduğu görülmektedir. Test veri setinden gelen bazı imgelere ilişkin çıktı sonuçları Şekil 5.9'da gösterilmektedir. Şekilde soldan sağa doğru: a) Girdi imgesi, b) Test çıktısının rekonstrüksiyonun anlık bir görüntüsü, c) Referans gerçek derinlik haritasının rekonstrüksiyonun anlık bir görüntüsü, d) ve e) Çıktı rekonstrüksiyonlarının doku kaplamalı iki farklı görme açısından alınmış anlık görüntüleri, f) Referans gerçek değerli ve test çıktısı derinlik haritaları arasındaki farkı gösteren sıcaklık haritaları görülmektedir. Piksel bazlı hata oranları yukarıdan aşağıya doğru şu değerler olarak bulunmuştur: 5.385 ± 6.713 , 3.108 ± 3.074 , 0.649 ± 0.844 , 2.307 ± 1.937 ve 0.356 ± 0.882 .

Derinlik haritaları çıktıları önce bir nokta bulutu dosyasına dönüştürülmekte ve MeshLab yazılımı (Cignoni vd., 2008) kullanılarak yüzey rekonstrüksiyonu işlemi gerçekleştirilmektedir. Bu aşamada, noktalara ait normaller hesaplandıktan sonra Poisson yüzey rekonstrüksiyon yaklaşımı uygulanır. Bazı düzensizlikleri gidermek ve pürüzsüzlüğü sağlayabilmek için Laplace yüzey düzleştirme algoritması da bir sonraki adım da çalıştırılır. Burada seçilebilecek farklı düzleştirme algoritmaları söz konusudur.

Derinlik değerleri elde olan test veri setindeki yüz test imgelerinin haricinde, internet üzerinden indirilen bazı portre imgeleri de görsel inceleme amacıyla işlenmiştir. Bu çalışmaya ilişkin çıktılar da Şekil 5.10'da gösterilmektedir. Şekildeki her bir satırda, soldan sağa, a) girdi 2-boyutlu imge, b) çıktı derinlik haritası, c) yüzey rekonstrüksiyon edilmiş anlık görüntü, d) ve e) doku kaplamalı rekonstrüksiyon edilmiş ve iki farklı görme açısından alınmış anlık görüntüler sunulmaktadır. Veritabanlarından gelen test imgelerine ait çıktıların, bu imgelerle karşılaştırıldığında daha iyi ve pürüzsüz olduğu görülmektedir. Bunun sebebinin test imgeleri ile eğitim seti imgelerinin aynı kaynaktan gelmeleri, benzer aydınlatma ve görüntüleme şartları altında alınmaları olduğu söylenebilir. Bununla beraber, test için ayrılan imgeler hiçbir zaman eğitim sürecinde yer almamıştır. Ayrıca, bu imgeler için gerçek değerli referans derinlik haritaları elde bulunmadığından nicel performans ölçümü yapılamamaktadır.

Çizelge 5.3 Önerilen yöntemle derinlik kestirici olarak tasarlanan ağların yapılarını, kayıp fonksiyonlarını ve testlerin sonucunda ölçülen başarımlarını gösteren çizelge.

Üreteç Ağ	Ayrıtıcı Ağ	Kayıp Fonksiyonu	Başarımlar Oranı (%)
U-Net	PatchGAN	CGAN+L1	96.210 ±06.397
U-Net	PatchGAN	CGAN+L2	95.705 ±06.588
SHG	DCGAN	WGAN+L1	94.394 ±05.072
U-Net	DCGAN	WGAN+L1	97.970 ±03.403

Önerilen üretici yöntemde oluşabilecek olası aşırı uyumlanma durumlarını araştırmak için eğitim seti rasgele seçilmek üzere iki kategoriye ayrılmıştır; eğitim grubu ve geçerleme grubu. İlk grup tüm eğitim setinin %80'ini oluştururken, ikinci grup %20'sini oluşturmaktadır. Her bir epok sonrası üreteç ağının kayıp fonksiyonu çıktısı kaydedilmiştir ve Şekil 5.11'de gösterilmiştir. Geçerleme kayıp değerinin eğitim kayıp eğrisini belli bir uzaklıkta takip ederek monoton olarak azaldığı gözlemlenmektedir. Bu sonuç da, bu eğitim esnasında aşırı uyumlanma probleminin ortaya çıkmadığını göstermektedir.

5.3.1 Yakınsama sorunları ve öğrenme oranı optimizasyonu

GAN'ların eğitilmesinin oldukça zor olduğu, kararlı bir yakınsamanın gerçekleşmesinde ortaya çıkabilecek çok sayıda sorunun olduğu hem bu tez çalışmasında görülmüş, hem de diğer araştırmacılar tarafından belirtilmiştir. Bu çalışmada, eğitim sürecinin eniyileştirilmesi amacıyla birden fazla eniyileştirme algoritması araştırılmış ve testleri yapılmıştır. Bu algoritmalar arasında, Stokastik Gradyan İnişi (SGD), RMSProp, Adam ve Adamax (Kingma ve Ba, 2014), AdaGrad (Duchi vd., 2011), ADADELTA (Zeiler, 2012) ve Nadam yer almaktadır. Özet olarak, Adam algoritmasının derinlik kestirimi görevi için en kararlı eğitim sürecini sağladığı gözlemlenmiştir.

Eğitim sürecinde, monoton olarak azalan bir öğrenme oranının sabit bir öğrenme oranına göre çok daha iyi sonuç veren bir teknik olduğu görülmektedir. Bunu yaparken, epoklar ilerledikçe katmanların ağırlıklarının güncellenmesinde yavaş yavaş azalan bir öğrenme oranı kullanılır. Böylece eğitimi yapılan ağ, bu süreçte olabildiğince çok öğrenebilir. Fakat, bu azalmadaki değişimin derecesi oldukça dikkatli bir şekilde ayarlanmalıdır. Öğrenme oranı düzenleme fonksiyonu için birden fazla öneri söz konusudur. Bunlar arasında üstel azalış, tek adımda ya da birden çok adımda düzenleme, kosinüs tavlama (*annealing*), veya bir metriğin iyileşmesi durduğunda işlev gösteren öğrenme oranını düşürücü fonksiyonlar yer almaktadır. Bu tez çalışmasında, bu algoritmalar kapsamlı bir şekilde test edilmiş ve kosinüs fonksiyonuna bağlı bir düzenlemenin en etkin öğrenme oranı düşürücü yaklaşım olduğu gözlemlenmiştir. Buna göre, öğrenme oranı her bir eniyileştirme adımından hemen önce şu fonksiyona göre güncellenir (*How to adjust learning rate*):

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{T_{cur}}{T_{max}}\pi)). \quad (5.4)$$

Burada η_{max} ifadesi başlangıç öğrenme oranını, η_{min} ifadesi sondaki bitiş oranını, T_{cur} şu andaki ağırlık güncelleme adımını, T_{max} güncelleme adımlarının toplam sayısını ifade etmektedir. η_{min} çoğu uygulamada 0 olarak seçilmektedir. Bununla beraber, bizim

deneylerimizde η_{min} ifadesini η_{max} 'ın bir oranı, örneğin 1/20 olarak seçmenin daha iyi bir yakınsama sonucu doğurduğu gözlemlenmiştir.

5.3.2 Önerilen yöntemin önceki yöntemlerle karşılaştırılması

Bu tez çalışması kapsamında geliştirilen ve önerilen yöntem, önceki yöntemlerle nicel olarak karşılaştırılmış, önceki yöntemlere göre varsa kaydettiği ilerlemeler ve onlara göre olan dezavantajları araştırılmıştır. Elde edilen karşılaştırma sonuçları bu bölümde ortaya konulmaktadır.

5.3.2.1 Önerilen yöntemin otokodlayıcı yapılarla karşılaştırılması

Bu tez çalışmasında, derinlik kestirimi amacıyla önerilen GAN ağ yapısını, diğer derin öğrenmeye dayalı ağ yapıları ile karşılaştırmak için, Zhang, vd.nin (2017) geliştirdiği Daraltan Yığın Otokodlayıcı (SCAE) ağı uygulanmıştır. Daraltan otokodlayıcılar girdi verisini daha düşük boyutlarda temsil etme görevinde etkili olduklarını ispatlamış yapılarıdır. Zhang, vd.nin yönteminin akış şekli şu şekildedir: İlk olarak, 2-boyutlu girdi imgeleri için bir Yığın Otokodlayıcı ağı eğitilir, böylece bu imgeler daha düşük bir boyutta gösterilebilir şekilde kodlanır. Buna benzer başka bir otokodlayıcı ağı 3-boyutlu modeller yani nokta bulutları için de eğitilir. Bu iki ağ, sadece düz bir katmandan oluşan üçüncü bir ağ ile bir araya getirilir. Böylece birleştirilen üç parçalı bu yapı son kez eğitilir. Amaç, bir girdi 2-boyutlu imge verildiğinde bu imgeye karşılık gelen 3-boyutlu gösterimi, bütün ağın yukarıdan-aşağıya yapısı içerisinde kestirebilmektir.

Bu çalışmada uygulanan deneylerde, her otokodlayıcı ağ için yazarların önerisi olan 3-katmanlı yapı uygulanmıştır. Karşılaştırmaların doğru bir şekilde yapılabilmesi için, iki veritabanında da yer alan imgeler 70×70 piksel çözünürlüklü gri-seviye imgelere dönüştürülmüştür. Şurası aşıkardır ki, herhangi bir evrimsel katman olmadan orijinal imgelerin sadece düz doğrusal katmanlar ile işlenmesi çok büyük bir hesaplama gücü gerektirecektir. Örneğin, 10.000 adet nörona sahip bir katmana $256 \times 256 \times 3$ ölçeğinde bir girdi imgesi verildiğinde yaklaşık 2 milyar ağırlık katsayısı oluşur. Bu çalışmada gerçekleştirilen SCAE yapısı 2-boyutlu imgeler için $4900 - 500 - 100 - 10$, 3-boyutlu noktasal veriler içinse $14900 - 1000 - 100 - 10$ 'dur.

