

# ÖZİNİTELİK TABANLI İMGE ÇAKIŞTIRMA

Şahin Işık  
Yüksek Lisans Tezi

Bilgisayar Mühendisliği Ana Bilim Dalı

Haziran 2014

## JÜRİ VE ENSTİTÜ ONAYI

**Şahin Işık**'ın "**Öznitelik Tabanlı İmge Çakıştırma**" başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi 16.06.2014 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	<u>Adı Soyadı</u>	<u>İmza</u>
Üye (Tez Danışmanı) :	Yard. Doç. Dr. Kemal ÖZKAN	.....
Üye :	Doç. Dr. Cüneyt AKINLAR	.....
Üye :	Yard. Doç. Dr. Muzaffer DOĞAN	.....
Üye :		.....
Üye :		.....

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun  
..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

## ÖZET

**Yüksek Lisans Tezi**

### ÖZNETELİK TABANLI İMGE ÇAKIŞTIRMA

**Şahin IŞIK**

**Anadolu Üniversitesi**

**Fen Bilimleri Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman: Yard. Doç. Dr. Kemal ÖZKAN**

**2014, 85 Sayfa**

Bu çalışmada görüntü eşleştirme ve çakıştırma için eğrilik tabanlı öznitelik çıkarma yaklaşımı önerilmektedir. Bu amaç doğrultusunda, eğriler arasındaki geometrik uzaklık Hausdorff yöntemi kullanılarak hesaplanmaktadır. Sonra ikinci bir eşleştirme olarak, eğriler üzerinde Harris noktaları bulunup eşleştirilmektedir. En iyi eşleşen eğriler bulunduğunda, referans ve hedef resim arasındaki geometrik dönüşüm RANSAC yöntemi kullanılarak tahmin edilip görüntüler çakıştırılmaktadır. Deneysel sonuçlar elde etmek için Mikolajczyk ve Schmid tarafından önerilen ve en iyi bilinen veritabanı kullanılmaktadır. Sonuçların karşılaştırılması için kesinlik ve duyarlılık kriterleri temel alınarak yapılmaktadır. Önerilen yöntemin sonuçları ile SIFT, SURF, MSER, BRISK, FREAK, ORB, FAST ve BRIEF öznitelik ve betimleyici çıkarıcı yöntemlerin sonuçları ile karşılaştırılmaktadır. Yapılan testlerde elde ettiğimiz sonuçlara göre rotasyon ve zoom, bulanıklık, görüş açısı, ışık şiddeti, JPEG sıkıştırma değişimlerinde ve yüksek gürültü şartlarında diğer yöntemlerin performansında düşüş olduğu görülürken önerilen yöntemin ise performansını koruduğu gözlenmektedir.

**Anahtar Kelimeler:** Öznitelik çıkarma, görüntü eşleştirme, imge çakıştırma, eğri tabanlı imge çakıştırma.

## ABSTRACT

Master of Science Thesis

### FEATURE BASED IMAGE REGISTRATION

Şahin IŞIK

Anadolu University  
Graduate School of Sciences  
Computer Engineering Program

Supervisor: Assist. Prof. Dr. Kemal ÖZKAN

2014, 85 pages

In this study, a curve based feature extraction method is proposed in order to image matching and registration. For this purpose, geometric distance between curves are computed by using Hausdorff distance method. Later, Harris points on curves are determined and matched as a second matching. Once the best matched curves obtained, the geometric transformation between the reference and target images is estimated by using RANSAC method and images are registered. The widely known database which is proposed by Mikolajczyk and Schmid is carried to obtain experimental results. The comparison of results are done based on precision and recall criteria. The results of this study are compared with results of SIFT, SURF, MSER, BRISK, FREAK, ORB, FAST and BRIEF feature and descriptor extraction methods. By looking the results, while a decline is observed on performance of other methods, the proposed method is able to survive its performance in case of rotation and zoom, blurring, viewpoint, illumination, JPEG compression changes and high level noise conditions.

**Keywords:** Feature extraction, image matching, image registration, curve based image registration

## TEŞEKKÜR

Yüksek lisans öğrencisi olduğum günden beri bu tez çalışmasının her aşamasında desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren her açıdan kendime örnek almaya çalıştığım değerli hocam Sayın Kemal Özkan'a teşekkürlerimi sunuyorum.

Yine eğitim-öğretim hayatım boyunca üzerimde emeği olan hocalarıma en içten teşekkürlerimi sunarım.

Özellikle çok değerli arkadaşlarım olan Emrah Göksülük'e, Muhammed Yüce'ye, Mehmet Özmen'e, Sinan Sarıyıldız'a ve diğer tüm arkadaşlarım ve dostlarıma gönülden teşekkür ederim.

Ayrıca beni yetiştirirken maddi ve manevi olarak benden desteklerini esirgemeyen ve bugünlere ulaşmamı sağlayan sevgili ailemin tüm fertlerine bütün kalbimle teşekkür ederim.

Şahin Işık

Haziran 2014

## İÇİNDEKİLER

<b>ÖZET</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>TEŞEKKÜR</b>	<b>iii</b>
<b>ŞEKİLLER DİZİNİ</b>	<b>vi</b>
<b>ÇİZELGELER DİZİNİ</b>	<b>viii</b>
<b>SİMGELER ve KISALTMALAR DİZİNİ</b>	<b>ix</b>
<b>1 GİRİŞ</b>	<b>1</b>
1.1 Literatür Taraması.....	4
1.2 Tez Organizasyonu .....	5
<b>2 ÖZİNİTELİK ÇIKARMA YÖNTEMLERİ</b>	<b>8</b>
2.1 Scale Invariant Feature Transform (SIFT).....	8
2.1.1 Ölçekleme Uzayında Tepe Noktaları Bulma.....	9
2.1.2 Anahtar Noktaları Lokalize Etme.....	10
2.1.3 Betimleyicilere Yön Atamak ve Yerel Betimleyiciler Çıkarmak...11	
2.1.4 Seçilen Öz nitelikleri Eşleştirme .....	13
2.2 PCA-SIFT .....	13
2.3 Speeded Up Robust Features (SURF) .....	13
2.3.1 Anahtar Noktaları Bulma .....	14
2.3.2 Dönmeden Etkilenmeyen Anahtar Noktalarını Bulma.....	16
2.3.3 Öz nitelikleri Eşleştirme .....	17
2.4 Features From Accelerated Segment Test (FAST).....	17
2.5 Binary Robust Invariant Scalable Keypoints (BRISK) .....	18
2.6 Maximally Stable Extremal Regions (MSER).....	19
2.7 Binary Robust Independent Elementary Features (BRIEF) .....	19
2.8 Oriented FAST and Rotated BRIEF (ORB) .....	20
2.9 Fast Retina Keypoint (FREAK).....	20
<b>3 ÖNERİLEN YÖNTEM</b>	<b>22</b>
3.1 Ön işlem.....	24
3.1.1 Sanatsal Kenar ve Köşe Koruyarak Yumuşatma.....	25

3.1.2	Canny Kenar Tanıma Yöntemi.....	27
3.1.3	Ortak Vektör Tabanlı Kenar Tanıma Yöntemi (CVAED) .....	31
3.1.4	Ortak Vektör Tabanlı Çok Ölçekli-Uzmanlı Kenar Tanıma .....	37
3.1.5	Gürültülü Resimlerde Kenar Çıkarma: LPGPCA-ED.....	43
3.2	Öznitelik Çıkarma.....	49
3.3	Benzer Eğrilerin Bulunması.....	50
3.3.1	Hausdorff Uzaklığı .....	52
3.4	Öznitelik Eşleştirme.....	53
3.5	Geometrik Dönüşümün Hesaplanması ve Görüntü Çakıştırma.....	54
3.5.1	İlgin Dönüşüm .....	55
3.5.2	RANSAC.....	56
3.5.3	Görüntü Çakıştırma .....	57

#### **4 HESAPSAL SONUÇLAR** **59**

4.1	Veritabanı.....	59
4.2	Karşılaştırma Metriği .....	60
4.3	Rotasyon (Rotation) ve Zum (Zoom) Değişimi Etkileri.....	62
4.4	Bulanıklık (Blur) Değişimi Etkileri .....	63
4.5	Görüş Açısı (Viewpoint) Değişimi Etkileri .....	66
4.6	Işık Şiddeti (Illumination) Değişimi Etkileri .....	68
4.7	JPEG Sıkıştırma (JPEG Compression) Değişimi Etkileri .....	69
4.8	Gürültülü (Noise) Değişimi Etkileri .....	72
4.9	Çalışma Sürelerini Karşılaştırma .....	76

#### **5 SONUÇLAR ve ÖNERİLER** **78**

#### **KAYNAKÇA** **80**

## ŞEKİLLER DİZİNİ

Şekil 1.1	Görüntü çakıştırma .....	2
Şekil 1.2	Bir devreye ait iki farklı görüntünün çakıştırılması.....	3
Şekil 1.3	Görüntü çakıştırma örnek .....	4
Şekil 2.1.1.1	İki ölçek uzay arasındaki farkların (DoG) bulunması .....	9
Şekil 2.1.1.2	Üç ölçek kullanılarak en büyük pik noktası bulma .....	10
Şekil 2.1.2.1	Doğru tepe noktaları ile bulunan tepe noktaları .....	10
Şekil 2.1.3.1	Yerel betimleyici çıkarma .....	12
Şekil 2.3.1.1	Türevlerin yakınsamasına karşılık Box filtreler .....	14
Şekil 2.3.1.2	İntegral image .....	15
Şekil 2.3.1.3	Box filtreler ile oluşturulan piramit.....	15
Şekil 2.3.1.4	Ölçeklerden elde edilen öznelik sayısı .....	15
Şekil 2.3.2.1	SURF yönteminde betimleyici elde etme.....	16
Şekil 2.3.2.2	SURF öznelik çıkarma .....	17
Şekil 2.4.1	(a) A bir ilgi noktası (b) bir etrafından vektör oluşturma.....	18
Şekil 3.1	Genel olarak görüntü çakıştırma işlemleri.....	22
Şekil 3.2	Önerilen Yöntem.....	23
Şekil 3.1.1.1	Artistik yönteminin davranışlarının sergilenmesi .....	25
Şekil 3.1.1.2	Sanatsal köşe ve kenar koruyarak resmi yumuşatma .....	26
Şekil 3.1.2.1	Canny yüksek eşik değeri çıktıları .....	29
Şekil 3.1.2.2	Canny alçak eşik değeri çıktıları .....	29
Şekil 3.1.2.3	Canny yöntemi ile elde edilen kenar çıktıları.....	30
Şekil 3.1.3.1	Ortak vektör yöntemi ile elde edilen ortak ve fark resimleri.....	34
Şekil 3.1.3.2	Kenarları bağlama yönleri .....	35
Şekil 3.1.3.3	OVY tabanlı kenar çıkarma çıktıları .....	36
Şekil 3.1.4.1	Çok ölçekli çok uzmanlı kenar tanıma modeli .....	38
Şekil 3.1.4.2	Önerilen yöntem ile elde edilen 9 farklı gradyen haritası .....	38
Şekil 3.1.4.3	İki vektöre ait Farksızlık ve Farklılık altuzayları .....	39
Şekil 3.1.4.4	Tek bir gradyen haritası.....	40
Şekil 3.1.4.5	Gürültü resimlerde çok-ölçekli çok-uzmanlı kenar tanıma.....	41
Şekil 3.1.4.6	Gürültüsüz resimlerde önerilen yöntemin performansı.....	42
Şekil 3.1.5.1	LPGPCA-ED yöntemi .....	44

Şekil 3.1.5.2	Gürültülü resimde sınıf seçmek.....	45
Şekil 3.1.5.3	LPGPCA ve Canny yöntemi ile elde edilen kenar segmentleri ..	49
Şekil 3.3.1	Hausdorff uzaklığı ve ikinci eşleştirme ile benzeyen eğriler .....	51
Şekil 3.3.1.1	Hausdorff uzaklık yönteminin çalışma prensibi.....	52
Şekil 3.4.1	İki farklı eğri üzerinde eşleşen öznelikler.....	54
Şekil 3.5.2.1	RANSAC çalışma prensibi.....	56
Şekil 3.5.3.1	Görüntü çakıştırma genel görünüm .....	57
Şekil 3.5.3.2	Döndürülmüş iki ev resminin çakıştırılması.....	58
Şekil 4.1.1	Deney resimleri .....	60
Şekil 4.3.1	Bark ve Baot resimleri üzerindeki sonuçlar .....	62
Şekil 4.3.2	Zum ve Rotasyon için Precision ve recall değerleri.....	63
Şekil 4.4.1	Bikes ve Trees resimleri üzerindeki sonuçlar .....	64
Şekil 4.4.2	Bulanıklık için precision ve recall değerleri .....	65
Şekil 4.5.1	Graf ve Wall resimleri üzerindeki sonuçlar .....	66
Şekil 4.5.2	Görüş açısı değişimi için precision ve recall değerleri .....	67
Şekil 4.6.1	Leuven resimleri üzerindeki sonuçlar .....	69
Şekil 4.6.2	Parlaklık değişimi için precision ve recall değerleri.....	69
Şekil 4.7.1	Ubc resimleri üzerindeki sonuçlar .....	71
Şekil 4.7.2	JPEG sıkıştırma değişimi için precision ve recall değerleri.....	71
Şekil 4.8.1	SURF, MSER, FAST, HARRIS ve BRISK sonuçlar .....	73
Şekil 4.8.2	Birinci resim üzerinde MSME-CVAED kenar çıkarıcı sonuçları..	73
Şekil 4.8.3	Birinci resim üzerinde LPGPCA-ED kenar çıkarıcı sonuçları .....	74
Şekil 4.8.4	Birinci resim üzerinde CVAED kenar çıkarıcı sonuçları.....	74
Şekil 4.8.5	Birinci resim üzerinde Canny kenar çıkarıcı sonuçları .....	75
Şekil 4.8.6	İkinci resim üzerinde MSME-CVAED kenar çıkarıcı sonuçları ...	75
Şekil 4.8.7	İkinci resim üzerinde LPGPCA-ED kenar çıkarıcı sonuçları .....	75
Şekil 4.8.8	İkinci resim üzerinde CVAED kenar çıkarıcı sonuçları .....	76
Şekil 4.8.9	İkinci resim üzerinde Canny kenar çıkarıcı sonuçları.....	76

## ÇİZELGELER DİZİNİ

Çizelge 4.9.1 Zamansal sonuçlar .....	77
---------------------------------------	----

## SİMGELER ve KISALTMALAR DİZİNİ

SIFT	: Scale Invariant Feature Transform
SURF	: Speeded Up Robust Features
ORB	: Oriented FAST and Rotated BRIEF
MSER	: Maximally Stable Extremal Regions
BRISK	: Binary Robust Invariant Scalable Keypoints
OVY	: Ortak Vektör Yöntemi
FAST	: Features from Accelerated Segment Test
BRIEF	: Binary Robust Independent Elementary Features
FREAK	: Fast Retina Keypoint
PCA	: Principal Component Analysis
DoG	: Difference of Gaussian
SSD	: Sum of Squared Distances
SUSAN	: Smallest Univalued Segment Assimilating Nucleus
CENSURE	: Center Surround Extremas for Realtime Feature Detection and Matching
ROC	: Receiver Operating Characteristic
TBA	: Temel Bileşen Analizi
JPEG	: Joint Photographic Experts Group
RANSAC	: Random Sample Consensus
CVAED	: Common Vector Based Edge Detection
LMMSE	: Linear Minimum Mean Square-Error Estimation
MSME-CVAED	: CVA Based Multi Segment and Multi Expert Edge Detection
LPGPCA	: Principle Component Analysis on Local Pixel Grouping
LPGPCA-ED	: PCA on Local Pixel Grouping Based Edge Detection

## 1 GİRİŞ

Görüntü işlemenin alt dallarından biri olan görüntü çakıştırma birçok probleme çözüm getirmesi açısından son derece hayati bir konudur. Görüntü çakıştırma, genel olarak, aynı sahneye ait farklı açılardan alınmış iki görüntünün ya da ardışık iki video karesinin üzerinde aynı nesnelere ait piksellerin, geometrik eşleşmeleri yardımıyla, üst üste bindirilmesi olarak bilinir. Temelde görüntü iyileştirmenin bir alt dalı olan görüntü çakıştırma birçok alanda yaygın olarak kullanılmakta ve sınıflandırma algoritmaların yapacağı işi kısa zaman dilinde çözebilmektedir.

Görüntü çakıştırmanın uygulandığı ve kullanıldığı birçok alan bulunmaktadır. En basit görüntü çakıştırma işlemi, iki görüntü arasında sadece öteleme farkı olması durumunda bağlı yer değiştirmenin bulunmasıdır. Bunun için görüntülerin katı olduğunu (bir bütün halinde hareket ettiğini) ve aralarında sadece doğrusal ötelenme olduğu varsayılır. Bu tür görüntü çakıştırma işlemleri, bir füzenin hareket halindeyken elde ettiği görüntü setini bir referans imge (odaklanılan hedef) ile karşılaştırılması şeklinde veya çekilen ardışık iki görüntü arasındaki farklara bakarak belli bir görüntüleme sisteminin hızının ölçülmesi gibi pek çok uygulamanın konusu olmuştur [1]. Bir manzaradaki değişikliklerin sezilmesi veya endüstriyel ürünlerin otomatik denetlenmesi gibi, aritmetik çıkartma işlemine dayalı görüntü işleme ve bilgisayarla görme uygulamaları da karşılaştırılan görüntülerin ancak başarıyla çakıştırılmasından sonra gerçekleştirilebilir.

Genel olarak geliştirilen bir görüntü çakıştırma yöntemi aşağıdaki amaçlar için kullanılır:

Medikal görüntülerin çakıştırılması [2-4] ile hastalıklı doku ve hücrelere ait anatomik yapının şeklini kesinleştirebilir ve fonksiyonu hakkında yorum yapılabilir.

Demir yolları üzerinde hangi cisimlerin olduğu ve rayların kontrolü gibi demir yollarının güvenliğini ilgilendiren konularda görüntü çakıştırma yöntemi kullanılabilir.

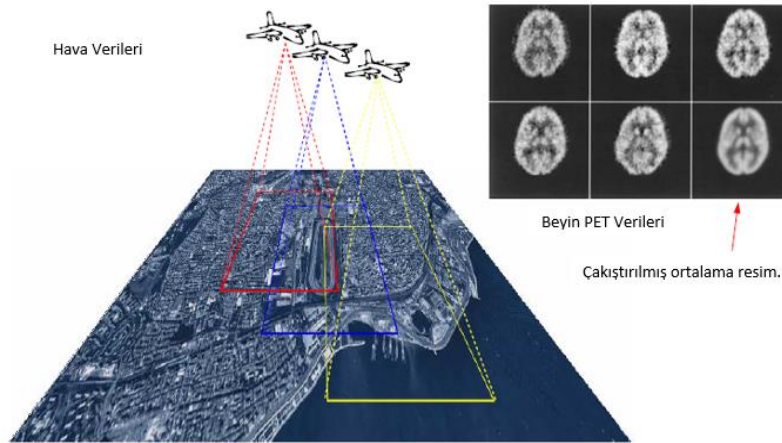
Askeriye ait uçak, helikopter ve insansız hava araçları, hedefi takip etmek [5] ve kendini korumak zorundadır. Çok iyi bir görüntü çakıştırma ve tanıma yöntemi ile hedefin ne olduğu belirlenebilir. İlaveten, askeriye ait araçların güvenli bir

uçuş yapması için, vurulmayacak bir yükseklikten uçmaları gerekir. Bunun için yine aşağıda var olan tehditlerin belirlenmesinde fayda vardır. Görüntü çakıştırma bunun içinde kullanılabilir. Yine askeriye ait otomatik olarak bir yerdeki hareketliliği ve değişikliği yakalamak [6, 7] için görüntüler çakıştırılabilir.

Elimizde bir doğal veya turistik yere ait görüntü dizisi varsa, çok iyi bir çakıştırma yöntemi ile bu görüntülerden panoramik görüntüler [8-10] elde edebilir. Bunun için Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented Fast and Rotated Brief (ORB), Maximally Stable Extremal Regions (MSER) ve Binary Robust Invariant Scalable Keypoints (BRISK) vb. öznelik seçme yöntemleri kullanılır. Bu metotlardan dönen öznelikler kullanılıp görüntü çakıştırılır.

Yine stereo görüntü veya görüntünün netliği ve kalitesinin mükemmel sayılacak bir düzeye erişmesi için (3-Boyutlu görüntü oluşturmak için kullanılan yöntem) için görüntülerin çakıştırılmasına [11, 12] ihtiyaç vardır.

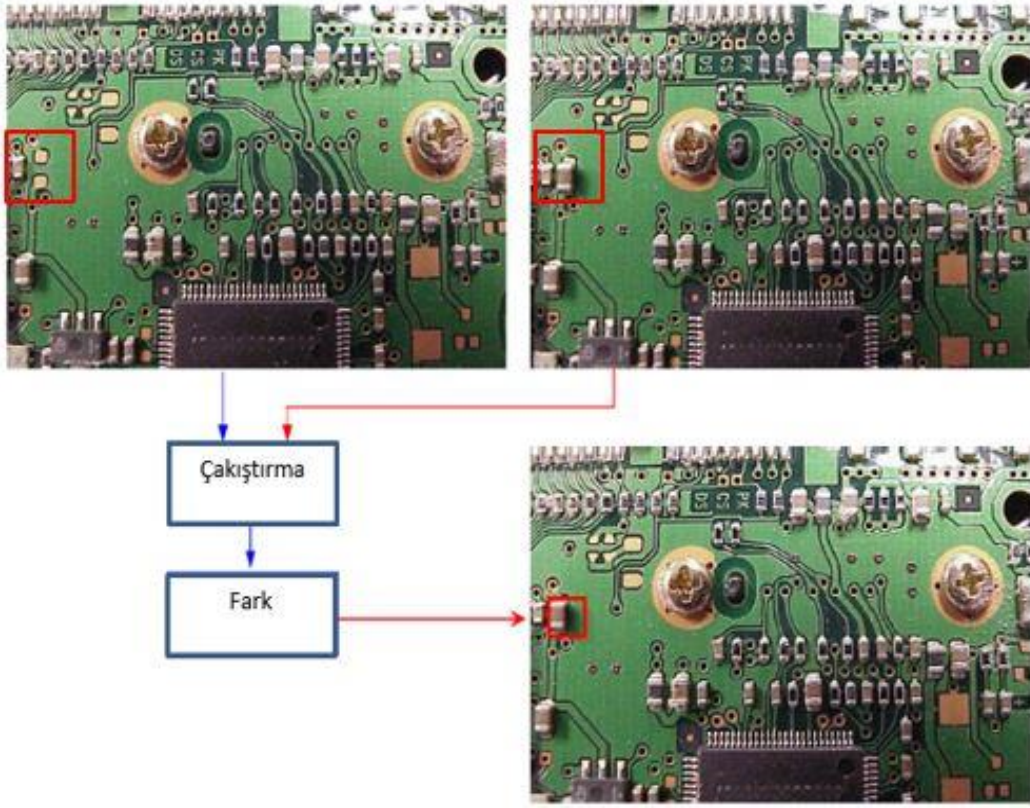
Farklı uydulardan, farklı bakış açılarından veya farklı zamanlarda elde edilen görüntülerin bir araya getirilmesi bazı çalışmalar için çok büyük öneme sahiptir. Bu bağlamda, uyduya ait görüntülerin birbiri ile çakıştırılarak [13-15] iyileştirilmesi hedeflenmektedir. Böylece arazi planlaması, kadastro oluşturulması ve arazi izlenmesi gibi işlemler yapılmaktadır. Yine uydu görüntülerden gök cisimlerinin tanınması için görüntüler çakıştırılmaktadır.



Şekil 1.1 Görüntü çakıştırma.

İmge çakıştırmanın kullanım alanlarından biri olan düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntü elde etmeye örnek Şekil 1.1'de [16]

verilmiştir. Bu örneğe bakıldığında uçağın iki farklı açıdan çektiği resimlerin gösterimi yapılmıştır. Uçak bu şekilde resimleri çektikten sonra yüksek çözünürlük elde etme algoritması ile eldeki görüntülerden zengin içeriği sahip tek bir görüntü elde edilebilir. Yüksek çözünürlük elde etme algoritmalarının ilk adımı imgelerin çakıştırılmasıdır. Yine aynı konuya benzer başka bir örnek de imgedeki gürültülerin giderilmesidir. Elimizde olan gürültü imgelerin ortalaması alınarak gürültü giderilebilmekte ancak elde edilen sonuç imgesi bulanık olmaktadır. Eğer imgeler çakıştırma işlemi gerçekleştirilip ortalaması alınırsa bulanıklık etkisi Şekil 1.1'de görüldüğü üzere azalmaktadır.



**Şekil 1.2** Bir devreye ait iki farklı görüntünün çakıştırılması

Şekil 1.2'de [16] görüldüğü gibi, iki görüntü arasında bazı farklar bulunmaktadır. Örneğin sağ tarafta bulunan resimde bir devre elemanın olmadığını veya kaybolduğunu net olarak görülmektedir. Normalde eğer elimizde birçok resim varsa, insanın bunu göz ile tek tek inceleyip böyle bir detayın bulması günler alabilir. İşte görüntü çakıştırma bu tür sorunlar için devreye girmektedir. Görüntü çakıştırma kullanılarak iki resim çakıştırılıp fark alma işlemine tabi tutulursa eksik olan malzeme kolayca tespit edilebilmektedir.

Şekil 1.3’de görüntü çakıştırma örnek [17] verilmiştir. Verilen referansta farklı örnekler mevcuttur.



(Referans Resim)

(Hedef Resim)



(Çakıştırma)

Şekil 1.3 Görüntü çakıştırma örnek

### 1.1 Literatür Taraması

Son 20 yılda görüntü çakıştırma üzerinde yapılan makale ve bildiri çalışmaları dünya genelinde oldukça yaygındır. Bu tezdeki amacımız önerilmiş bütün çalışmaları detaylı bir şekilde anlatmak değil, sadece bu alandaki en önemli ve gelişmekte olan konular hakkında bir bilgi vermektir. Görüntü çakıştırma üzerine geniş kapsamlı bir araştırma 1992 yılında Brown [18] tarafından yapılmıştır. Brown yapmış bu olduğu çalışmada, görüntü çakıştırma yöntemlerinin karakteristik özelliklerine göre üç kategoriye ayırmıştır; öznitelik uzayı kullananlar, benzerlik ölçütü kullananlar ve araştırma uzayı ve stratejisi kullananlar. Yine Brown kendi çalışmasında, çakıştırma için önerilen yöntemleri dört gruba ayırmıştır; ardışık-

ilişkili yöntemler, fourier tabanlı yöntemler, nokta eşleştirme ve esnek model tabanlı eşleştirme tabanlı yöntemler.