Derinlik haritalarının elde edilmesinden sonra, kök ortalama kare (RMS) hataları şu şekilde hesaplanmaktadır:

$$E_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\tilde{x}_{3D}(i) - x_{3D}(i))^2}. \quad (5.5)$$

SCAE ve üreteçli yöntemlerin RMS hata sonuçlarını karşılaştıran bir gösterim Şekil 5.12’de verilmektedir. Üreteçli yöntemlerin çok daha iyi sonuçlar verdiği aşıkardır. SCAE yönteminin beklenenden daha az başarılı olmasının arkasında yatan nedenlerin sunlar olduğu söylenebilir:

1. Zhang, vd.nin (2017) yürüttüğü çalışmada sentetik olarak üretilmiş yüz modelleri ve imgeleri kullanılmıştır. Bu da çözümü aranan problem için en ideal girdileri sağlamaktadır. Böylece eğitilmiş bir ağ yapısı, gerçek dokulara, albedolara ve gölgelemelere sahip gerçek bir imgeyle karşılaştığında kötü sonuçlar doğurmaktadır.
2. Otokodlayıcı ağlar, yapıları gereği kayıplı sıkıştırıcılar gibi davranış gösterirler. Çalışma akışlarında her bir katmanda girdi verisine ait belli bir oranda bilgi detayı yok olmaktadır. Bu çalışmada da, bu durum, imgelerin katman çıktılarındaki ara değerlerinden geri çatılması ve bunların katman girişlerindeki imgelerle karşılaştırılması sonucu gözlemlenmiştir.

Zhang, vd.nin (2017) çalışmasında rekonstrüksiyon hatalarını gösteren Tablo 2’de, geri çatma hatasının dördüncü katmanla birlikte belirgin bir şekilde artmasına karşılık, bu tez çalışmasında elde edilen sonuçlara göre bu durum üçüncü katmanla birlikte yaşanmaktadır. Bilgi detayındaki ani kayboluş, öznelik sayılarının giderek azalması neticesinde ortaya çıkmaktadır. Bu özellik, yüz imgelerinin diğer nesnelere ayrıştırılması işlemine aranabilecek bir durum olabilir. Bununla birlikte, bu durum bir yüzü diğerinden ayırt etmede kesinlikle istenmeyecek bir durumdur. Buna zıt olarak, U-Net ağının (Ronneberger vd., 2015) atlama katmanlarına sahip olması ile büyük ve gerçek imgelerin rekonstrüksiyonunda sağlam bir yapı ortaya koyduğu söylenebilir (256×256 ’ya karşılık 70×70 imge boyutları).

5.3.2.2 Önerilen yöntemin SFS yöntemleri ile karşılaştırılması

Bu tez çalışması kapsamında Kemelmacher-Shlizerman ve Basri’nin (2011) ortaya koyduğu tonlamadan şekil yöntemine (SFS) dayalı algoritma da üretici modellerle

karşılaştırılması için uygulanmıştır. Bahsedilen yöntem, derin öğrenmenin ortaya çıkmasından önceki döneme ilişkin son ve en gelişmiş tekniklerden biri olarak kabul edilmektedir. Yazarlar, yöntemlerinde 2-boyutlu bir yüz imgesi için derinlik kestirimini veri setinde yer alan bütün bireylerin ortalama yüz görüntüsünü referans olarak alan bir başka imge ile birlikte girdi olarak kullanmaktadırlar. Burada, referans bireye ait görüntünün derinlik haritası da bilinmektedir. Bununla birlikte, Lambert Yansıma Modeli'nin aydınlatma bileşeni için küresel harmonikler cinsinden açılım da uygulanmaktadır. Algoritmanın detaylı açıklaması Bölüm 3.6'da yapılmıştır. Buna göre, sırasıyla: 1) Referans imgenin derinlik verisi ve albedo haritası kullanılarak bir doğrusal açılım sonucu dört adet aydınlatma katsayısı kestirilir, 2) İlk adımda elde edilen aydınlatma katsayıları, referans imgenin albedo haritası ve düzenleme terimleri ile oluşan büyük bir doğrusal denklem kümesinin oluşturduğu bir seyrek matris çözümü ile her bir piksele ait derinlik değerleri kestirilir, 3) İkinci adıma benzer şekilde, girdi imgenin albedo bilgisi kestirilir. Son adımda, ilk iki adımda bulunan değerler kullanılır.

Bu tez çalışmasında, referans girdi imgesi için her bir veritabanındaki imgelerin ortalaması ayrı ayrı alınmıştır. Bu referans 2-boyutlu imge ve karşılığı olan derinlik haritası kullanılarak o veritabanında yer alan test imgelerinin kestirimi gerçekleştirilmiştir. Referans albedo haritasının kestirimi için, Brooks ve Horn'un (1985) noktasal ışık kaynağı kestirimi algoritması uygulanmış ve referans girdi imgenin bilinmeyen albedo haritası kestirilmiştir. Bütün bu işlemlerin nihai sonuçları Şekil 5.12'de gösterilmektedir. Bu algoritmanın oldukça hızlı çalışan bir yöntem olmasına ve herhangi bir eğitim işlemine ihtiyaç duymamasına karşılık, derinlik kestirimi probleminin çözümünde üreteçli modellerin genel olarak çok daha iyi bir sonuç verdiği görülmektedir.

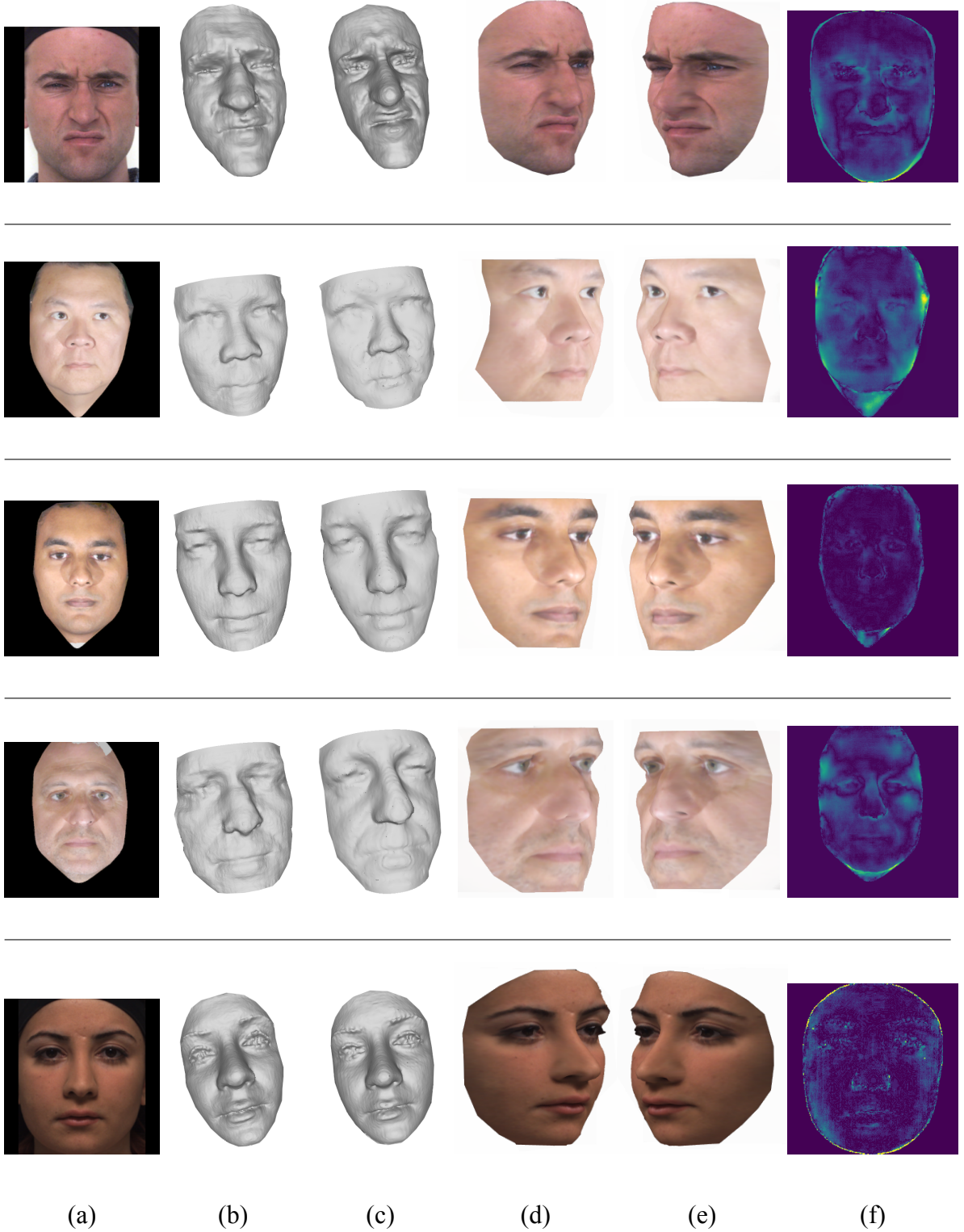


(a)

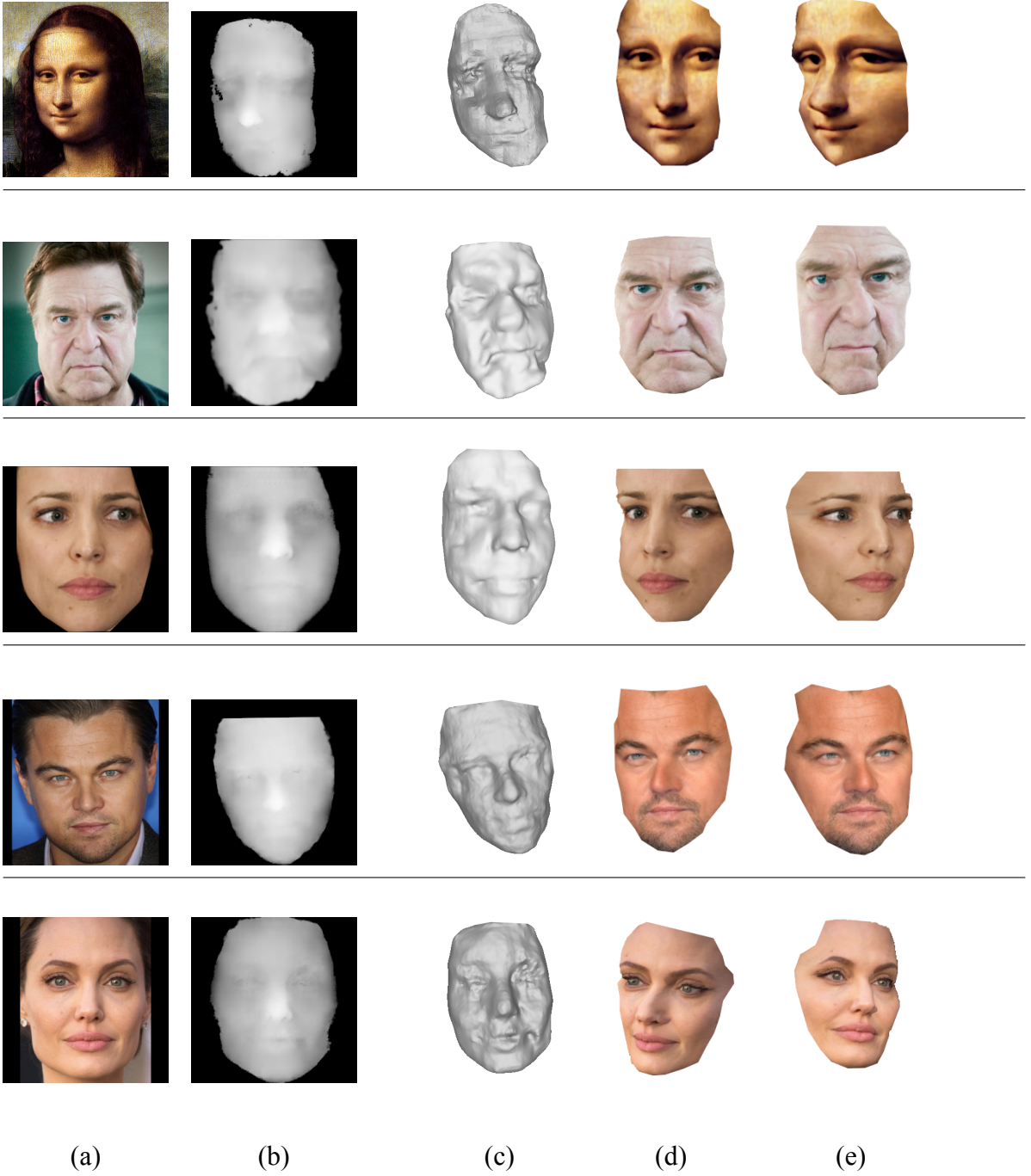
(b)

(c)

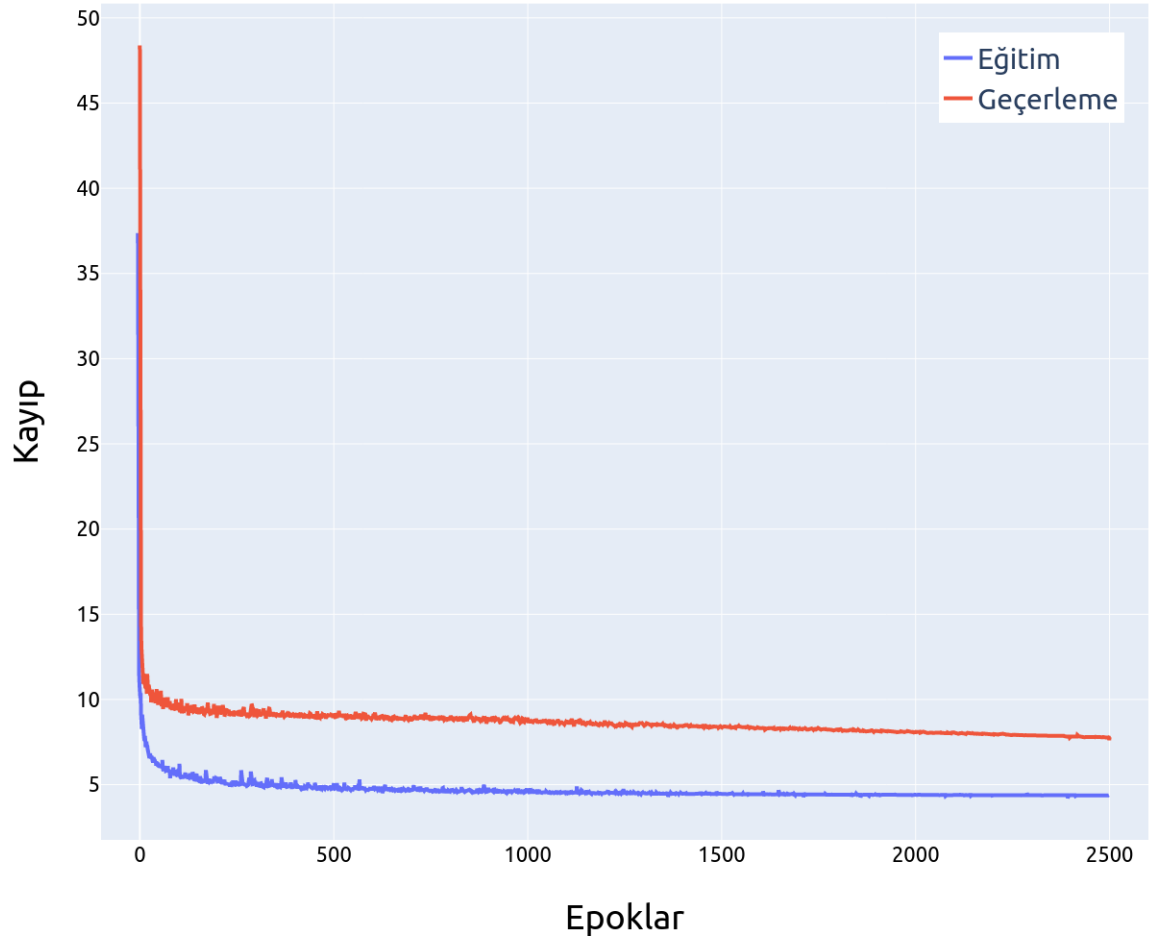
Şekil 5.8 Tek bir imge ve referans bireye ait derinlik haritası ile derinliğin kestirimi algoritmasının test veritabanında yer almayan üç rasgele bireye ait portre imgeleri ile çalıştırılması sonucunda elde edilen çıktılar. a) İki boyutlu girdi test imgesi, b) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeyin bir açıdan alınmış anlık görüntüsü, c) Algoritma çıktısında elde edilen derinlik haritası ile oluşturulmuş yüzeye girdi imgesindeki dokunun eklenmesi sonucu elde edilen anlık görüntü



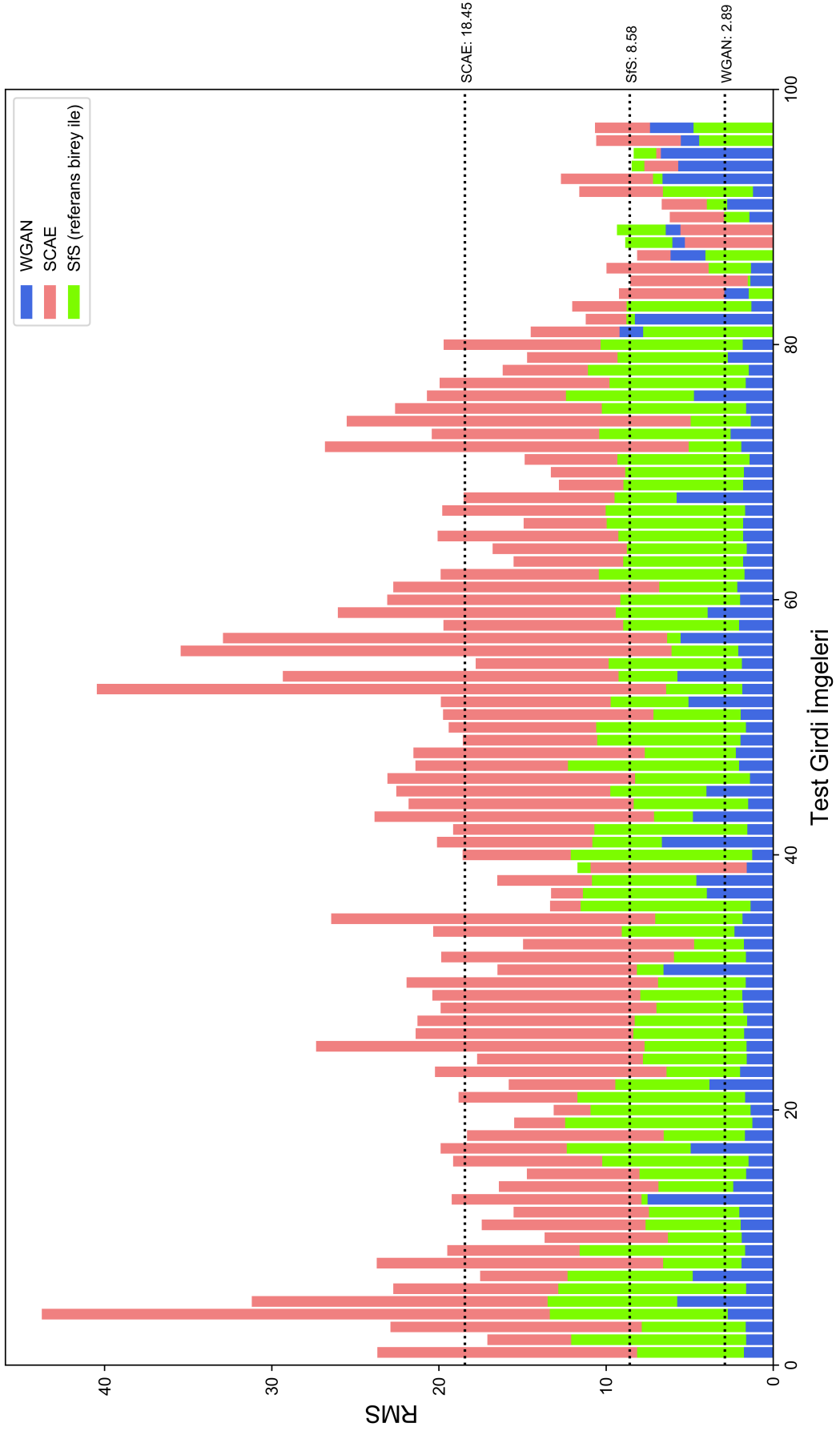
Şekil 5.9 Önerilen WGAN derinlik kestirici ağ için test çıktıları. a) İki boyutlu girdi test imgesi, b) Önerilen yöntemle elde edilen derinlik haritası kullanılarak rekonstrüksiyonu yapılan yüzeyin görüntüsü, c) Referans gerçek derinlik haritası kullanılarak rekonstrüksiyonu yapılan yüzeyin görüntüsü, d) ve e) Önerilen yöntem çıktı rekonstrüksiyonlarının doku kaplamalı iki farklı görme açısından alınmış anlık görüntüleri. (Test imgeleri veritabanlarından gelmektedir, fakat eğitim aşamasında kullanılmamışlardır.)