Brown'dan sonra, J. B. Antoine Maintz medikal görüntü çakıştırma üzerine 1998 de yaptığı kapsamlı bir araştırmada [19], Elsen ve çalışma arkadaşlarının 1993'de önerdiği kıstasları kullanarak [20], görüntü çakıştırma için kullanılan yöntemleri dokuz gruba ayırmıştır; boyutluluk, çakıştırma temelinin niteliği, dönüşümün niteliği, dönüşümün alanı, etkileşim, optimizasyon prosedürü, içerdiği yaklaşımlar, konu ve amaç. Bunlardan sonra, Barbara Zitova ve Jan Flusser yaptıkları geniş bir çalışmada [21] görüntü çakıştırma yöntemlerini alan ve öznitelik tabanlı olmak üzere iki gruba ayırmıştır.

Bir başka çalışmada [22], geometrik dönüşüme tabanlı yöntemler, nokta tabanlı yöntemler, yüzey tabanlı yöntemler ve koyuluk tabanlı yöntemler olmak üzere görüntü çakıştırma yöntemlerini dört farklı gruba ayırarak incelenmiştir. Yine Somaraju Boda görüntü çakıştırma üzerine yaptığı yüksek lisans tezinde [23] yedi farklı sınıf şeklinde görüntü çakıştırma yöntemlerini ele almıştır. Bu sınıflar; alan ve öznitelik tabanlı, dönüşüm modeli kullananlar, araştırma tabanlı ve direkt yöntemler, uzaysal alan tabanlı yöntemler, frekans alan tabanlı yöntemler, tekli ve çoklu yaklaşımlar ve resim benzerlik tabanlı yöntemlerdir.

## 1.2 Tez Organizasyonu

Bu tez çalışmasında genel olarak öznitelik tabanlı imge çakıştırma üzerinde çalışılmıştır. Bu kapsamda eğri tabanlı yeni bir görüntü çakıştırma yöntemi önererek literatürde var olan yöntemler ile karşılaştırmalar sunulmuştur. Karşılaştırma kriteri olarak kesinlik ve duyarlılık ölçütleri kullanılmıştır. Önerilen tez çalışması genel olarak aşağıdaki aşamalardan oluşmaktadır:

- İlk olarak deney imgelerinden eğrileri çıkartılır.
- Bu eğrilerin yatay ve dikey gradyen değerleri hesaplanır ve temel yönleri TBA ile bulunulur. Sonra gradyenler temel yön üzerine map edilerek normalize edilir.
- İki eğriye ait normalize edilmiş katsayılara Hausdorff yöntemi uygulanarak aralarındaki uzaklık bulunulur.

- İkinci bir eşleştirme olarak gradyen haritası çıkarılmış imgedeki eğriler üzerinden Harris noktaları [24] elde edilir. Burada Harris noktaları yerine SURF veya SIFT noktaları kullanılabilir.
- İkinci eşleştirme ile Harris noktalarının eşleşme sayısına göre bir eğriye benzeyen diğer eğrilerin sayısı azaltılır.
- Sonra en küçük Hausdorff uzaklığı temel alınarak en iyi eşleşen eğriler belirlenir.
- En iyi eşleşen eğriler üzerindeki Harris öznitelikleri kullanılarak iki imge arasındaki dönüşüm matrisi bulunur.
- En son olarak dönüşüm matrisi kullanılarak dönen imge yeniden döndürülür ve iki imge Matlab yardımıyla çakıştırılır.

Bu tezin düzeni şu şekildedir:

Bölüm 1’de genel olarak imge çakıştırma ve literatürde olan çalışmalardan bahsedilmiştir. Yine görüntü çakıştırmaya motivasyon olması bakımından ve örnekler verilmiştir.

Bölüm 2’de öznitelik tabanlı imge eşleştirme ve çakıştırmanın ön adımlarından olan son zamanlarda en çok kullanılan öznitelik çıkarma yöntemleri hakkında genel bilgi verilmiştir. Özellikle içlerinden en popüler olan SIFT, Principal Component Analysis SIFT (PCA-SIFT), SURF, Features From Accelerated Segment Test (FAST), BRISK, MSER, Binary Robust Independent Elementary Features (BRIEF), ORB ve Fast Retina Keypoint (FREAK) yöntemleri genel olarak açıklandı.

Bölüm 3’te ise önerilen yöntem, genel olarak bir görüntü eşleştirme ve çakıştırma yönteminin takip etmesi gereken aşamalar kullanılarak anlatılmıştır. Bu bölümde ön işlem olarak imgenin kenar ve köşelerinin korunarak yumuşatılması ve kenar segmentlerinin kenar tanıma yöntemleri kullanarak nasıl elde edildiği örnek verilerek sunulmuştur. Yeni olarak önerdiğimiz kenar tanıma yöntemleri sergilendikten sonra eğri tabanlı öznitelik nedir ve nasıl yapılır aşamalar ile açıklanmıştır. Eğrileri birbiriyle eşleştirmek için kullandığımız Hausdorff yöntemi ve eğrileri eşleştirirken hangi adımları takip ettiğimiz açıklanmıştır. Yine ve bir ön gösterge olması için örnekler verilerek gösterilmiştir. Sonra iki eğri üzerindeki kullanılarak imgeler arasındaki ilgin dönüşümün nasıl hesaplandığı anlatılmıştır.

Dönüşümün hesaplanmasında kullanılan parametreler, fonksiyonlar ve metrikler bölüm sonunda paylaşılmıştır.

Bölüm 4'te önerdiğimiz yöntem ile elde edilen deneysel sonuçlar literatürde en popüler olan öznitelik çıkarma yöntemlerinin sonuçları ile karşılaştırılmıştır. Sonuçlar hem sayısal hem de görsel şekilde sergilenmiştir. Karşılaştırma kriterleri olarak kesinlik ve duyarlılık metriklerinin nasıl hesaplandığı ve hangi yazılım dilleri ve kullanılan veritabanı hakkında bilgi verilmiştir.

Bölüm 5'te ise bu yönteminin literatürdeki diğer çalışmalardan üstünlüğü, diğer çalışmaların üstünlükleri ve gelecekte bu çalışmanın devamı olarak ne yapabileceğimiz üzerinde yorumlar yapılmıştır. Yine farklı imge deformasyonlarda hangi yöntemleri kullanılması gerektiği üzerinde açıklamalar yapılmıştır.

## 2 ÖZİNİTELİK ÇIKARMA YÖNTEMLERİ

Artan teknolojiye paralel olarak, akıllı cihazların sayısı gittikçe artmaktadır. Önceden sadece 4 işlem yapabilen bilgisayarlar, günümüzde insanların biyolojik özelliklerine örneğin sesine, parmak izine ve yüzüne bakarak onun kim olduğunu bize haber vermektedir. Yine internetin yaygın olarak kullanıldığı bu çağda kaçak araç, cihaz ve diğer çalıntı eşyaların bulunması, geliştirilen yeni program ve algoritmalarla sadece bir kaç dakika tutmaktadır. Şüphesiz ki, bu programların gelişmesinde öznitelik çıkarma ve belirleme yapan algoritmaların katkısı oldukça fazladır. Özellikle resim verilerinde, ölçeklenme, dönme, aydınlanma, gürültü, sıkıştırma ve bulanıklaşma gibi bozulmalara maruz kalmış verileri eşleştirmek için 21. Yy'a kadar birçok güçlü öznitelik çıkarma yöntemleri önerilmiştir. Literatürde en yağın olarak kullanılan yöntemler aşağıda özetlenmiştir.

### 2.1 Scale Invariant Feature Transform (SIFT)

SIFT öznitelik metodunu, literatürdeki var olan öznitelik çıkarma metotlardan farklı ve üstün kılan en önemli özellik; farklı ölçek ve dönüşümlerde, değişen parlaklıklarda, görünüşteki küçük açı değişimlerinde, gürültü eklenmiş, karışık ve yıpranmış resimlerde değişmeyen öznitelikleri bulmayı amaç edinmesidir. Lowe tarafından 2004'te önerilmiştir [25]. SIFT metodunda 4 aşama vardır. Bu aşamaların hepsi aşağıda daha detaylı olarak anlatılmıştır.

- Ölçekleme uzayında tepe noktaları bulma (scale-space extrema detection).
- Bir önceki aşamadan elde edilen tepe noktaların konumlarının doğruluğunu artırmak (keypoint localization).
- En son aşama olarak betimleyicilere yön atamak ve
- Yönleri bulunan bu betimleyiciler içinden farklı ölçek ve dönüşümlerde yönü değişmeyen olanları seçmek (generation of the keypoint descriptors).

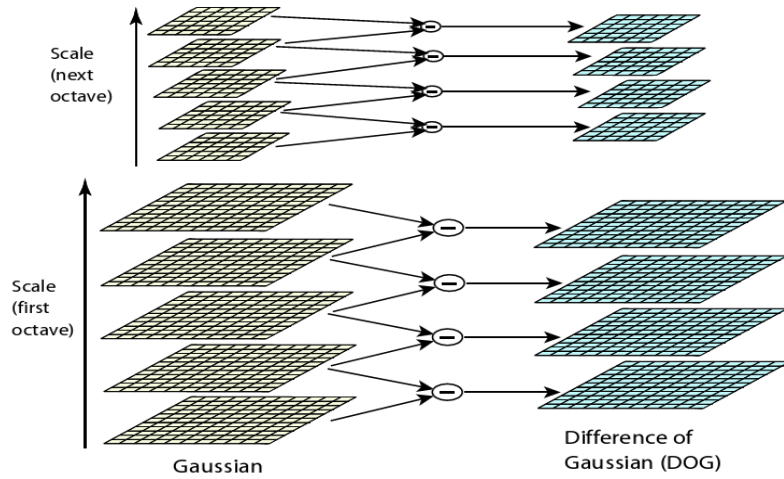
SIFT metodu üzerinde yapılan değişiklikler ile PCA-SIFT, SURF ve GLOH gibi metotlar türemiştir. Bu çalışmada SIFT içim verilen imgeler [26]'den alınmıştır.

### 2.1.1 Ölçekleme Uzayında Tepe Noktaları Bulma

Tepe noktalarını tespit etmek için ilk olarak bir resme ait tekrar eden farklı görüntü koşullardaki konumları ve ölçekleri belirlenmektedir. Farklı ölçeklerde değişmeyen konumları bulmak için ölçek uzayı diğer adıyla sürekli ölçek fonksiyonu kullanılır. Bu fonksiyon ile farklı ölçeklerde değişmeyen öznelikler tespit edilir. Verilen  $I_b(x, y)$  resmini bir Gauss ölçek değeri konvolüsyon yapılarak ölçek uzayı fonksiyonu  $L(x, y, \sigma)$  elde edilir.

$$L(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) * I_b(x, y) \quad (2.1.1.1)$$

Burada \* konvolüsyon operatörü, x ve y bir pikselin konumu ve  $\sigma = \sqrt{2}$  ise Gauss ölçeğinde kullanılan standart sapma değeridir.



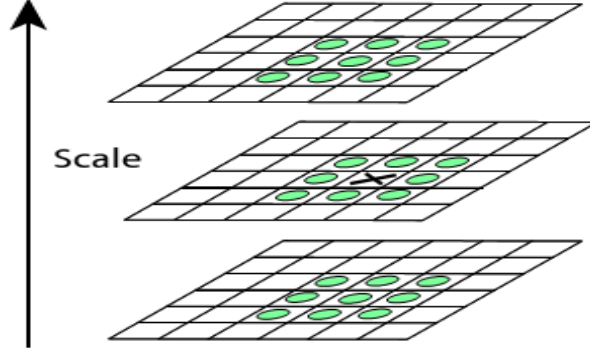
Şekil 2.1.1.1 İki ölçek uzayı arasındaki farkların (DoG) bulunması

Değişmeyen öznelikleri hızlı ve verimli şekilde bulmak için Lowe tepe ölçek uzayı yöntemini önermiştir. Bu yöntemle göre; bir resme ait birbirine yakın olacak şekilde ölçeklendirilmiş iki ölçek uzayının aralarındaki fark (DoG) bulunarak tepe ölçek uzayı elde edilir.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.1.1.2)$$

$D(x, y, \sigma)$ 'teki tepe noktasını bulmak için önce işlenen merkez pikselin değeri etrafındaki 8 tane piksel ile sonra üstteki ölçek uzaydaki 9 piksel değeri ile ve alttaki ölçek uzaylardaki 9 piksel değeri ile karşılaştırılır. Böylece toplam 26

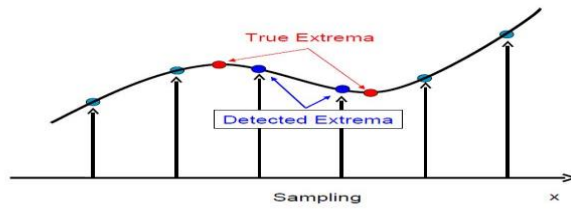
piksel değeri karşılaştırılarak en büyük veya en küçük değerini veren nokta aday nokta seçilir.