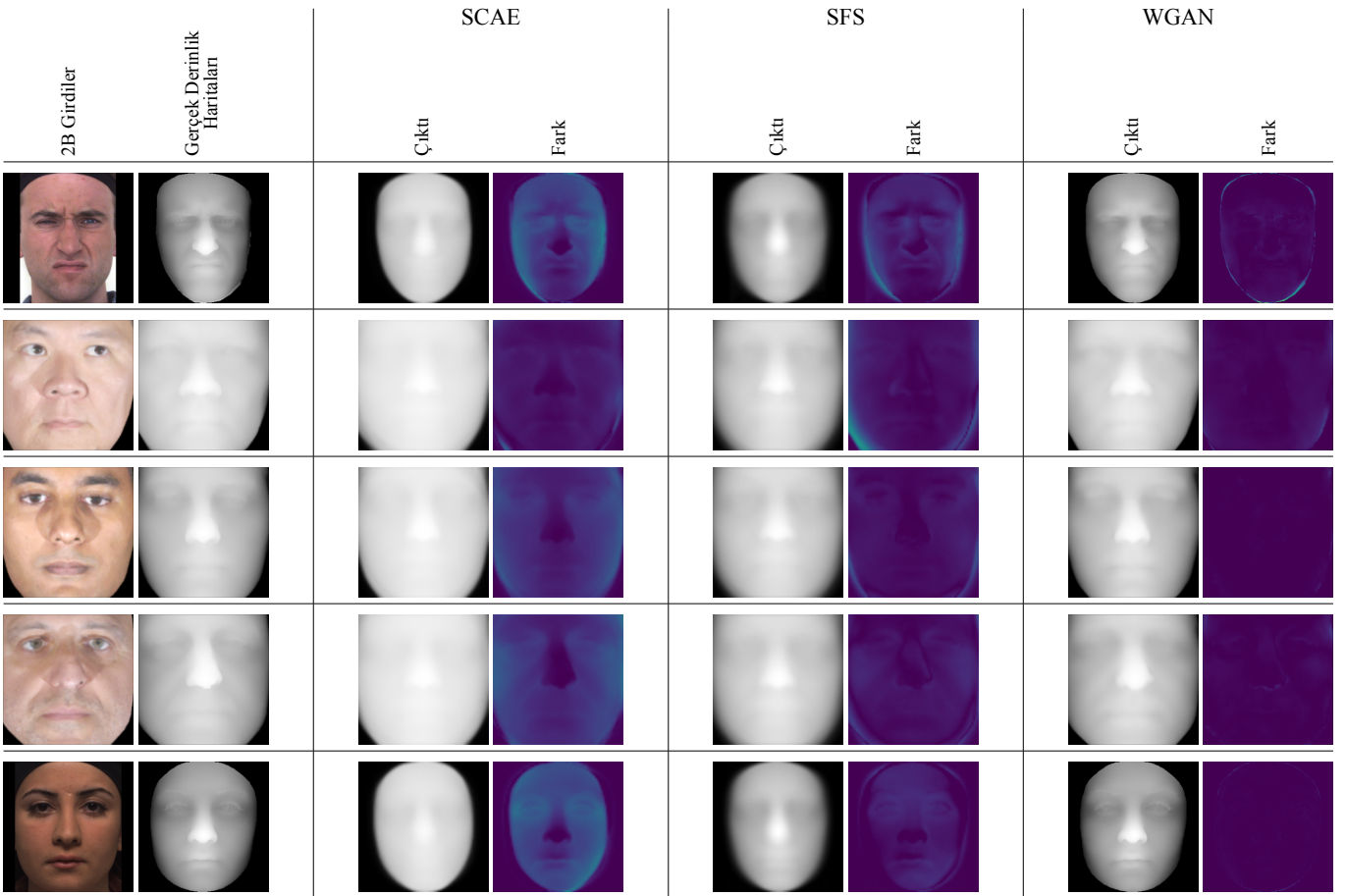
Şekil 5.10 İnternet üzerinden indirilen portre imgeleri için görsel inceleme amaçlı üretilmiş test çıktıları. a) İki boyutlu girdi test imgeleri, b) Önerilen yöntemle elde edilmiş çıktı derinlik haritaları, c) Derinlik haritası kullanılarak rekonstrüksiyon edilmiş yüzeyin görüntüsü, d) ve e) doku kaplamalı rekonstrüksiyon edilmiş ve iki farklı görme açılarından alınmış anlık görüntüler.



Şekil 5.11 Üreteç ağının kayıp fonksiyonu çıktısının her epok için çizilmesini gösteren şekil.



Şekil 5.12 a) WGAN, b) SCAE ve c) SFS yöntemleri için RMS hatalarının karşılaştırılması. Ortalama RMS değerleri dikey kesikli çizgilerle belirtilmiştir.



Şekil 5.13 a) WGAN, b) SCAE ve c) SFS yöntemleri için çıktı derinlik haritaları ve bunların gerçek değerlerle olan farkları.

6. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, ilk olarak gerçekleştirilen yöntem Kemelmacher-Shlizerman ve Basri'ye ait olan yaklaşımdır. Bu yöntem, detaylı olarak incelenmiş ve uygulaması yapılmıştır. Burada kullanılan ortalama referans imgenin, bütün test imgeleri için en iyi sonucu verdiği söylenemez. Herhangi bir girdi test imgesi için en iyi sonucu veren referans yüzeyin, o test imgesindeki yüze en yakın yüzey görüntüsü olacağı söylenebilir. Bunu başarmak üzere, farklı referans imgelerden oluşan ve PCA ya da benzeri bir algoritmik farklılaştırma yöntemiyle seçilen bir havuzdan faydalanmak gelecek çalışmalar için önerilebilir. Farklı bir yaklaşım tarzı da, yüze ait düz bölgeler ve farklılıkların belirgin ve hızlı değiştiği bölgeler için ayrı ayrı çalışan, yerleştirilmiş bir algoritmanın geliştirilmesi olabilir.

İnsan yüzüne ait görüntülerde, ani değişiklik gösteren bölgelere ait yüzeyler en az başarıyla oluşturulan kısımlar olarak görülmektedir. Bu bölgelerin başlıcaları; burun uçları, göz çukurları, göz çevresi ve ağızdır. Sınır bölgelerinde başarılı bir çıktı elde etmek için, girdi test imgesindeki bireyin nötr bir pozisyonda bulunuyor olması ve fotoğraf makinasına tam karşıdan bakıyor olması gerekmektedir.

Derinlik kestirimi problemi, doğası gereği eksik tanımlı yani çözüme yönelik yeterli veri içermeyen bir problemdir. Bununla birlikte, uzun süreden beri araştırmacılar bu problem üzerinde çalışmışlar ve literatürde yer almış çok sayıda yaklaşım, algoritma ve teknik geliştirmişlerdir. Bilgisayarla görü araştırmalarında son senelerde dikkat çeken yaklaşım tarzı, eskiden beridir üzerinde çalışılan problemlerin artan oranda, hızlı bir şekilde gelişme gösteren yapay zekâ tekniklerine ve özellikle derin öğrenme temelli çözümlere evrilmesidir.

Bu tez çalışmasında, insan yüzü içeren bir 2-boyutlu girdi imgesinden 3-boyutlu rekonstrüksiyon için derinlik kestirimi probleminin çözümüne yönelik yeni bir yöntem önerilmiştir. Bu yeni yaklaşım, yapay zekâ yaklaşımlarına ve özellikle derin öğrenme tekniklerine dayanmaktadır. Bu yöntemin temel bileşeni, Çekişmeli Üreteç Ağ (GAN) olarak adlandırılan bir ağ yapısıdır. Bu ağ yapısı, derinlik kestirimi problemi için geliştirilerek, düşük oranda hata payına sahip, hızlı ve kararlı bir yöntemin literatüre girmesi sağlanmıştır (Arslan ve Seke, 2019).

Bu tez çalışmasında, birden fazla GAN yapısı üzerinde deneyler gerçekleştirilmiş ve Koşullu Wasserstein GAN'ın (WGAN) sağlam ve iyi sonuçlar üreten bir çözüm olduğu sonucuna varılmıştır. Sunulan yöntem, başta geleneksel yöntemler olmakla birlikte diğer yöntemlerle nicel olarak karşılaştırılmış ve sonuçları yorumlanmıştır. Karşılaştırma yapılan yöntemlerden biri, başka bir derin öğrenme tabanlı olan ve otokodlayıcıları temel alan bir yaklaşımdır. Diğer bir yöntem ise, geleneksel olarak SFS yaklaşımına dayanan fakat küresel harmonik açılımlarını kullanan, alanında kabul görmüş bir çalışmadır. Öne sürülen ve karşılaştırılan yöntemler üzerinde yoğun olarak gerçekleştirilen testlerden sonra WGAN ağ yapısının tek bir 2-boyutlu girdi imgesinden insan yüzüne ait derinlik haritasının kestirilmesinde ve böylece 3-boyutlu yüz modelinin üretilmesi görevinde, diğer yöntemlerden daha başarılı çıktılar ürettiği sonucuna ulaşılmıştır.

Bu yaklaşımın ümit verici bir yöntem olmasına karşın, üreteçli modellerin eğitimleri esnasında büyük zorluklarla karşılaşılması, varılan sonuçlardan bir diğeridir. İleriki çalışmalarda, yeni modellerin, ağ yapılarının ve öğrenme yaklaşımlarının geliştirilmesi, ayrıca yakınsama problemlerini gidermek için eniyileştirme yöntemleri üzerinde çalışmalar yürütülmesi gerekmektedir. Ayrıca, 3-boyutlu biçimlenebilir modeller gibi bazı geleneksel yöntemlerin derin öğrenme teknikleri ile, özellikle de üreteçli modellerle birleştirilerek karmaşık bir işlem olan derinlik kestirimi probleminin çözümünde yeni arayışlara yönelmek mümkün olabilecektir.

KAYNAKLAR DİZİNİ

Alcantarilla, P. F., Bartoli, A. ve Davison, A. J. (2012), KAZE features, European Conference on Computer Vision: p. 214–227. Springer.

Alpaydın, E. (2013), Yapay Öğrenme. Boğaziçi Üniversitesi Yayınevi.

Arjovsky, M. ve Bottou, L. (2017), Towards principled methods for training generative adversarial networks, arXiv preprint arXiv:1701.04862,

Arjovsky, M., Chintala, S. ve Bottou, L. (2017), Wasserstein gan, arXiv preprint arXiv:1701.07875,

Arslan, A. T. ve Seke, E. (2015), Implementation of 3D face construction from a face image, 2015 9th International Conference on Electrical and Electronics Engineering (ELECO): p. 256–258. IEEE.

Arslan, A. T. ve Seke, E. (2017), Illuminant direction estimation methods, 2017 5th International Conference on Advanced Technology & Sciences (ICAT'17): p. 451–454.

Arslan, A. T. ve Seke, E. (2019), Face Depth Estimation With Conditional Generative Adversarial Networks, IEEE Access, 7: 23222–23231.

Atick, J. J., Griffin, P. A. ve Redlich, A. N. (1996), Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images, Neural Computation, 8(6): 1321–1340.

Basri, R. ve Jacobs, D. W. (2003), Lambertian reflectance and linear subspaces, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25(2): 218–233.

KAYNAKLAR DİZİNİ (devam)

- Bay, H., Tuytelaars, T. ve Van Gool, L. (2006), Surf: Speeded up robust features, European Conference on Computer Vision: p. 404–417. Springer.
- Berthelot, D., Schumm, T. ve Metz, L. (2017), BEGAN: boundary equilibrium generative adversarial networks, arXiv preprint arXiv:1703.10717,
- Bilişim Sözlüğü (2016), Türkiye Bilişim Derneği. <http://www.tbd.org.tr/index.php?sayfa=sozluk&mi1>. Erişim tarihi: 14.03.2016.
- Bishop, C. M. vd. (1995), Neural networks for pattern recognition. Oxford University Press.
- Biswas, S., Aggarwal, G. ve Chellappa, R. (2009), Robust estimation of albedo for illumination-invariant matching and shape recovery, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5): 884–899.
- Blake, A. (1985), Boundary conditions for lightness computation in Mondrian world, Computer Vision, Graphics, and Image Processing, 32(3): 314–327.
- Blanz, V. ve Vetter, T. (1999), A morphable model for the synthesis of 3D faces, Proceedings of the 26th annual conference on Computer graphics and interactive techniques: p. 187–194. ACM Press/Addison-Wesley Publishing Co.
- Blanz, V. ve Vetter, T. (2002), Reconstructing the complete 3D shape of faces from partial information, Informationstechnik und Technische Informatik, 44(6): 295–302.
- Blanz, V., Romdhani, S. ve Vetter, T. (2002), Face identification across different poses and illuminations with a 3d morphable model, Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on: p. 202–207. IEEE.

KAYNAKLAR DİZİNİ (devam)

- Bradski, G. (2000), The OpenCV Library, Dr. Dobb's Journal of Software Tools,
- Brooks, M. J. ve Horn, B. K. (1985), Shape and source from shading, Proceedings of the 9th international joint conference on Artificial intelligence-Volume 2: p. 932–936. Morgan Kaufmann Publishers Inc.
- Chojnacki, W., Gibbins, D. ve Brooks, M. J. (1994), Revisiting Pentland's estimator of light source direction, JOSA A, 11(1): 118–124.
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F. ve Ranzuglia, G. (2008), MeshLab: an Open-Source Mesh Processing Tool, Eurographics Italian Chapter Conference. Ed. V. Scarano, R. D. Chiara ve U. Erra. The Eurographics Association. ISBN: 978-3-905673-68-5.
- Cootes, T. F., Edwards, G. J. ve Taylor, C. J. (2001), Active appearance models, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6): 681–685.
- Davis, T. A. (2004), Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method, ACM Transactions on Mathematical Software (TOMS), 30(2): 196–199.
- Dovgard, R. ve Basri, R. (2004), Statistical symmetric shape from shading for 3D structure recovery of faces, *Computer Vision-ECCV 2004*. Springer: p. 99–113.
- Duchi, J., Hazan, E. ve Singer, Y. (2011), Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research, 12(Jul): 2121–2159.

KAYNAKLAR DİZİNİ (devam)

Faugeras, O., Luong, Q.-T. ve Papadopoulo, T. (2004), The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications. MIT press.

Georghiades, A. S., Belhumeur, P. N. ve Kriegman, D. J. (2001), From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6): 643–660.

Gibbins, D., Brooks, M. J. ve Chojnacki, W. (1991), Light source direction from a single image: a performance analysis, Australian Computer Journal, 23(4): 165–174.

Gonzalez, R. C. ve Woods, R. E. (2018), Digital Image Processing. 4th Edition. Pearson. ISBN: 9781292223049.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. ve Bengio, Y. (2014), Generative adversarial nets, Advances in Neural Information Processing Systems: p. 2672–2680.

Goodfellow, I., Bengio, Y. ve Courville, A. (2018), Derin Öğrenme. Buzdağı Yayınevi. ISBN: 9786058213296.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. ve Courville, A. C. (2017), Improved training of wasserstein gans, Advances in Neural Information Processing Systems: p. 5767–5777.

Gupta, S., Markey, M. K. ve Bovik, A. C. (2010a), Anthropometric 3D face recognition, International Journal of Computer Vision, 90(3): 331–349.

KAYNAKLAR DİZİNİ (devam)

- Gupta, S., Castleman, K. R., Markey, M. K. ve Bovik, A. C. (2010b), Texas 3D face recognition database, Image Analysis & Interpretation (SSIAI), 2010 IEEE Southwest Symposium on: p. 97–100. IEEE.
- Harris, C. G., Stephens, M., vd. (1988), A combined corner and edge detector., Alvey Vision Conference. Cilt 15 50: p. 10–5244. Citeseer.
- Hartley, R. ve Zisserman, A. (2003), Multiple View Geometry in Computer Vision. 2nd Edition. New York, NY, USA: Cambridge University Press. ISBN: 0521540518.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. ve Hochreiter, S. (2017), Gans trained by a two time-scale update rule converge to a local nash equilibrium, Advances in Neural Information Processing Systems: p. 6626–6637.
- Hobbel, M. (2013), 3D Face Reconstruction from a Single Image. Utrecht University.
- Horn, B. K. (1970), Shape from shading: A method for obtaining the shape of a smooth opaque object from one view, Doktora Tezi. Massachusetts Institute of Technology, Cambridge, MA.
- Horn, B. K. (1989), Obtaining Shape from Shading Information, Shape from Shaping. Ed. B. K. Horn ve M. J. Brooks. MIT Press: p. 123–171. ISBN: 978-0-262-51917-5.
- Huang, X., Li, Y., Poursaeed, O., Hopcroft, J. ve Belongie, S. (2017), Stacked generative adversarial networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: p. 5077–5086.

KAYNAKLAR DİZİNİ (devam)

- İdemem, M. (2015), Lineer Sınır-Değer Problemleri ve Özel Fonksiyonlar. İTÜ Vakfı Yayınları.
- Isola, P., Zhu, J.-Y., Zhou, T. ve Efros, A. A. (2017), Image-to-image translation with conditional adversarial networks, arXiv preprint,
- Jiang, D., Hu, Y., Yan, S., Zhang, L., Zhang, H. ve Gao, W. (2005), Efficient 3D reconstruction for face recognition, *Pattern Recognition*, 38(6): 787–798.
- Jolicoeur-Martineau, A. (2018), The relativistic discriminator: a key element missing from standard GAN, arXiv preprint arXiv:1807.00734,
- Kanatani, K., Sugaya, Y. ve Kanazawa, Y. (2016), *Guide to 3D Vision Computation: geometric analysis and implementation*. Springer.
- Karras, T., Aila, T., Laine, S. ve Lehtinen, J. (2017), Progressive growing of gans for improved quality, stability, and variation, arXiv preprint arXiv:1710.10196,
- Kemelmacher-Shlizerman, I. ve Basri, R. (2011), 3d face reconstruction from a single image using a single reference face shape, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2): 394–405.
- Kemelmacher, I. ve Basri, R. (2006), Molding face shapes by example, *Computer Vision–ECCV 2006*. Springer: p. 277–288.
- Kingma, D. P. ve Ba, J. (2014), Adam: A Method for Stochastic Optimization. arXiv: 1412.6980 [cs.LG].

KAYNAKLAR DİZİNİ (devam)

- Koppal, S. J. (2014), Lambertian Reflectance, *Computer Vision: A Reference Guide*. Ed. K. Ikeuchi. Boston, MA: Springer US: p. 441–443. ISBN: 978-0-387-31439-6.
- Krizhevsky, A., Sutskever, I. ve Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*: p. 1097–1105.
- Lee, C.-H. ve Rosenfeld, A. (1985), Improved methods of estimating shape from shading using the light source coordinate system, *Artificial Intelligence*, 26(2): 125–143.
- Leo, M. J. ve Manimegalai, D (2011), 3D modeling of human faces-A survey, *Trendz in Information Sciences and Computing (TISC), 2011 3rd International Conference on*: p. 40–45. IEEE.
- Leutenegger, S., Chli, M. ve Siegwart, R. Y. (2011), BRISK: Binary robust invariant scalable keypoints, *Computer Vision (ICCV), 2011 IEEE International Conference on*: p. 2548–2555. IEEE.
- Li, C. ve Wand, M. (2016), Precomputed real-time texture synthesis with markovian generative adversarial networks, *European Conference on Computer Vision*: p. 702–716. Springer.
- Li, Y. ve Shen, L. (2018), cC-GAN: A Robust Transfer-Learning Framework for HEp-2 Specimen Image Segmentation, *IEEE Access*, 6: 14048–14058.
- Liao, H.-b., Chen, Q.-h., Zhou, Q.-j. ve Guo, L. (2012), Rapid 3D face reconstruction by fusion of SFS and Local Morphable Model, *Journal of Visual Communication and Image Representation*, 23(6): 924–931.