Şekil 2.1.1.2 Üç ölçek kullanılarak en büyük pik noktası bulma

### 2.1.2 Anahtar Noktaları Lokalize Etme

İkinci aşama aday noktaların konumlarının doğru tespit edilmelidir. Örneğin elimizde 233x189 boyutlarında bir resimden 832 tane DoG tepe noktası (Extrema) elde edilebilmektedir. Bu noktaların hepsinin konumunun Şekil 2.1.2.1’de verildiği gibi doğru tespit edilmemiş olabilir. SIFT yönteminin bulduğu tepe noktaları (Detected Extrema) ile doğru tepe noktaları (True Extrema) arasında bir kaymanın olduğu görülmektedir.



Şekil 2.1.2.1 Doğru tepe noktaları ile bulunan tepe noktaları

Bu problemi ortadan kaldırmak için yani tepe noktalarını kaydırmak için seçilen noktalara Şekil 2.1.2.1’de gösterildiği gibi [27] Taylor serisi genişlemesi (Taylor Series Expansion) uygulanır. Bu amaç için 3D eğri uydurulur.

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (2.1.2.1)$$

Eşitlik Şekil 2.1.2.1’de gösterildiği gibi DoG fonksiyonun türevini alıp sıfıra eşitlersek aşağıdaki gibi tepe noktalarının gerçek konumlarını tespit edebiliriz.

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (2.1.2.2)$$

Yine ikinci aşamada düşük kontrasta sahip pikseler belli bir eşik değerine tabi tutularak elenir. Örneğin; [0 1] aralığında değerlere sahip noktalar arasından 0.3 değerinden küçük olanlar elenebilir. Aynı şekilde kenar üzerinde olmayan noktaları temel eğri analizi yöntemi (principal curvature analysis) ile ihmal edilmektedir. Bunun için Harris metodundaki düşünce kullanılır yani kenar üzerinde dağılım büyük özdeğerler (eigenvalues) verirken kenar dışında küçük özdeğerler (eigenvalues) vermelidir. Bunun için aşağıdaki kurala göre belli bir eşik değerinden küçük olanlar elenir.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.1.2.3)$$

$$Tr(H) = D_{xx} + D_{yy} \quad (2.1.2.4)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 \quad (2.1.2.5)$$

$$\frac{Tr(H)^2}{Det(H)} = \frac{(r+1)^2}{r} \quad (2.1.2.6)$$

Yukarda Eşitlik 2.1.2.2 ile Eşitlik 2.1.2.6 arasında SIFT yönteminin düşük kontrasta sahip pikselleri eleme stratejisi göstermektedir. Hessian matrisinde (H) verilen  $D_{xx}$  en büyük özdeğeri gösterirken,  $D_{yy}$  en küçük özdeğeri göstermektedir. Burada bir eşik değeri belirlenir ve  $r$  değeri eşik değerinden küçük olan noktalar elenmektedir. SIFT yönteminde genel olarak  $r$  değeri 10 olarak alınmaktadır.

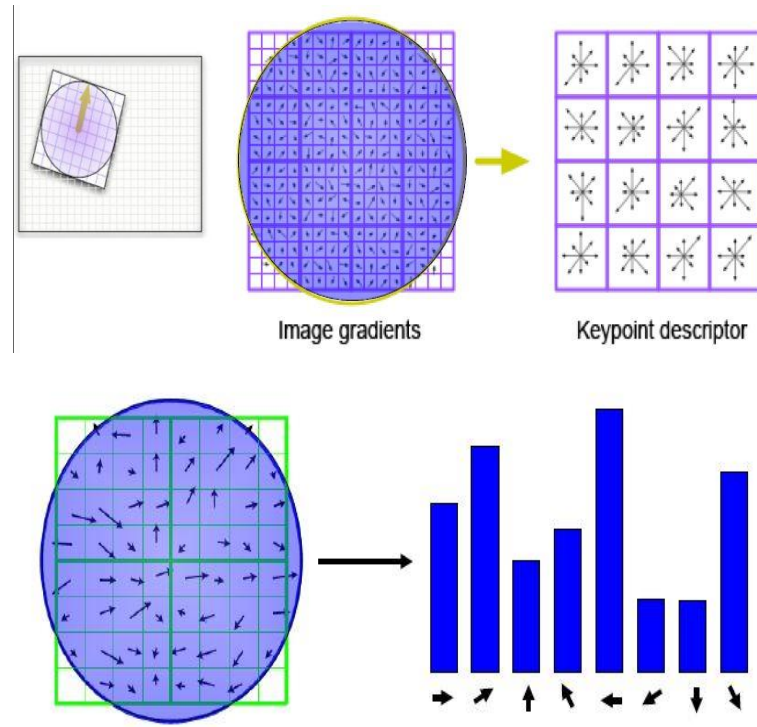
### 2.1.3 Betimleyicilere Yön Atamak ve Yerel Betimleyiciler Çıkarmak

Üçüncü aşamada aday nokta etrafındaki noktaların yönleri belirlenir. Aşağıda görüldüğü gibi  $m$  gradyen değerini belirlerken,  $\theta$  bir noktanın açısını vermektedir. Aşağıdaki eşitlikler verildiği gibi  $m$  değeri hesaplanırken normalize edilmiş yatay ve dikey türevlerinin karelerinin toplamının karekökü alınır. Sonra yatayda alınan türevin dikeyde alınan türeve oranı bize bütün noktaların açısını vermiştir.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.1.3.1)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (2.1.3.2)$$

Bu aşamada 360 derece 36 parçaya bölünerek, 12 parçaya (her biri 10°) sahip bir yön histogramı elde edilir. Burada bir nokta histograma eklenirken yön büyüklüğü ile standart sapması 1,5 ilgi ölçüğü olan Gauss pencere fonksiyonundaki katsayı değerlerinin çarpımının değeri temel alınır.



Şekil 2.1.3.1 Yerel betimleyici çıkarma

Son aşamada ise yerel betimleyiciler elde edilir. Şekil 2.1.3.1’de görüldüğü gibi gradyen resmi 4x4 lük karelere bölünür. Her kareye 8 dilimlik histograma uygulanarak 4x4x8=128 tane betimleyici elde edilir. Yine histograma eklenecek noktanın yönelim büyüklüğü değeri ile bu değerin ayrıca varyansı 0,5 olan Gauss pencere fonksiyonu ile çarpılmasıyla histograma oy verilir. Burada görüldüğü gibi histogramın dilim sayısı artarsa betimleyici sayısı da doğru orantılı olarak artmaktadır. Yani 8 dilim yerine 12 dilimlik histograma kullanılırsa 4x4x12=192 tane betimleyici elde edilir. Son olarak elde edilen betimleyiciler L-2 normuna göre normalize edilir. Böylece ışık şiddetindeki değişimlerin resim üzerindeki etkisi azaltılır.

#### 2.1.4 Seçilen Öznitelikleri Eşleştirme

SIFT öznitelik metodunda [26], öznitelikler eşleştirmek için uzaklıkların karelerinin toplamı (SSD) yöntemi kullanılır. Buna göre  $f_1$  ve  $f_2$  iki tane anahtar nokta olsun, bu noktaların etrafındaki betimleyiciler  $i$  ile isimlendirirse, SSD değeri aşağıdaki gibi hesaplanır.

$$SSD(f_1, f_2) = \sum_{i=1}^N (f_{1i} - f_{2i})^2 \quad (2.1.5.1)$$

Ama SSD değerini bulmak yetmez, bunun için bir eşik değeri  $t$  belirlenir. Farz edelim ki  $f_1$  ve  $f_2$  iki öznitelik ve bunların SSD değeri SSD1 ve SSD2 olsun, yine aynı şekilde  $f_1$  ve  $f_3$  arasındaki SSD değeri SSD2 olsun. SIFT aşağıdaki kuralı sağlayan özniteliği eşleşmiş olarak kabul eder.

$$\frac{SSD1}{SSD2} < t \quad (2.1.5.2)$$

Eşitlik 2.1.5.2’de verilen  $t$  değeri 0.8 seçildiğinde nesne tanıma için iyi sonuçlar elde edildiği gözlemlenmiştir.

#### 2.2 PCA-SIFT

Bu öznitelik çıkarma metodu [28], SIFT metodundaki sadece 4. aşamada farklı bir yöntem kullanarak betimleyicileri elde etmektedir. Diğer aşamalar SIFT metodu ile aynıdır. SIFT metodunda son aşamada 4x4 parçalara histogramı uygulanarak betimleyici elde edildiği anlatılmıştı. Ama PCA-SIFT metodunda 39x39 parça alınır, bu parçadaki yatay ve dikey gradyen değerleri tek bir vektör olacak şekilde yan yana eklenir ve  $2 \times 39 \times 39 = 3042$  boyutuna sahip bir vektör elde edilir. Bu vektöre temel bileşen analizi uygulanarak (PCA), boyutu düşürülür. Örneğin; 3042 den 36 ya düşürülür.

#### 2.3 Speeded Up Robust Features (SURF)

SIFT metodundan daha hızlı bir yöntem geliştirilmeye çalışılmıştır. Bay ve Tuytelaars tarafından 2006’da önerilmiştir [29]. SURF yöntemi, SIFT yönteminden daha hızlı çalışmasına rağmen, görüş açısı değişikliği (viewpoint change) ve

parlaklık deęişiklięi (intensity change) gibi durumlarda SIFT kadar iyi sonuçlar vermemektedir.

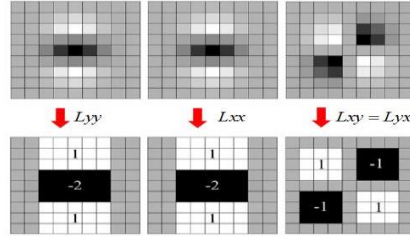
### 2.3.1 Anahtar Noktaları Bulma

SURF algoritması Hessian matrisini ve integral resmi (integral image) kullandıęından dolayı SIFT algoritmasından daha hızlı olmuştur. Hessian matrisinin damla (blob) ortaya çıkarma özellięinden yararlanarak resimdeki anahtar noktalar bulunur. Hessian matrisi simetriktir yani  $L_{xy}(x, \sigma)$  ve  $L_{yx}(x, \sigma)$  birbirine eşittir. Verilen bir I resminde,  $\sigma$  ölçeęindeki bir  $x=(x,y)$  noktasının Hessian Matrisi  $H(x, \sigma)$  Őu Őekilde elde edilir:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.3.1.1)$$

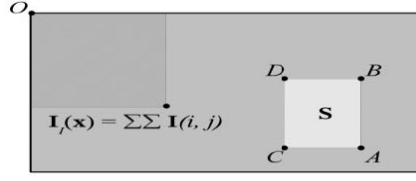
$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (2.3.1.2)$$

Burada  $L_{xx}(x, \sigma)$  , I resmindeki  $x$  noktasının Gauss filtresinin ikinci dereceden türevi alınarak evrimleştirilmesii ile elde edilen sonucu ifade etmektedir. Yine  $L_{xy}(x, \sigma)$  ve  $L_{yy}(x, \sigma)$  aynı Őekilde elde hesaplanır. SURF algoritması da çok ölçekli çalışmaktadır.



Őekil 2.3.1.1 Türevlerin yakınsamasına karŐılıklı Box filtreler

DeęiŐen boyutlardaki Box filtre dizisini (Box Filters) kullanarak elde ettięi sonuçlardan piramit oluŐturur. Box filtresi Gauss ikinci dereceden parçalı türevinin yakınsanması ile bulunur. Gauss ikinci dereceden parçalı türevinin yakınsanması denklem 2.3.1.2'deki hesaplanmaktadır.  $\sigma=1.2$  ölçeęindeki Gauss ikinci dereceden türevlerin yakınsamasına karŐılıklı gelen 9x9 boyutundaki Box filtreler 2.3.1.1 de verilmiŐtir:

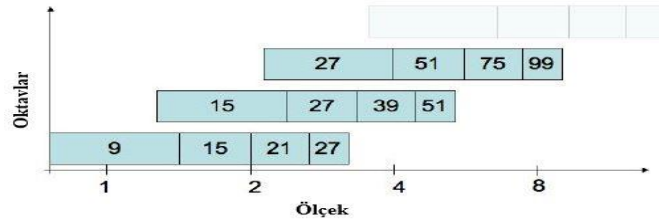


Şekil 2.3.1.2 İntegral image

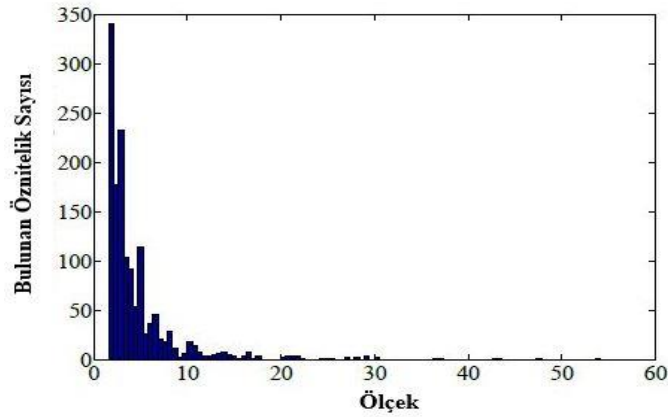
Şekil 2.3.1.1’de verilen Box filtreler [30] integral image denilen yöntem ile elde edilir [31]. Şekil 2.3.1.2’de görüldüğü gibi integral image kullanarak resimde  $S$  ile belirtilen alanın içindeki piksellerin toplamını aşağıdaki denklemi kullanarak bulabiliriz. İntegral image yöntemi Box filtrelerin daha hızlı hesaplanmasında kullanılır. Aşağıdaki denklemde görüldüğü gibi  $A, B$  ve  $C, D$  değerleri kullanılarak aradaki karenin değeri çok hızlı bir şekilde elde edilir.

$$S = (A + D) - (B + C) \quad (2.3.1.2)$$

*İntegral image* yöntemi ile elde edilen Box filtreler (Şekil 2.3.1.1 ) resme uygulanarak Şekil 2.3.1.3’deki gibi yığın (stack) oluşturulur [29]. Yine aynı çalışmada ölçek sayısı artıka bulunan özneliklerin sayısının düştüğü vurgulanmıştır (şekil 2.3.1.4).