KAYNAKLAR DİZİNİ (devam)

- Lowe, D. G. (2004), Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60(2): 91–110.
- Lucic, M., Kurach, K., Michalski, M., Gelly, S. ve Bousquet, O. (2017), Are gans created equal? a large-scale study, arXiv preprint arXiv:1711.10337,
- MXNet (2019), Apache MXNet. <https://mxnet.apache.org>. Erişim tarihi: 22.09.2019.
- Maghari, A., Venkat, I., Liao, I. Y. ve Belaton, B. (2013), PCA-based Reconstruction of 3D Face shapes using Tikhonov Regularization, *Int. J. Advance. Soft Comput. Appl*, 5(2).
- Maghari, A. Y., Liao, I. Y. ve Belaton, B. (2012), Effect of facial feature points selection on 3D face shape reconstruction using regularization, *Neural Information Processing*: p. 516–524. Springer.
- Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z. ve Smolley, S. P. (2017), Least squares generative adversarial networks, *Computer Vision (ICCV), 2017 IEEE International Conference on*: p. 2813–2821. IEEE.
- Marschner, S. R., Westin, S. H., Lafortune, E. P., Torrance, K. E. ve Greenberg, D. P. (1999), Image-based BRDF measurement including human skin, *Rendering Techniques '99*. Springer: p. 131–144.
- Matas, J., Chum, O., Urban, M. ve Pajdla, T. (2004), Robust wide-baseline stereo from maximally stable extremal regions, *Image and Vision Computing*, 22(10): 761–767.
- MathWorks (2015), MATLAB Optimization Toolbox Documentation. <http://www.mathworks.com/help/optim/index.html>. Erişim tarihi: 22.12.2015.

KAYNAKLAR DİZİNİ (devam)

- Mathieu, M., Couprie, C. ve LeCun, Y. (2015), Deep multi-scale video prediction beyond mean square error, arXiv preprint arXiv:1511.05440,
- Metz, L., Poole, B., Pfau, D. ve Sohl-Dickstein, J. (2016), Unrolled generative adversarial networks, arXiv preprint arXiv:1611.02163,
- Mingolla, E. ve Todd, J. T. (1986), Perception of solid shape from shading, *Biological Cybernetics*, 53(3): 137–151.
- Mirza, M. ve Osindero, S. (2014), Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784,
- Morel, J.-M. ve Yu, G. (2009), ASIFT: A new framework for fully affine invariant image comparison, *SIAM Journal on Imaging Sciences*, 2(2): 438–469.
- Newell, A., Yang, K. ve Deng, J. (2016), Stacked hourglass networks for human pose estimation, *European Conference on Computer Vision*: p. 483–499. Springer.
- Nilsson, N. J. (2011), *Yapay Zekâ: Geçmişi ve geleceği*. Boğaziçi Üniversitesi Yayınevi. ISBN: 9786056842511.
- OpenCV (Open Source Computer Vision Library) (2015). <http://opencv.org>.
- Patel, A. ve Smith, W. A. (2012), Driving 3D morphable models using shading cues, *Pattern Recognition*, 45(5): 1993–2004.

KAYNAKLAR DİZİNİ (devam)

- Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. ve Efros, A. A. (2016), Context encoders: Feature learning by inpainting, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: p. 2536–2544.
- Pentland, A. P. (1982), Finding the illuminant direction, JOSA, 72(4): 448–455.
- Prince, S. J. (2012), Computer vision: models, learning, and inference. Cambridge University Press.
- PyTorch Documentation, How to adjust learning rate. <https://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate>. Erişim tarihi: 2018-11-17.
- Radford, A., Metz, L. ve Chintala, S. (2015), Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434,
- Ramamoorthi, R ve Hanrahan, P (2001), On the Relationship between Radiance and Irradiance: Determining the Illumination from Images of a Convex Lambertian Object, J. Optical Society of America A, 18(10): 2448–2459.
- Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. ve Lee, H. (2016), Generative adversarial text to image synthesis, arXiv preprint arXiv:1605.05396,
- Ronneberger, O., Fischer, P. ve Brox, T. (2015), U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention: p. 234–241. Springer.

KAYNAKLAR DİZİNİ (devam)

- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. ve Chen, X. (2016), Improved techniques for training gans, *Advances in Neural Information Processing Systems*: p. 2234–2242.
- Savran, A., Alyüz, N., Dibekliolu, H., Çeliktutan, O., Gökberk, B., Sankur, B. ve Akarun, L. (2008), Bosphorus database for 3D face analysis, *European Workshop on Biometrics and Identity Management*: p. 47–56. Springer.
- Shimshoni, I., Moses, Y. ve Lindenbaum, M. (2000), Shape reconstruction of 3D bilaterally symmetric surfaces, *International Journal of Computer Vision*, 39(2): 97–110.
- Smith, W. A. ve Hancock, E. R. (2006a), Estimating facial albedo from a single image, *International Journal of Pattern Recognition and Artificial Intelligence*, 20(06): 955–970.
- Smith, W. A. ve Hancock, E. R. (2006b), Recovering facial shape using a statistical model of surface normal direction, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12): 1914–1930.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. ve Wojna, Z. (2016), Rethinking the inception architecture for computer vision, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*: p. 2818–2826.
- Szeliski, R. (2010), *Computer Vision: Algorithms and Applications*. Springer Science & Business Media.
- Tieleman, T. ve Hinton, G. (2012), Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural Networks for Machine Learning*, 4(2): 26–31.

KAYNAKLAR DİZİNİ (devam)

- Tuytelaars, T., Mikolajczyk, K., vd. (2008), Local invariant feature detectors: a survey, Foundations and Trends® in Computer Graphics and Vision, 3(3): 177–280.
- Türkçe Bilim Terimleri Sözlüğü (2019), Türkiye Bilimler Akademisi (TÜBA). <http://www.tubaterim.gov.tr>. Erişim tarihi: 13.03.2019.
- Wang, X. ve Gupta, A. (2016), Generative image modeling using style and structure adversarial networks, European Conference on Computer Vision: p. 318–335. Springer.
- Woodham, R. J. (1980), Photometric method for determining surface orientation from multiple images, Optical Engineering, 19(1): 191139.
- Worthington, P. L. (2005), Re-illuminating single images using Albedo estimation, Pattern Recognition, 38(8): 1261–1274.
- Zeiler, M. D. (2012), ADADELTA: An Adaptive Learning Rate Method. arXiv: 1212.5701 [cs.LG].
- Zhang, H., Goodfellow, I., Metaxas, D. ve Odena, A. (2018), Self-Attention Generative Adversarial Networks, arXiv preprint arXiv:1805.08318,
- Zhang, R., Tsai, P.-S., Cryer, J. E. ve Shah, M. (1999), Shape-from-shading: a survey, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(8): 690–706.
- Zhao, J., Mathieu, M. ve LeCun, Y. (2016), Energy-based generative adversarial network, arXiv preprint arXiv:1609.03126,

KAYNAKLAR DİZİNİ (devam)

- Zhao, W. Y. ve Chellappa, R. (2001), Symmetric shape-from-shading using self-ratio image, *International Journal of Computer Vision*, 45(1): 55–75.
- Zhao, W. ve Chellappa, R. (2000), Illumination-insensitive face recognition using symmetric shape-from-shading, *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. Cilt 1*: p. 286–293. IEEE.
- Zheng, Q. ve Chellappa, R. (1991), Estimation of illuminant direction, albedo, and shape from shading, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7): 680–702.
- Zhou, S. K., Aggarwal, G., Chellappa, R. ve Jacobs, D. W. (2007), Appearance characterization of linear lambertian objects, generalized photometric stereo, and illumination-invariant face recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2): 230–245.
- Zhu, J.-Y., Krähenbühl, P., Shechtman, E. ve Efros, A. A. (2016), Generative visual manipulation on the natural image manifold, *European Conference on Computer Vision*: p. 597–613. Springer.

EK AÇIKLAMALAR

Ek Açıklamalar - A: Doğrusal Olmayan Optimizasyon için MATLAB Betiği	105
Ek Açıklamalar - B: Kod Örnekleri	110
Ek Açıklamalar - C: Derin Öğrenme Tabanlı Yönteme İlişkin Kod Örnekleri	116

Ek Açıklamalar - A: Doğrusal Olmayan Optimizasyon için MATLAB Betiği

Aşağıda, Bölüm 3.6.4'te açıklanan MATLAB kodunda geçen ve doğrusal olmayan optimizasyonda her bir piksel için ilgili $F(x) = 0$ fonksiyonunu ve **J** Jakobian matrisi üreten betik verilmektedir:

```

1  % İmge boyutu (rows x cols)
2  % Referans derinlik matrisi: R
3  % Albedo haritası matrisi: albedo
4  % Test imgesi: I
5  % 9 adet aydınlatma harmonik katsayısı: l0, l1, l2, l3
6  %           l4, l5, l6, l7, l8 ve önceki adımda bulundu.
7
8  % Diğer sabitler:
9  a0 = pi;
10 a1 = 2*pi/sqrt(3);
11 a2 = 2*pi/sqrt(8);
12 c0 = 1/sqrt(4*pi);
13 c1 = sqrt(3/(4*pi));
14 c2 = 3*sqrt(5/(12*pi));
15
16 % Nref(x,y) her piksel için hesaplanıyor:
17 Nref = zeros(rows,cols);
18 for a=1:rows
19     for b=1:cols
20         if a ~= rows
21             q = R(a+1,b) - R(a,b);
22         else
23             q = 0;
24         end
25         if b ~= cols
26             p = R(a,b+1) - R(a,b);
27         else
28             p = 0;
29         end
30         Nref (a,b) = sqrt (p^2+q^2+1);
31     end
32 end

```

```

33
34 % Gausyen kernel 3x3 boyutunda ve sigma=0.15
35 kernel = fspecial('gaussian',3,0.15);
36 Cref = filter2(kernel,R);
37
38 % İndisleri tutmak için iki adet kolon vektörü tanımlanıyor:
39 S1 = zeros(0);
40 for j=1:rows-1
41     S1 = [S1, (j-1)*cols+1:1:j*cols-1];
42 end
43
44 S2 = zeros(0);
45 for j=2:rows-1
46     S2 = [S2, (j-1)*cols+2:1:j*cols-1];
47 end
48
49 ssize = length(S2);
50
51 % fun fonksiyonu burada başlıyor:
52 function [F,J] = fun(x)
53     % Girdi: kolon vektörü x
54     % Toplam piksel sayısı:
55     len = length(x);
56     step = 1;
57
58     % Toplam fonksiyon sayısı hesaplanıyor,
59     % ilk ve son satır ve kolonlar hariç:
60     fncs = (rows*cols+(rows-2)*(cols-2))-cols-rows+1;
61
62     % xi katsayıları hesaplanıyor: (Bknz: Bölüm 4.1.4)
63     % İlk üçü skalar:
64     d1 = (-a1*c1*l1*N - a1*c1*l2*N + a2*c2*l5 + a2*c2*l6);
65     d2 = (a1*c1*l1*N - a2*c2*l5);
66     d3 = (a1*c1*l2*N - a2*c2*l6);
67     % Diğerleri vektörel olmalı:
68     d4 = (a2*c2*l4)*ones(len,1);
69     d5 = (a2*c2*l7)*ones(len,1);
70     d6 = (-a2*c2*l7)*ones(len,1);

```

```

71     d7 = (-a2*c2*14-2*a2*c2*17)*ones(len,1);
72     d8 = (-a2*c2*14+2*a2*c2*17)*ones(len,1);
73     d9 = (a2*c2*14)*ones(len,1);
74
75     % F ve J başlatılıyor, J sparse:
76     F = zeros(fncs, 1);
77     J = sparse(fncs, len);
78
79     % Her bir piksel için F(x) = 0 hesaplanıyor:
80     for j=1:rows
81         for k=1:cols
82             ind = (j-1) * cols + k;
83             N = Nref(j,k);
84             % Son satır ve kolon hariç:
85             if (j~=rows) && (k~=cols)
86                 F(step) = -I(j,k) + ...
87                     (albedo(j,k) / N^2) .* ...
88                     (a0*c0*N^2*10 - a1*c1*N*13 + ...
89                     a2*c2*18*(3-N^2) + ...
90                     d1(ind) .* x(ind) + ...
91                     d2(ind) .* x(ind+1) + ...
92                     d3(ind) .* x(ind+cols) + ...
93                     d4(ind) .* x(ind)^2 + ...
94                     d5(ind) .* x(ind+1)^2 + ...
95                     d6(ind) .* x(ind+cols)^2 + ...
96                     d7(ind) .* x(ind)*x(ind+1) + ...
97                     d8(ind) .* x(ind)*x(ind+cols) + ...
98                     d9(ind) .* x(ind+1)*x(ind+cols) );
99                 step = step + 1;
100             end
101         end
102     end
103
104     % Regülasyondan kaynaklanan doğrusal
105     % denklemler de sisteme eklenmeli:
106     for j=1:rows
107         for k=1:cols
108             ind = (j-1) * cols + k;

```

```

109         F(step) = -1 * lambda * ...
110             (R(j,k)-Cref(j,k)) + ...
111             lambda * ( x(ind) - ( ...
112                 kernel(1,1) .* x(ind-cols-1) + ...
113                 kernel(1,2) .* x(ind-cols)   + ...
114                 kernel(1,3) .* x(ind-cols+1) + ...
115                 kernel(2,1) .* x(ind-1)       + ...
116                 kernel(2,2) .* x(ind)         + ...
117                 kernel(2,3) .* x(ind+1)       + ...
118                 kernel(3,1) .* x(ind+cols-1) + ...
119                 kernel(3,2) .* x(ind+cols)   + ...
120                 kernel(3,3) .* x(ind+cols+1)));
121         step = step + 1;
122     end
123 end
124 last = step - 1;
125
126 % Sparse J matrisi hazırlanmalı, hız açısından
127 % bu matris döngü içinde oluşturulmamalı, sparse
128 % matrisler toplamı olmalıdır. Her bir türev ifadesi
129 % için ayrı bir matris var.
130 % j_(k,i) ifadesi için:
131 J01 = sparse(S1, S1, d1(S1) + d7(S1) .* x(S1+1) + ...
132             d8(S1) .* x(S1+cols) + 2*d4(S1) .* x(S1), ...
133             last, len);
134
135 % j_(k,i+1) ifadesi için:
136 J02 = sparse(S1, S1+1, d2(S1) + d7(S1) .* x(S1) + ...
137             d9(S1) .* x(S1+cols) + 2*d5(S1) .* x(S1+1), ...
138             last, len);
139
140 % j_(k,i+n) ifadesi için (n=cols):
141 J03 = sparse(S1, S1+cols, d3(S1) + d8(S1) .* x(S1) + ...
142             d9(S1) .* x(S1+1) + 2*d6(S1) .* x(S1+cols), ...
143             last, len);
144
145 % Regülasyon maskesinden kaynaklı türevleri de içermeli:
146 S3 = ones(ssize,1);

```

```

147     kernel = -lambda * kernel;
148
149     J04 = sparse (S2, S2-cols-1, kernel(1,1)*S3, last, len);
150     J05 = sparse (S2, S2-cols , kernel(1,2)*S3, last, len);
151     J06 = sparse (S2, S2-cols+1, kernel(1,3)*S3, last, len);
152     J07 = sparse (S2, S2-1      , kernel(2,1)*S3, last, len);
153     J08 = sparse (S2,S2,(lambda+kernel(2,2))*S3, last, len);
154     J09 = sparse (S2, S2+1      , kernel(2,3)*S3, last, len);
155     J10 = sparse (S2, S2+cols-1, kernel(3,1)*S3, last, len);
156     J11 = sparse (S2, S2+cols  , kernel(3,2)*S3, last, len);
157     J12 = sparse (S2, S2+cols+1, kernel(3,3)*S3, last, len);
158
159     % Böylece, tüm alt sparse matrisler hazırlandı.
160     % Bunların toplamı J matrisini oluşturacak:
161     J = J01 + J02 + J03 + J04 + J05 + J06 ...
162         J07 + J08 + J09 + J10 + J11 + J12;
163
164     % Bu fonksiyon hem F fonksiyonları hem de J Jakobyen
165     % matrisi çıktısı olarak hazırlar. fun fonksiyonu sonu:
166     end

```

Ek Açıklamalar - B: Kod Örnekleri

Albedo ve tek nokta aydınlatma vektörü kestirimi: Brooks ve Horn'un (1985) gösterdiği yönteme göre albedo kestirimi yapan algoritma. n_x , n_y , n_z ifadeleri bilinen yüzey normalleridir. Çıktılar: Hesaplanan albedo haritası `albedo_map` ve tek nokta aydınlatma kaynağı vektörü `light`.

```

1 sum_n = zeros(3,3); sum_En = zeros(3,1);
2 for a=1:rows
3     for b=1:cols
4         n = [nx(a,b); ny(a,b); nz(a,b)];
5         sum_n = sum_n + (n*n');
6         sum_En = sum_En + P(a,b) * n;
7     end
8 end
9
10 light = sum_n \ sum_En; % inv(sum_n)*sum_En
11
12 albedo_map = zeros (rows,cols);
13
14 for a=1:rows
15     for b=1:cols
16         n = [nx(a,b); ny(a,b); nz(a,b)];
17         albedo_map (a,b) = P(a,b) / dot(light,n);
18     end
19 end

```

Girdi imgeler için küresel harmonikler cinsinden aydınlatma katsayılarının bulunması: Kemelmacher-Shlizerman ve Basri'nin (2011) gösterdiği şekilde girdi test imgesinin aydınlatma şartlarına ait dört harmonik katsayı hesaplayan kod örneği aşağıda verilmiştir. Girdiler: Referans imgeye ait yüzey normalleri n_x , n_y , n_z , referans imgeye ait albedo haritası `albedo_map` ve test girdi imgesi `Test`. Çıktı katsayıları `lcoeff` vektöründedir.

```

1 A = zeros(rows*cols,4);
2 a0 = pi;
3 a1 = 2*pi/sqrt(3);
4 c0 = 1/sqrt(4*pi);

```

```

5 c1 = sqrt(3/(4*pi));
6 step = 1;
7 for a=1:rows
8     for b=1:cols
9         rho = albedo_map(a,b);
10        A (step,1) = rho * a0 * c0;
11        A (step,2) = rho * a1 * c1 * nx(a,b);
12        A (step,3) = rho * a1 * c1 * ny(a,b);
13        A (step,4) = rho * a1 * c1 * nz(a,b);
14        step = step + 1;
15    end
16 end
17 lcoeff = A\(\reshape(Test',rows*cols,1));

```

Fotometrik Stereo Kod Örneği: Aşağıdaki örnekte, Yale-B veritabanında (Georghiades vd., 2001) yer alan imgelerle bir fotometrik stereo çözümü sunulmaktadır.

```

1 function [N,rho,depth1,depth2,depth3,randDepths,RX] =
   ↪ yaleBDepth(pathname, subject_name)
2
3 filename = fullfile(pathname, [subject_name '_P00_Ambient.pgm']);
4 ambimage = im2double(imread(filename));
5 [rows, cols] = size(ambimage);
6
7 d = dir(fullfile(pathname, [subject_name '_P00A*.pgm']));
8
9 filenames = {d(:).name};
10 total_images = numel(filenames);
11
12 Ang = zeros(2, total_images);
13 imarray = zeros(rows, cols, total_images);
14
15
16 for j = 1 : total_images
17     m = strfind(filenames{j}, 'A')+1;
18     Ang(1,j) = str2num(filenames{j}(m:(m+3)));
19     m = strfind(filenames{j}, 'E')+1;

```

```

20     Ang(2,j) = str2num(filenamees{j}(m:(m+2)));
21     imarray(:,:,j) =
        ↪ im2double(imread(fullfile(pathname,filenamees{j})));
22 end
23
24 [X,Y,Z]= sph2cart(pi*Ang(1,+)/180,pi*Ang(2,+)/180,1);
25 lightdirs = [Y;Z;X];
26 lightdirs = lightdirs';
27
28 % Her bir imgeden Ambient çıkarılıyor:
29 for j = 1 : total_images
30     imarray(:,:,j) = imarray(:,:,j) - ambimage;
31 end
32 imarray(find(imarray(:)<0)) = 0.0;
33
34 % Veri tekrar ölçekleniyor:
35 for j = 1 : total_images
36     imarray(:,:,j) = mat2gray(imarray(:,:,j));
37 end
38
39 rho = zeros(rows,cols);
40 N = zeros(rows,cols,3);
41 f_1 = zeros(rows,cols);
42 f_2 = zeros(rows,cols);
43 X = zeros(rows,cols,3);
44
45 % Fotometrik Stereo (her bir piksel için)
46 for i=1:rows
47     for j=1:cols
48         b = reshape(imarray(i,j,:),[total_images 1]);
49         A = lightdirs;
50         x = A\b;
51         X(i,j,:) = x;
52         rho(i,j) = sqrt(x(1)^2 + x(2)^2 + x(3)^2);
53         N(i,j,:) = x./rho(i,j);
54         f_1(i,j) = x(1) / x(3);
55         f_2(i,j) = x(2) / x(3);
56     end