Şekil 2.3.1.3 Box filtreler ile oluşturulan piramit

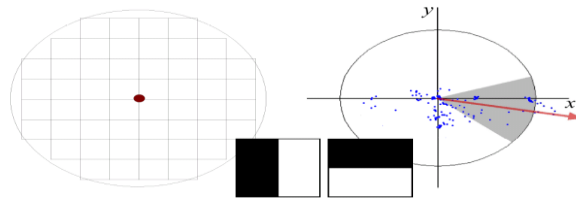


Şekil 2.3.1.4 Ölçeklerden elde edilen öznelik sayısı

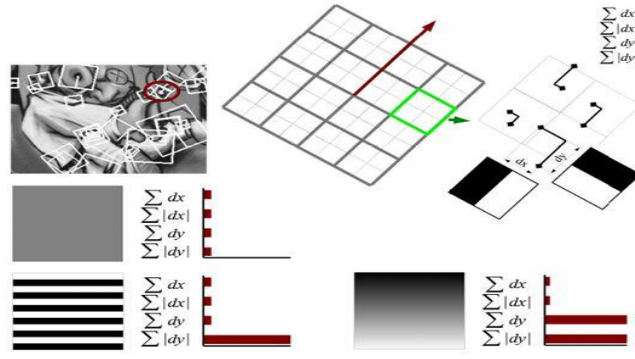
Şekil 2.3.1.4’de verildiği gibi bir noktanın anahtar nokta olup olmadığı Box filtrelerden elde edilen ölçeklerden yararlanarak bulunur. Ardışık 3 ölçekten 3x3 lük alanlar seçilir. Toplamda 3x3x3=27 tane piksel arasında en yüksek gradyene sahip piksel anahtar nokta elde edilir.

### 2.3.2 Dönmeden Etkilenmeyen Anahtar Noktalarını Bulma

SURF algoritması, Şekil 2.3.2.1-2’de gösterildiği gibi anahtar nokta etrafında çember şeklinde bir alan seçer ve bu alana Haar wavelet filtrelerini uygulayarak betimleyiciler (descriptors) elde etmektedir [26]. Haar wavelet dönüşümü seçmesinin temel nedeni; parlaklığa karşı duyarsızdır ve İntegral image ile hızlı bir şekilde hesaplanır. Bunun için anahtar nokta etrafında  $20\sigma$  büyüklüğünde bir alan belirlenir, yani üzerinde bulunduğumuz ölçeği elde ederken kullandığımız  $\sigma$ ’nın 20 katı büyüklüğünde bir alan ve bu alan  $5\sigma$  olacak şekilde 4x4 lük karelere bölünür. Bu 4x4 lük alanlara Haar wavelet filtresini yatay ve dikey şekilde uygulanıp, x ve y yönündeki türevler hesaplanır, sırasıyla  $dx$  ve  $dy$ . Elde edilen türev değerleri, anahtar noktadan uzaklıklarına göre  $2\sigma$  ile ağırlık atanır. Sonra çember  $60^\circ$  derecelik parçalara bölünüp, her parçadaki Haar wavelet filtresinden elde edilen  $\sum dx$ ,  $\sum dy$ ,  $\sum |dx|$  ve  $\sum |dy|$  değerleri bulunup toplamda  $4 \times (4 \times 4) = 64$  tane öznitelik vektörü oluşturulur. Eğer  $dx < 0$ ,  $dx > 0$ ,  $dy < 0$  ve  $dy > 0$  gibi kriterler kullanılıp bu şartları sağlayan noktaların toplamı alınılırsa bu öznitelik vektörün boyutu 128’i bulmaktadır. Buradaki işlemler Şekil 2.3.2.2 de gösterilmiştir.



Şekil 2.3.2.1 SURF yönteminde betimleyici elde etme



Şekil 2.3.2.2 SURF öznitelik çıkarma

### 2.3.3 Öznitelikleri Eşleştirme

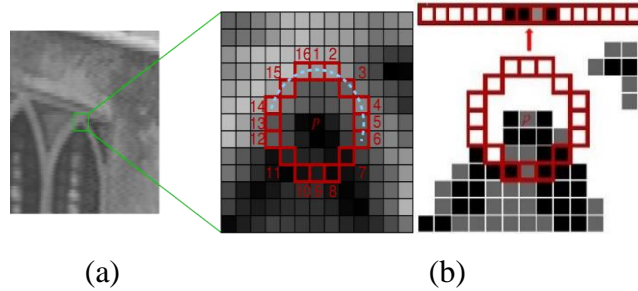
SURF algoritması yine daha hızlı çalışmak için farklı iki resimden elde edilen öznitelikleri eşleştirirken hızlı bir yöntem geliştirmiştir. SURF yöntemine göre iki ayrı noktadan elde edilen Hessian matrislerinin diyagonal elemanlarının toplamı o matrislerin yani "*trace*" işlemi, bize matrisin işaretini verir. Basit bir şekilde, eğer iki matrisin işareti aynı ise bu noktalar eşleşmektedir kararı verilir.

### 2.4 Features From Accelerated Segment Test (FAST)

FAST köşe çıkarma yöntemi kısmen SUSAN köşe kriteri tabanlı olarak çalışır [32, 33]. SUSAN yöntemine benzer şekilde, FAST yöntemi bir aday noktanın köşe olup olmadığına karar vermek için 16 piksellik çember (yarıçapı 3 olan Bresenham çemberi) kullanır. Şekil 2.5.1 (a) da çizildiği gibi, parlaklık değeri  $I_p$  olan bir  $p$  aday noktası seçtiğimiz farz edelim. Saat yönünde her bir piksel 1 den 16 ya kadar numaralandırılır. Algoritmanın hızlı çalışması için ilk olarak,  $I_p$  değerini çemberdeki 1, 5, 9 ve 13 değerleri ile karşılaştırılır. Eğer bu noktalar arasından en az üç tanesi belirlenen eşik değerini sağlıyorsa bu durumda  $p$  noktası ilgi noktası olarak seçilir. Bir başka açıdan, eğer seçilen 4 nokta arasından en az üç noktanın piksel değeri  $I_p + T$  değerinden büyük veya küçük değilse,  $p$  noktası ilgin nokta olarak seçilir. Tersisi durumunda, eğer seçilen 4 nokta arasından en az üç noktanın piksel değeri  $I_p + T$  değerinden büyük veya küçük ise, tüm 16 piksellere bakılır ve bu 16 piksel arasından en az 12 bitişik noktanın verilen şartı sağlaması gerekir. Benzer şekilde, imgedeki geriye kalan tüm noktalar için aynı işlem uygulanır.

Bazı kısıtlamalardan dolayı, örnek olarak eğer bitişik noktası 12 den küçük ise algoritma iyi çalışmamaktadır. Ayrıca optimal bir sayıya karar verilemediğinden dolayı ve bazen birkaç tane ilgi noktası komşu olarak bulunulduğundan, bir tane makine öğrenimi metodunu bu sıkıntılardan kurtulmak için kullanmışlardır. Bu durumda, Şekil 2.4.1 (b)'de çizildiği gibi her bir  $p$  aday noktası için 16'lık vektör oluşturulup bir eğitim seti oluşturulmuştur. Bu durumda seçilen 16 pikselin her birine “karanlık (darker)”, “benzer (similar)” ve “parlak (brighter)” olmak üzere üç ayrı sınıfa ayrılmıştır. Bu kurala bağlı olarak, öznitelik vektörü  $v$  üç ayrı altkümeye ayrılmıştır;  $P_s$  (benzer noktalar),  $P_d$  (karanlık noktalar) ve  $P_b$  (parlak noktalar). Bundan sonra ID3 (bir karar ağacı sınıflandırıcı) eğitim setine uygulanır ve entropi azaltma kriterine bağlı kalınarak, bir pikselin köşe noktası olup olmadığına bakılır. Böylece birinci problem sınıflandırıcı yönteminin yardımıyla çözülür.

İkinci problem yani birçok köşe ilgi noktasının birbirine komşu olması problemi ise aday köşe noktaları bulunduktan sonra bir maksimum olmayanları bastırma yöntemi ile giderilebilir. Yay üzerindeki ve merkezdeki piksellerin mutlak değerlerinin toplamının farkı elde edilmesi ile yapılır, sonra iki ilgi noktasının değerleri karşılaştırılır ve küçük olan ihmal edilir. Aşağıda olan Şekil 2.4.1 (a-b) [34] referansında verilen web sitesinden alınmıştır.



Şekil 2.4.1 (a) A bir ilgi noktası (b) bir etrafından vektör oluşturma

## 2.5 Binary Robust Invariant Scalable Keypoints (BRISK)

SIFT, SURF ve benzeri yöntemlerde elde edilen yerel öznitelikler, bazı dönüşümlerde, örneğin dönme, görüş açısı değişimi ve diğer bozulmalarda bir imgeyi temsil etmeye yetebiliyor iken, fakat bu yöntemlerden elde edilen betimleyiciler ağ taşıma yükünü kısıtlı olduğu durumlarda veya çok büyük işlem gerektiren şartlarda kullanılmaları çok büyük sıkıntılara neden olmaktadır. İşte bu

açığın fark edilmesi ile birçok ikili betimleyicilerin ortaya çıkmasına neden olmuştur. FAST tabanlı olan BRISK metodu [35] bunlardan sadece biridir.

Genel olarak BRISK üç parçadan oluşmaktadır: bir örneklem örüntüsü, yön kompensasyonu ve örneklem çiftleri. Burada örneklem örüntüsü almak demek FAST yöntemindeki çember üzerindeki noktaları almak anlamına geliyor. Sonra bu çiftler kısa mesafeli ve uzun mesafeli çiftler olarak iki alt kümeye ayrılır. Rotasyondan etkilememe için, uzun mesafeli çiftlerin gradyen değerlerinin toplamına bakılarak belirlenir ve kısa mesafeli çiftler elde edilen yönlere göre döndürülür. Son olarak tüm çiftler için birinci ve ikinci noktaların parlaklık değeri karşılaştırılır. Eğer birinci noktanın değeri ikincisinden büyük ise çıktı 1 olur aksi durumda çıktı 0 olur. Böylece 512 çift için 512 tane bitten oluşan betimleyici oluşturulur. Eşleştirme aşamasında Öklid uzaklığı yerine daha kısa çalışan Hamming uzaklığı kullanılır. Bunu gerçekleştirmek için sadece XOR işleminin toplam değerine bakılması ikili betimleyicilerin karşılaştırılması için yeterlidir.

## 2.6 Maximally Stable Extremal Regions (MSER)

Bir MSER bölgesi [36-38] belirli bir eşik değeri üzerindeki ve değişen eşik değerine göre bağlılıkları kalıcı olan birbirine bağlı nokta kümesinden oluşmaktadır. Diğer bir kelime ifade edersek, seçilen bölge bir grup eşik değeri dizisine göre değişmeyen yerel ikili şekillerdir. Referans [36] de verilen çalışmaya göre, MSER bölgeleri bulma Watershedding yöntemine benzer şekilde çalışır. Tıpkı bir sinema filmini hızlı hızlı oynatmak gibi, eşik değeri [0-255] arasında değişecek şekilde değiştirilir ve bu arada tüm sahnelerde değişmeyen veya çok az değişen bir birine bağlı bölgeler MSER bölgesi olarak adlandırılır.

## 2.7 Binary Robust Independent Elementary Features (BRIEF)

BRIEF betimleyicisi [39, 40] yöntemlerinden ilham alınarak bulunmuş betimleyici çıkaran yöntemdir [41]. Kompakt, hızlı çalışan ve çok ayırtedici olan Hamming uzaklığı hızlı eşleştirme ve iyi bir tanıma performansı olan bir yöntemdir. Fakat imgenin döndürülmesi veya bozulması durumuna çok hassastır. Önemli bir nokta olarak vurgulanmak gerekirse, BRIEF yöntemi sadece bir betimleyici çıkarıcıdır, yani bir anahtar nokta çıkarmaz. Bu yüzden bir anahtar nokta bulan

SURF, SIFT, FAST, HARRIS veya MSER gibi yöntemler ile çalışmak zorundadır. BRIEF'in önerildiği çalışmada [41] yazar SURF yerine CENSURE anahtar nokta bulucuyu kullanmamızı önermektedir. BRIEF yönteminde bir parça (patch) alınır ve gürültüye karşı bir ön işlem olarak yumuşatılır. Sonra bu parça içinde aldığı pikselleri birbiriyle karşılaştırarak 1 veya 0 gibi çıktılar üretir. Örneğin, p ve q bir parçadan alınan iki piksel olsun;

$$\tau(\text{Patch}; p, q) := \begin{cases} 1 & \text{eğer } I(p) < I(q) \\ 0 & \text{değilse} \end{cases} \quad \text{olur.} \quad (2.8.1)$$

Yani p'nin parlaklık değeri (I(p)) q pikselinden (I(q)) küçük ise 1 değilse 0 şeklinde ikili dizi oluşturmaktadır. Daha sonra aşağıdaki denklemde gösterildiği gibi  $n_d$  boyutlu bir BRIEF dizisi oluşturmaktadır.

$$f_{n_d}(\text{patch}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(\text{patch}; p_i, q_j) \quad (2.8.2)$$

## 2.8 Oriented FAST and Rotated BRIEF (ORB)

ORB yöntemi [42] FAST ve BRIEF birleşimi olan bir yöntemdir. Değişen ölçeklerde değişmeyen anahtar noktaları bulmak için imgenin ölçek piramidini çıkararak FAST yöntemini modifiye eder. Her bir ölçekte, anahtar noktalar FAST yöntemini uygulayarak çıkartılır. Anahtar noktalar elde edildiğinde, Harris köşe metriği kullanılarak anahtar noktalar sıralanır ve sadece en üstten N tane nokta belirli bir eşik değerine bağlı olarak seçilir. Dönmeye karşı güçlü noktalar elde etmek için, yerel yönlerin belirlenmesi için parlaklık kütle merkezi (bir yerel parçadaki piksellerin ağırlıklandırılmış değeri) boyunca birinci momentler hesaplanarak bulunur. Daha sonra döndürülen parça üzerinde BRIEF betimleyicileri hesaplanır ve ikili dizi şeklinde ORB betimleyicileri oluşturulur.

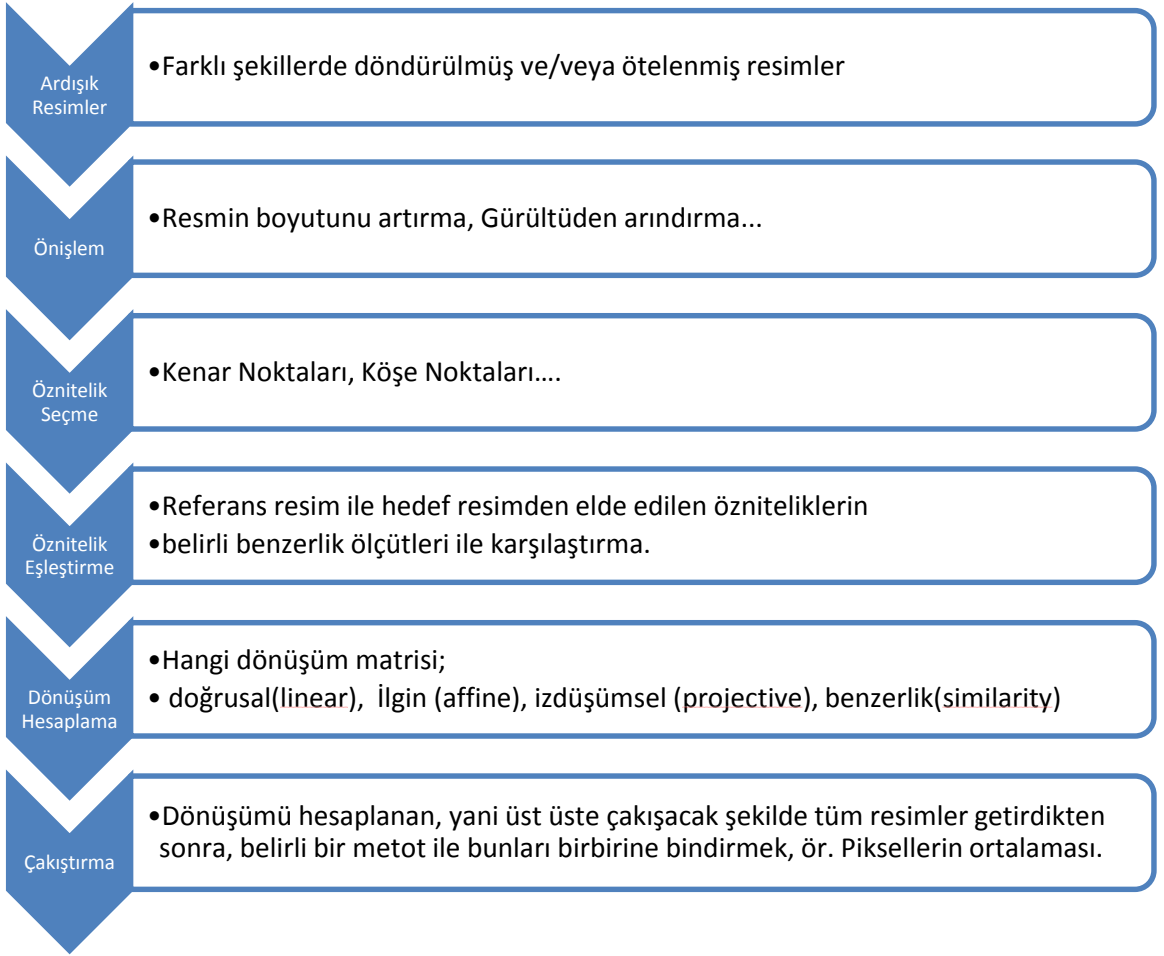
## 2.9 Fast Retina Keypoint (FREAK)

FREAK yöntemi [43] de bir ikili betimleyicidir ve örnekleme örüntüsü ile çift seçmeyi BRISK yönteminden ödünç alır. Genel olarak, gözdeki retina bölgesinden ilham alınarak bulunulan ve merkezden retina örnekleme gidildikçe yoğunluğu üstel olarak azalan çembersel bir örüntü seçen bir betimleyicidir. Rotasyona karşı dirençli betimleyiciler elde etmek için seçilen bir parçada, merkez noktasına simetri

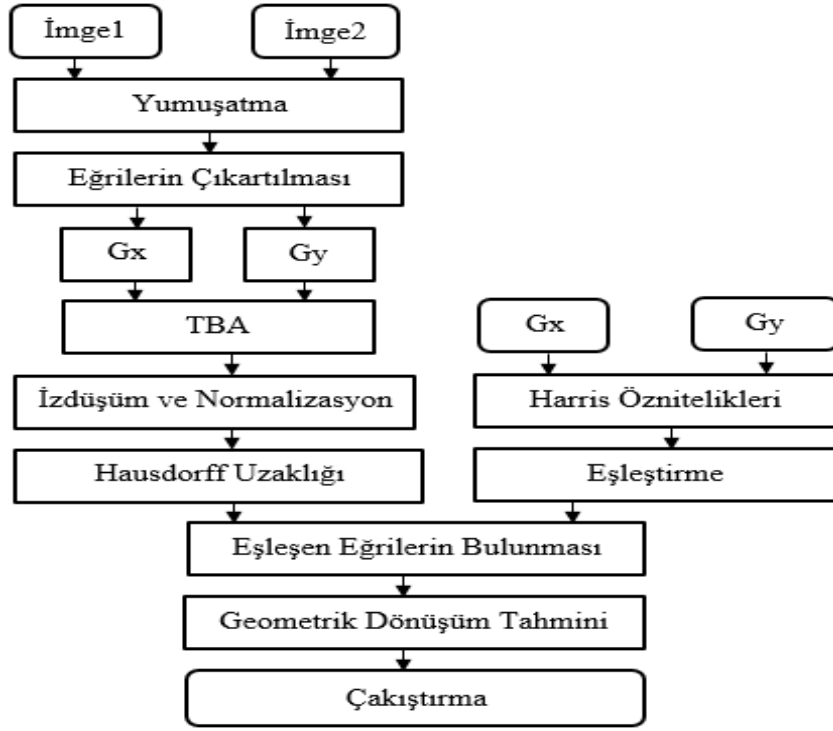
olan çiftlerin yerel gradyenlerin toplamını alır. Betimleyici oluşturma aşaması için ORB de kullanılan yöntemin bir benzerini kullanmaktadır. Basit olarak en az ilişkili örüntüyü seçer. Genel olarak en iyi performansı ikili betimleyici dizisinin uzunluğu 512 olunca vermektedir.

### 3 ÖNERİLEN YÖNTEM

Giriş bölümünde değindiğimiz gibi görüntü çakıştırmanın birden fazla yolu bulunmaktadır. Bu çalışmada önerilen ise genel olarak piksel tabanlı ve özel olarak eğri tabanlı bir görüntü çakıştırma yöntemidir. Genel olarak öznitelik tabanlı bir görüntü çakıştırma işlemi beş aşamadan oluşmaktadır. Şekil 3.1’de gösterildiği gibi bu aşamalar sırasıyla; önişlem, öznitelik seçme, öznitelik eşleştirme, dönüşüm hesaplama ve referans resim ile dönüşümü hesaplanan resmin çakıştırılmasıdır.



Şekil 3.1 Genel olarak görüntü çakıştırma işlemleri



Şekil 3.2 Önerilen Yöntem

Önerilen yöntemin çalışma prensibi Şekil 3.2’de verilmiştir. Verilen şekle göre, ön işlem olarak imge sanatsal köşe ve kenar koruyarak imge yumuşatılır ve imgedeki kenarların bulunarak eğriler çıkartılır. Elde edilen eğrilerin koordinatlarına türev uygulanarak yatay (Gx) ve dikey (Gy) gradyenler elde edilir. Bu gradyenlere TBA uygulanır ve en büyük özdeğere karşılık gelen özvektör kullanılarak eğrilerin temel yönleri tespit edilir. Eğriler temel yön üzerine izdüşüm edilerek normalize edilir. Farklı eğrilerden elde edilen normalize edilmiş katsayılara Hausdorff yöntemi uygulanarak geometrik uzaklık hesaplanır. Bunun yanında ikinci bir eşleştirme olarak imgelerin gradyen değerleri bulunulur ve gradyen imgedeki eğri üzerinden Harris öznitelikleri çıkartılır ve eğriler eşleştirilir. İkinci eşleştirmenin katkısı yanlış eşleşen eğrileri ihmal etmek ve kümedeki eğri sayısını azaltmaktır. Buna göre ikinci eşleştirmede elde edilen eğriler arasından en küçük Hausdorff uzaklığına sahip olan eğriler en iyi eşleşmiş olarak kabul edilmektedir. Sonra bu eşleşen eğriler arasındaki geometrik dönüşüm hesaplanır ve imgeler çakıştırılır.

### 3.1 Önişlem

Genel olarak bir ham veriyi işlemeden önce bazı önişlemlerden geçirince yapılan işin türüne göre doğruluğun artığı gözlemlenmiştir. Örneğin bir kenar tanıma için resmi Gauss yumuşatma filtreden geçirmekte fayda vardır. Çünkü resimde gürültü olabilir ve yumuşatma işlemi gürültüyü bastıran bir tekniktir. Bir başka önişlem de resimde segmentasyon yapmak için verileri normalize etmek, verileri bir matematiksel yöntem ile gürültüden arındırmak ya da resmi yüksek geçirgen filtreden geçirip gereksiz ve tekrarlayan verileri eleme bir çeşit önişlem diyebiliriz.

Aynı şekilde çakıştırılacak resimlerde de bu saydığımız tüm önişlemlerin yanında resmin boyutunu artırma veya resmi başka alan dönüştürme (TBA alanı, OVY alanı vb.) gibi önişlemlerin yapılmasında doğruluğun artırılması bakımından çok büyük öneme sahiptir. Bunun yanında resmi döndürme, boyutunu artırma ve x, y eksenlerinde değişik oranlarda boyutlandırma işlemleri iki aşamadan oluşmaktadır. Bu aşamalar eldeki piksel değerlerini kullanarak resmi döndürmek ve döndürülen resmin tamsayı piksel koordinatları enterpolasyon kullanarak bulunması işlemleridir.

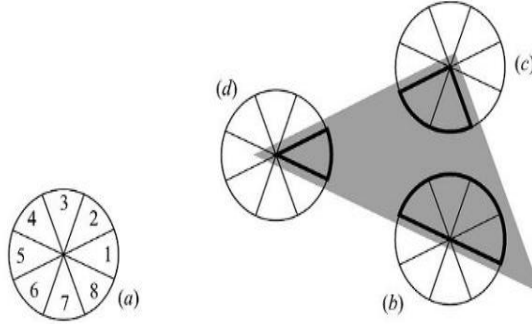
Bu çalışmada önişlem olarak resimleri bir yumuşatma işleminden geçirilmiştir. Bunun için [44] de verilen sanatsal resimleme için kenar ve köşeleri koruyarak yumuşatma yöntemi kullanıldı. Bunun yerine [45] de sözü edilen yumuşatma filtrelerini kullanabilirdi. Bu filtrelere örnek olarak; Kuwahara Filtresi (Kuwahara Filter [46]), Gauss Kuwahara Filtresi (Gaussian Kuwahara Filter [47]), Yapısal Kapatma (Structural Closing [48]), Alan Açıp-Kapama (Area Open-closing [49]), Çift-Yönlü Filtreleme (Bilateral Filtering [50]), Medyan Filtreleme (Median Filter [51]) veya Gauss yumuşak geçirgen (Gaussian Low-Pass Filter [52]) filtreleri verilebilir.

İkinci önişlem olarak Canny algoritması kullanılarak resimdeki kenar kesitleri elde edildi. Bir önceki stepte uygulanan yumuşatma işlemi ile gereksiz ve doku desenleri (texture) detaylar resimde en aza indirilmiştir. Canny algoritmasından dönen kenar kesitleri bu çalışmanın can alıcı noktalarından biridir. Çünkü bu kesitler ilerleyen aşamalarda birbiriyle eşleştirilecektir.