```



```

57 end
58
59 F_1 = cumsum(f_1,2); % her bir satır için sütunlar boyunca
60                     % kümülatif toplam
61 F_2 = cumsum(f_2,1); % her bir sütun için satırlar boyunca
62                     % kümülatif toplam
63
64 depth1 = zeros(rows,cols); % önce en üst satır için
65 depth2 = zeros(rows,cols); % önce en sol sütun için
66
67 for i=1:rows
68     for j=1:cols
69         depth1(i,j) = F_1(1,j) + F_2(i,j);
70         depth2(i,j) = F_2(i,1) + F_1(i,j);
71     end
72 end
73
74 % Rastgele yollardan toplam alma:
75 randNo = 5;
76 randDepths = zeros(rows,cols,randNo);
77
78 % Rastgele yollar
79 for i=1:rows
80     for j=1:cols
81         for z=1:randNo
82             dd = f_1(1,1) + f_2(1,1);
83             pathi = (i-1) + (j-1);
84             rast = randi(2,pathi,1);
85             ai = 1; aj=1;
86             for k=1:pathi
87                 if ai == i
88                     aj = aj + 1;
89                     dd = dd + f_1(ai,aj);
90                     continue
91                 end
92                 if aj == j
93                     ai = ai + 1;
94                     dd = dd + f_2(ai,aj);

```

```

95         continue
96     end
97     if rast(k) == 1
98         ai = ai + 1;
99         dd = dd + f_2(ai,aj);
100    end
101    if rast(k) == 2
102        aj = aj + 1;
103        dd = dd + f_1(ai,aj);
104    end
105    end
106    randDepths(i,j,z) = dd;
107 end
108 end
109 end
110
111 % Doğrusal sistem çözümü Ax=b
112 t1 = ones(rows * cols * 12,1);
113 t2 = ones(rows * cols * 12,1);
114 t3 = zeros(rows * cols * 12,1);
115 b = zeros(rows * cols,1);
116 stp = 1; b_stp = 1;
117
118 for i=1:rows
119     for j=1:cols
120         ind = (i-1) * cols + j;
121         if (j==cols) || (i==rows)
122             t1(stp) = ind;
123             t2(stp) = ind;
124             t3(stp) = 2.0;
125             stp = stp + 1;
126             %b(ind) = f_1(i,j) + f_2(i,j);
127             %b_stp = b_stp + 1;
128             b(ind) = 0.0;
129             b_stp = b_stp + 1;
130         continue
131     end
132     t1(stp) = ind;

```

```

133     t2(stp) = ind+1;
134     t3(stp) = 1.0;
135     stp = stp + 1;
136
137     t1(stp) = ind;
138     t2(stp) = ind+cols;
139     t3(stp) = 1.0;
140     stp = stp + 1;
141
142     t1(stp) = ind;
143     t2(stp) = ind;
144     t3(stp) = -2.0;
145     stp = stp+1;
146
147     b(b_stp) = f_1(i,j) + f_2(i,j); b_stp = b_stp + 1;
148
149     end
150 end
151
152 refP = imread('refmean_P.png');
153 refP = im2double(rgb2gray(imresize(refP,[192 168])));
154
155 refR = imread('refmean_R.png');
156 refR = im2double(imresize(refR,[192 168]));
157
158 meanRand = mean(randDepths,3);
159 x0 = reshape(refR',[192*168 1]);
160 x1 = reshape(meanRand',[192*168 1]);
161 A = sparse(t1,t2,t3,rows*cols,rows*cols);
162 x3 = A\b;
163 depth3 = reshape(x3,[cols rows]);
164 depth3 = depth3';
165
166 end

```



```

32         encoder = [en_conv]
33         decoder = [de_relu, de_conv,
34                   Activation(activation='tanh')]
35         model = encoder + [inner_block] + decoder
36     else:
37         de_conv = Conv2DTranspose(channels=outer_channels,
38                                  kernel_size=4, strides=2, padding=1,
39                                  in_channels=inner_channels * 2, use_bias=use_bias)
40         encoder = [en_relu, en_conv, en_norm]
41         decoder = [de_relu, de_conv, de_norm]
42         model = encoder + [inner_block] + decoder
43     if use_dropout:
44         model += [Dropout(rate=0.5)]
45     self.model = HybridSequential()
46     with self.model.name_scope():
47         for block in model:
48             self.model.add(block)
49     def hybrid_forward(self, F, x):
50         if self.outERMOST:
51             return self.model(x)
52         else:
53             return F.concat(self.model(x), x, dim=1)
54 class UnetGenerator(HybridBlock):
55     def __init__(self, in_channels, num_downs, ngf=ngf,
56                 use_dropout=True):
57         super(UnetGenerator, self).__init__()
58         unet = UnetSkipUnit(ngf * 8, ngf * 8, innermost=True)
59         for _ in range(num_downs - 5):
60             unet = UnetSkipUnit(ngf * 8, ngf * 8, unet,
61                                 use_dropout=use_dropout)
62         unet = UnetSkipUnit(ngf * 8, ngf * 4, unet)
63         unet = UnetSkipUnit(ngf * 4, ngf * 2, unet)
64         unet = UnetSkipUnit(ngf * 2, ngf * 1, unet)
65         unet = UnetSkipUnit(ngf, in_channels, unet, outermost=True)
66         with self.name_scope():
67             self.model = unet
68     def hybrid_forward(self, F, x):
69         return self.model(x)

```

Ayrıtaç Ağ Modelinin Oluşturulması: Bölüm 4.2.2’de anlatılan DCGAN (Radford vd., 2015) ayrıtaç ağ modelinin kurulmasını gerçekleştiren kod bloğu.

```

1 class DCGAN_ayirtac(HybridBlock):
2     def __init__(self):
3         super(DCGAN_ayirtac, self).__init__()
4         with self.name_scope():
5             self.model = HybridSequential()
6             self.model.add(Conv2D(channels=128, in_channels=6,
7                                   kernel_size=4, strides=2, padding=1,
8                                   use_bias=False))
9             self.model.add(LeakyReLU(alpha=0.2))
10            self.model.add(Conv2D(channels=256, in_channels=128,
11                                   kernel_size=4, strides=2, padding=1,
12                                   use_bias=False))
13            self.model.add(BatchNorm())
14            self.model.add(LeakyReLU(alpha=0.2))
15            self.model.add(Conv2D(channels=512, in_channels=256,
16                                   kernel_size=4, strides=2, padding=1,
17                                   use_bias=False))
18            self.model.add(BatchNorm())
19            self.model.add(LeakyReLU(alpha=0.2))
20            self.model.add(Conv2D(channels=1024, in_channels=512,
21                                   kernel_size=4, strides=2, padding=1,
22                                   use_bias=False))
23            self.model.add(BatchNorm())
24            self.model.add(LeakyReLU(alpha=0.2))
25            self.model.add(Conv2D(channels=1, in_channels=1024,
26                                   kernel_size=4, strides=1, padding=1,
27                                   use_bias=False, activation='sigmoid'))
28        def hybrid_forward(self, F, input):
29            output = self.model(input)
30            return output

```

Eğitim süreci. Wasserstein GAN için eğitim gerçekleştiren kod bloğu.

```

1  ''' Üreteç ve ayırtaç ağlar oluşturulur ve Xavier ilklendirme
2  algoritması ile katsayıların ilk değerleri atanır. Ayrıca
3  her bir ağ için Adam optimizasyon fonksiyonu oluşturulur.
4  '''
5  netG = UnetGenerator(in_channels=3, num_downs=8, use_dropout=True)
6  netD = DCGAN_ayirtac()
7  netG.collect_params().initialize(mx.init.Xavier(), ctx=baglam)
8  netD.collect_params().initialize(mx.init.Xavier(), ctx=baglam)
9  trainerG = mx.gluon.Trainer(netG.collect_params(), 'adam',
10     {'learning_rate': 0.0002})
11  trainerD = mx.gluon.Trainer(netD.collect_params(), 'adam',
12     {'learning_rate': 0.0002})
13
14  ''' Her epok içinde tüm imge çiftleri batch'ler halinde
15  işlenecektir. İmgeler, [-1, 1] arasına normalize edilir.
16  İmge tensör yapısı şu şekildedir: [batch, kanallar, en, boy]
17  '''
18  epok_sayisi = 1000
19  for epok in range(epok_sayisi):
20     girdi_data.reset()
21     cikti_data.reset()
22     for girdi, cikti in zip(girdi_data, cikti_data):
23         for p in netD.collect_params().values():
24             p.data[:] = mx.nd.clip(p.data[:],
25                 -0.01, 0.01)
26         gercek_girdi = (girdi / 127.5) - 1.0
27         gercek_cikti = (cikti / 127.5) - 1.0
28         sahte_cikti = netG(gercek_girdi)
29
30     # Ayırtaç ağın eğitimi
31     with mx.autograd.record():
32         # Gerçek imgelerle eğitim
33         gercek_girdi_cikti = mx.nd.concat(gercek_girdi,
34             gercek_cikti, dim=1)
35         hataD_gercek = netD(gercek_girdi_cikti)
36         hataD_gercek.backward()
37

```



```
38     # Sahte imgelerle eğitim
39     sahte_girdi_cikti = mx.nd.concat(gercek_girdi,
40                                     sahte_cikti, dim=1)
41     hataD_sahte = netD(sahte_girdi_cikti)
42     hataD_sahte.backward()
43     hataD = hataD_gercek - hataD_sahte
44     trainerD.step(batch_sayisi)
45
46     # Üreteç ağın eğitimi
47     with mx.autograd.record():
48         sahte_cikti = netG(gercek_girdi)
49         sahte_girdi_cikti = mx.nd.concat(gercek_girdi,
50                                         sahte_cikti, dim=1)
51         hataG = netD(sahte_girdi_cikti)
52         hataG.backward()
53     trainerD.step(batch_sayisi)
54
```

ÖZGEÇMİŞ

Abdullah Taha Arslan, 1981 yılında Kırıkkale’de doğmuştur. Eğitim hayatını sırasıyla Eskişehir Milli Zafer İlkokulu, Eskişehir Fatih Fen Lisesi (1999), İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü’nde (2003) tamamlamıştır. California State University, East Bay’den İşletme Yüksek Lisans derecesi almıştır (2006). Yazılım sektöründe uzun seneler boyunca, yazılım geliştirici, sistem yöneticisi ve proje yöneticisi görevlerinde çalışmıştır. Birden fazla şirketin kurucu ortaklığını yapmıştır. Son olarak, ESOGÜ Teknopark’ta faaliyet gösteren Techy Bilişim Teknolojileri Şirketi’nin kuruculuğunu yapmıştır. Görüntü işleme ve yapay zekâ alanlarında Araştırma-Geliştirme faaliyetleri yürütmekte ve ulusal ve uluslararası destekli projelerin yöneticiliğini yapmaktadır.