### 3.1.1 Sanatsal Kenar ve Köşe Koruyarak Yumuşatma

Fotoğrafik resimlerden boyama gibi sanatsal efektler, köşe ve kenarları koruyarak resmini yumuşatılması ile elde edilebilir. Fakat tüm kenar ve köşe koruyan yöntemler aynı performansı sergileyememektedir [45]. Bu çalışma için [44] de önerilen yöntem kullanılmıştır. Bu yöntem ile elde örnekler için [45] de verilen web sitesini ziyaret edebilir.

Yukarda referans verilen çalışmada, basit bir doğrusal ve yinelemeli olmayan operatör kullanılarak boyama efektleri deney resmine eklenilmiştir. Bir pikselin etrafında yerel bir dairesel komşuluk oluşturularak  $N$  tane parçaya ayrılmaktadır. Sonra, bu daire üzerinde yine  $N$  tane ağırlıklandırılmış yerel ortalama ve  $N$  tane ağırlıklandırılmış yerel standart sapma hesaplanmakta ve her bir piksel için, hesaplanan  $N$  tane yerel ortalamanın lineer kombinasyonu kullanılarak bir çıktı elde edilir.  $N$  tane yerel ortalamanın katsayıları, yerel standart sapmaların azalan fonksiyonlarına denk gelmektedir. Bu çalışmada  $N$  sayısı 4 olarak, kenar keskinlik değeri 1 ve yumuşatma için kullanılan değer 3 olarak alınmıştır.



Şekil 3.1.1.1 Artistik yönteminin davranışlarının sergilenmesi

Şekil 3.1.1.1'de önerilen yöntemin çalışma prensibi gösterilmiştir. Kenarın hiç olmadığı alanlardan (durum  $a$ ), elde edilen çıktı, yerel ortalama (average) değerlerin aritmetik ortalamasına (mean) denk gelmektedir.



**Şekil 3.1.1.2** Sanatsal köşe ve kenar koruyarak resmi yumuşatma

Bu durumda, önerilen operatör GaussFiltresine çok benzer davranmaktadır ve bu şekilde doku desenleri (texture) ve gürültü (noise) yok edilmiştir. Bir kenarın oluşması durumunda (durum *b*), operatör ile kesişen sektörler, diğerlerine göre daha yüksek yerel standart sapma değerleri vermektedir. Böylece, yani durum *b* de gösterildiği gibi, kenar ile kesişen sektörler (5-8), lineer kombinasyona daha az bir katkı sağlamaktadırlar. Benzer şekilde, köşelerin var olması durumunda (durum *c*), ve keskin köşeler (durum *d*), sadece köşe noktasına düşen sektörler (6-7 durum *c* için ve 1 durum *d* için), hissedilebilir bir katkı sağlarlar iken diğer sektörlerin etkisi önemsizmeye değmeyecek kadar azdır.

Şunu vurgulamak gerekir ki, özellikle kenar ve köşeleri korumak için yerel ortalamalar (averages) hesaplamak için dairesel sektör şekilli bölgeler seçip kullanmak çok uygun olmaktadır. Önerilen yöntem, her bir piksel için, sektörlerin otomatik seçilmesine imkân vermektedir. Ayrıca, bu yöntem ile sektörlerin piksel veya köşe olduğu analiz sonucunda en iyi şekilde çözmektedir. Bundan dolayı bizde çalışmamızda bu yöntemi kenar olmayan noktaları bastırmak için kullandık. Şekil 3.1.1.2’de sanatsal köşe ve kenar koruyarak resim yumuşatma ile elde edilen görsel sonuçlar sergilenmiştir.

### 3.1.2 Canny Kenar Tanıma Yöntemi

Kenar tanıma bilgisayarla görü uygulamalarında imgedeki nesnelere belirgin özelliklerini ortaya çıkarmaya çalışan bir süreçtir. Genel olarak özellikler nesnelere foto metrik, geometrik ve fiziksel özelliklerinden ortaya çıkan süreksizliklerdir. Kenar bir nesne ile bölge, nesne ile nesne veya bölge ile bölge arasındaki değişimler olarak tanımlanır. Önerilen birçok kenar tanıma yöntemlerinin temel amacı imgelerde piksel değerlerindeki değişimlerin gerçekleştiği yerlerin belirlenmesidir. Nesne tanıma, hedef izleme ve bölütleme gibi daha ileri seviyeli imge işleme yöntemlerinin içinde ilk aşamalardan biri olarak yer aldığından, kenar tanıma birçok görüntü işleme probleminin başarısını etkileyen en önemli unsurlardan biridir.

Kenar tanıma için zaman içinde konuyla ilgili çeşitli yöntemler önerilmiştir. Optimal bir kenar haritası, kenar algılama tabanlı çalışan diğer işleri önemli bir şekilde etkilediğinden dolayı sürekli, iyi yerleştirilmiş, seğirmesiz ve tek piksel kalınlığında olmalıdır. Ayrıca, kenar algılama imge hakkında tüm önemli bilgileri sağlamalıdır. Bu amaç için, imgenin türevleri imge hakkında yeteri kadar bilgiyi sunabilmektedir. Bununla birlikte, türev alma işlemi, elektronik, anlamsal ve sayısallaştırma/niceleme etkileri gibi çeşitli gürültü kaynaklarına karşı oldukça hassas olduğundan dolayı tek bir nokta kenar olarak işaretlenebilmektedir. Bu tür etkilerden kurtulmak için resmin yumuşatılması birçok çalışmada [52-55] önemli bir şekilde vurgulanmaktadır. İlaveten, yumuşatma işleminin de bilgi kaybı ve lokalizasyon problemleri gibi dezavantajları bulunmaktadır. Biz bu çalışmada bu tür sorunları yok etmek için kenar tanıma işleminden önce bir ön işlem olarak resmi

etkili yöntem ile yumuşattık. Bu yöntemin nasıl çalıştığı bir önceki aşamada anlatıldı.

Resim kenarlar ve köşeler korunarak yumuşatıldıktan sonra Canny yöntemi kenar haritası çıkartıldı. Canny yöntemi [56] görüntü işleme ile ilgili birçok çalışmanın ilham kaynağı olmuştur. Bunun nedeni Canny yönteminin hızlı sürede ve değiştirilebilir parametreler kullanarak resimden kenarlar çıkartmasıdır. Canny kenar tanıma metodu 4 adımdan oluşmaktadır:

- Yumuşatma
- Gradyen hesaplama.
- Düşük gradyen değerlere sahip noktaları bastırma (Non-maximum suppression).
- İkili eşikleme (Hysteresis Thresholding) ile istenmeyen ayrıntıları yok etme ve noktaları birbirine bağlama.

Yaygın olarak yumuşatma için Gauss Filtresi kullanılır. Gradyen hesaplamadan önce resmin gri değerleri [0-255] elde edilir. Gradyen Hesaplama, genel olarak vektörel veya noktasal olarak türev alma işlemidir. Önce yatayda türev alınır,  $G_x$  hesaplanır. Sonra dikeyde türevde alınır,  $G_y$  bulunur. Takip eden denklemde verildiği gibi  $G$  gradyen haritası olmak üzere,  $G_x$  ve  $G_y$  mutlak değerleri toplamı veya  $G_x$  ve  $G_y$ 'nin karelerinin toplamının karekökü bize gradyen haritasını ( $G$ ) verir.

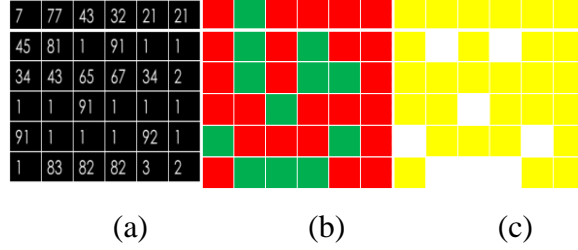
$$G = |G_x| + |G_y| \text{ veya } G = \sqrt{G_x^2 + G_y^2} \quad (3.1.2.1)$$

$$\theta = \arctan (G_y / G_x) \quad (3.1.2.2)$$

Gradyen haritası elde edildikten sonra kenar noktalarının yönü Eşitlik 3.1.2.2' deki gibi hesaplanır. Sonra bu yönler bağli olarak kenar inceltme işlemi yani *Non-maximum suppression* uygulanır. Diğer bir deyişle, belli bir yönde ilerlerken eğer işlenen noktanın gradyen değeri iki komşunun gradyen değerinden küçük ise sıfır değeri verilir. Böylece ince ve doğru şekilde kenarlar çıkmaktadır.

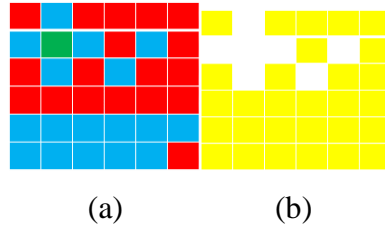
Daha sonra ikili eşikleme (Hysteresis thresholding) eşikleme ile zayıf noktaları eleyip, birbirinden ayrık kenar parçalarını birleştirme işlemi takip eder. Bunun için biri küçük, biri büyük iki eşik değeri belirlenir [57]. Eğer seçilen pikselin gradyen ağırlığı (gradient magnitude) değeri yüksek eşik değerinden büyük ve tepe noktası ise kenar noktası olarak işaretlenir. Alçak eşik değerini ise kenarları

birbirine bağlamak için kullanırız. Diğer bir deyiş ile eğer seçilen nokta tepe noktası olursa ve seçilen noktanın gradyen ağırlığı düşük eşik değerinden büyük olursa ve seçilen noktanın komşularından biri yüksek eşik değerinden büyük bir piksele bağlı ise seçilen nokta bir kenar noktası olarak işaretlenir.



Şekil 3.1.2.1 Canny yüksek eşik değeri çıktıları

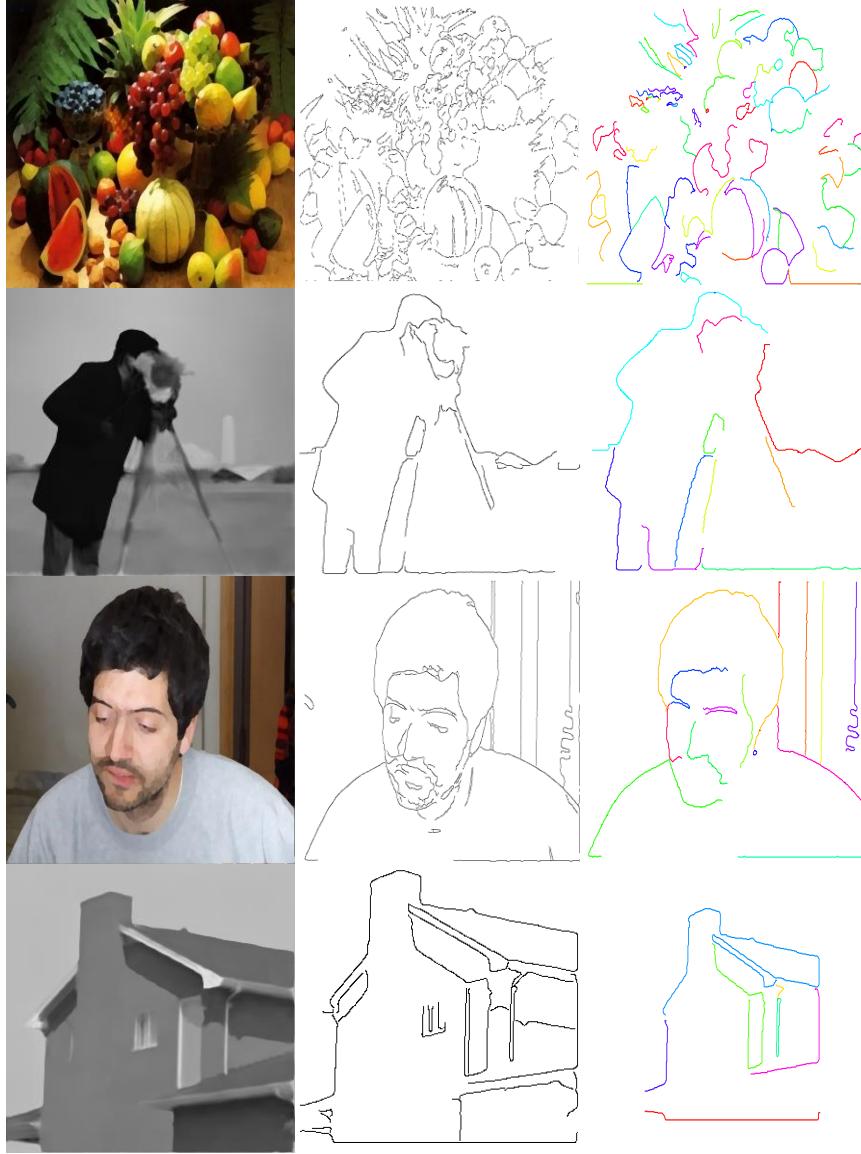
Şekil 3.1.2.1’de [57] (a) yüksek eşik değeri sonucu, (b) gradyen değerleri ve (c) tepe noktaları verilmiştir. Eşiklendirme Sonrası yüksek eşik değeri 80 olarak seçildikten sonra geriye kalan pikseller gösterilmiştir. Resimlerdeki kırmızı pikseller tepe olmayan noktaları, yeşil pikseller tepe noktalarını, sarı pikseller kenar olmayan noktaları ve beyaz pikseller eşiklendirme sonucu geriye kalan pikselleri göstermektedir.



Şekil 3.1.2.2 Canny alçak eşik değeri çıktıları

Şekil 3.1.2.2’de [57] (a) yüksek ve alçak eşik değeri uygulanınca elde edilmiş pikseller ve (b) alçak eşiklendirme sonucu ortaya çıkarılan kenar haritası resmedilmiştir. Genel olarak yüksek ve alçak eşik değerleri kullanıcıdan tarafından belirlenir. Mavi renk ile gösterilenler düşük tepe noktaları, yeşil ile gösterilenler yüksek tepe noktaları, kırmızı ile gösterilenler ise gereksiz olan noktalar, sarı ile gösterilenler kenar olmayan noktalar ve beyaz ile gösterilenler kenar noktalarıdır. Ortadaki resimde yüksek ve alçak eşiklendirme sonucu kalan pikseller (yeşil ve mavi) gösterilmiştir. Yine sağdaki resimde beyaz noktalar kenar noktası olarak işaretlenmiştir. Görüldüğü gibi kenar noktaları yüksek eşik değeri ile elde edilen

yüksek tepe noktalarının etrafında toplanmıştır. Yani komşularında biri yüksek eşik değerine sahip olmayan bir noktaya bağlı olan pikseller elenmiştir.



Şekil 3.1.2.3 Canny yöntemi ile elde edilen kenar çıktıları

Şekil 3.1.2.3'de Canny algoritması sonucu ortaya çıkan kenar örnekleri gösterilmiştir. Şekil 3.1.2.3'de görüldüğü gibi Canny yönteminden bazı kıstaslar kullanılarak elde edilen kenar haritaları sergilenmiştir. Soldaki sütunda olan resimler girdi resimleri, ortadaki sütunda olan resimler ise uzunluğu 100 piksel ve üzeri olan kenar parçaları görülmektedir. Matlab 2014a kullanılarak Canny yönteminin kodlarını kullanılmıştır. Canny yöntemindeki yüksek eşik değeri 0.2 ve

alçak eşik değeri 0.1 olarak seçilmiştir. Canny yönteminde kullanılan yumuşatma değeri ise 1 olarak seçilmiştir.

Bu çalışma için önemli olan kenar segmentleri sağ tarafta verilen segmentlerdir. Çünkü bu çalışmanın amacı eğri tabanlı kenar tanıma olduğu için kenar parçaları eğri olarak kullanılacaktır. Yani iki farklı görüntünün birbirine benzer olduğu ispatlamak için elde edilen eğri üzerindeki noktalar kullanılacaktır.

### 3.1.3 Ortak Vektör Tabanlı Kenar Tanıma Yöntemi (CVAED)

Bu çalışmanın amacı eğrilik tabanlı görüntü çakıştırma yani eğri üzerinden öznitelikler çıkartarak görüntü çakıştırma olduğundan dolayı, kenar tanıma bu çalışmanın önemli aşamalardan bir tanesidir. Bundan dolayı bu çalışmada yeni kenar çıkarma yöntemleri önerilmiştir. Bunlardan biriside OVY tabanlı kenar tanıma yöntemidir.

OVY örüntü tanıma uygulamalarında kullanılmış bir altuzay yöntemidir. OVY'nin uygulandığı alanlara örnek olarak ses tanıma, kişi tanıma, yüz tanıma, istenmeyen-eposta tanıma verebiliriz. OVY'de genel olarak öznitelik vektörünün iki bileşenden oluştuğu varsayılır [58]: (i) sınıfın ortak özelliklerini barındıran bileşen (ortak vektör), (ii) ait olduğu sınıfın ortak özelliklerinden geriye kalan ayırt edici özelliklerini barındıran ve her öznitelik vektörü için farklı olan bileşen (fark vektör). OVY yöntemini diğerlerinden ayıran farkı; yetersiz veri problemine çözüm getirmesidir. Genel olarak, diğer yöntemler yetersiz veri durumunda çalışmamaktadır [59]. Ortak vektörün hesaplandığı ve uygulandığı iki durum vardır. Seçilen öznitelik vektörlerin boyutu  $n$  ve bu vektörlerin sayıları  $m$  olmak üzere; eğer  $n \geq m$  ise yetersiz veri durumu, tersi durumunda yani  $n < m$  ise yeterli veri durumu oluşmaktadır.

OVY ile kenar tanıma yöntemi 3 aşamadan oluşmaktadır:

- GaussFiltre'si ile imgeyi yumuşatma ve OVY kullanarak gradyen haritası çıkarma.
- Tepe noktaların bulunması.
- Akıllı bir yöntem ile kenar noktaların birbirine bağlanması.

**Gauss Filtre'si İle İmgeyi Yumuşatma Ve OVY Kullanarak Gradyen Haritası Çıkarma:** Bir imgeyi yumuşatmadaki temel amaç imgedeki anlamlı

bilgiyi koruyarak imgeyi gürültü, aykırı değerler ve keskin geçişlerden kurtarmaktır. Bundan dolayı bu çalışmada yumuşatma işlemi için Gauss alçak-geçiren ve standart sapması 1 ve boyutu 5 olan bir filtre kullanılmıştır.

Gradyen haritası çıkarma işlemi ise OVY yönteminden yararlanarak bulunulmuştur. Bir altuzay dönüşüm tekniği olan OVY yöntemi örüntü ve ses tanıma görevleri için tatmin edici sonuçlar vermektedir. Herhangi bir sınıf  $\vec{a}_i$  ile gösterildiği varsayıldığında  $\vec{a}_i = \vec{a}_{com} + \vec{a}_{i,diff}$  şeklinde yazılması mümkün olmaktadır. Bir ortak vektör ( $\vec{a}_{com}$ ) öznelik vektörleri arasındaki farkların sınıf üyelerinden çıkarılmasıyla elde edilir ve sınıf boyunca değişmeyen vektöre denilir.  $\vec{a}_{i,diff}$  ise fark vektörünü göstermektedir. Ortak vektör yönteminde yeterli ve yetersiz veri durumu olmak üzere iki durum vardır. Öznelik vektörün boyutu vektör sayısından büyükse yetersiz, küçük veya eşit olduğunda yeterli veri durumu oluşmaktadır. Bu çalışmada ortak vektör yetersiz veri durumu kullanılmıştır

$$block = \begin{bmatrix} I_{i-2,j-1} & I_{i-2,j} & I_{i-2,j+1} \\ I_{i-1,j-1} & I_{i-1,j} & I_{i-1,j+1} \\ I_{i,j-1} & I_{i,j} & I_{i,j+1} \\ I_{i+1,j-1} & I_{i+1,j} & I_{i+1,j+1} \\ I_{i+2,j-1} & I_{i+2,j} & I_{i+2,j+1} \end{bmatrix} \quad (3.1.3.1)$$

Burada ortak vektör yöntemini daha detaylı anlatmak için bir blok verilmiştir. Eğer yatay yönde ortak vektör bulunacak ise her bir sütun vektör olarak seçilmektedir. Ama dikey yönde ortak vektör bulunacak ise bu sefer her bir satır vektör olarak seçilmektedir. Örnek olarak amacımız dikey yönde ortak vektör elde etmek olsun.

$$\vec{a}_i = \begin{bmatrix} I_{i-2,j-1} \\ I_{i-1,j-1} \\ I_{i,j-1} \\ I_{i+1,j-1} \\ I_{i+2,j-1} \end{bmatrix}, \vec{a}_{i+1} = \begin{bmatrix} I_{i-2,j} \\ I_{i-1,j} \\ I_{i,j} \\ I_{i+1,j} \\ I_{i+2,j} \end{bmatrix}, \vec{a}_{i+2} = \begin{bmatrix} I_{i-2,j+1} \\ I_{i-1,j+1} \\ I_{i,j+1} \\ I_{i+1,j+1} \\ I_{i+2,j+1} \end{bmatrix} \text{ ve } \vec{a}_{ave} = \frac{\vec{a}_i + \vec{a}_{i+1} + \vec{a}_{i+2}}{3} \quad (3.1.3.2)$$

Yukardaki eşitlikte verildiği gibi herbir sütun vektör olarak seçilmekte ve bu vektörlerin ortalaması ortalama vektörü verdiği görülmektedir. Burada seçilen vektörün boyutu 3 ve toplam vektör sayısı 5 olduğu açık olarak görülmektedir. Bu durumda ortak vektör yetersiz veri durumunu sağlamaktadır. Yani vektörün boyutu vektörlerin sayısından büyük olduğu görülmektedir.

Yetersiz veri durumunda bir sınıfa ait ortak vektör, ortak değişinti matrisinin sıfır özdeğerlerine karşılık gelen özvektörlerin lineer kombinasyonlarının yönündedir [60].

$$C_{n \times n} = \sum_{i=1}^m (\vec{a}_i - \vec{a}_{ave})(\vec{a}_i - \vec{a}_{ave})^T \quad (3.1.3.3)$$

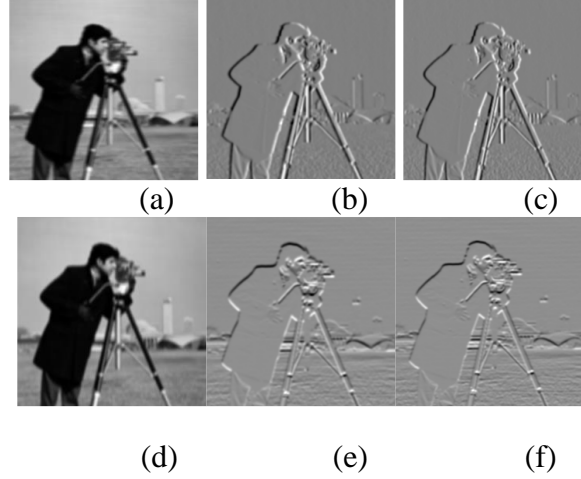
Bir sınıf için ortak değişinti matrisi Eşitlik 3.1.3.3'de gösterildiği gibi verilmiştir. Ortak değişinti matrisi öznelik vektörlerin ortalama vektöre göre değişimini vermektedir. Yetersiz veri durumunda ortak değişinti matrisinin sıfır olmayan özdeğerleri farklılık altuzayı B için birimlik taban oluşturmaktadır. Yine ortak değişinti matrisinin sıfır özdeğerlerine karşılık gelen özvektörler farksızlık  $B^\perp$  altuzayını germektedir. Bu durumda ortak vektör sıfırdan farklı öz değerlere karşılık gelen öz vektörlerin fark alt uzayı üzerine izdüşümlerinin vektörün kendisinden çıkarılmasıyla da elde edilmektedir.

Ortak vektör elde etmek için fark alt uzayı taban vektörleri veya farksızlık alt uzayı taban vektörleri kullanarak elde edilebilir. Eğer k tane farksızlık alt uzayı taban vektörünü kullanarak ortak vektör elde etmek hedeflenirse:

$$P_{cov}^\perp = \sum_{i=1}^k u_i u_i^T \quad (3.1.3.4)$$

$$\vec{a}_{com} = \sum (\vec{a}_{ave}^T u_i) u_i \quad \text{veya} \quad \vec{a}_{com} = P_{cov}^\perp \vec{a}_{ave} \quad (3.1.3.5)$$

Bu çalışmada yetersiz veri durumu olduğu kabul edilerek Eşitlik 3.1.3.5'deki gibi ortak vektör elde edilmiştir. Burada i. özdeğere karşılık gelen öz vektör  $u_i$  ile gösterilmiştir. Ayrıca  $P_{cov}^\perp$  ortak değişinti matrisini ve  $\vec{a}_{ave}$  ortalama vektörü göstermektedir. Aşağıda ortak vektör kullanılarak elde edilen ortak dikey ve yatay gradyenler gösterilmiştir.



**Şekil 3.1.3.1** Ortak vektör yöntemi ile elde edilen ortak ve fark resimleri

Şekil 3.1.3.1’de (a) ile (d) dikey ve yatay ortak resimleri, (b) ile (e) dikey ve yatay fark resimleri ve (c) ile (f) Sobel operatöründen elde edilen dikey ve yatay gradyen resimlerini göstermektedir.

**Tepe Noktaların Bulunması:** Bu çalışmada ortak vektör yöntemi kullanılarak türev işlemi gerçekleştirilmiştir. Geleneksel kenar bulma algoritmaları gibi bu çalışmada gradyen haritası yatay ve dikey türevlerin mutlak değerlerinin toplamı alınarak bulunulmuştur. Dikey yöndeki gradyen haritasının yatay yönüne bölünmesi ile herbir noktanın yönü tespit edilmiştir.

Ayrıca baskın olmayan noktaları yoketmek ve tepe noktaları çıkarmak için Otsu eşiklendirme yöntemi [61] çalıştırılmıştır. Otsu yöntemi sınıflar arası (inter-class) değişikliği artırarak veya sınıf içi (intra-class) değişikliği azaltarak optimal eşik değeri veren bir yöntem olarak bilinmektedir. Otsu yöntemini daha detaylı anlamak için [61] referansa bakılabilir. Kenar noktası belirlemek için aşağıdaki kural kullanılmıştır.

$$\forall (i, j), I(i, j) = \text{kenar\_değil}, \quad \text{eğer} \begin{cases} I(i, j) < Otsu\_Eşikleme \\ I(i, j) < I(i+n_x, j+n_y) \\ I(i, j) < I(i-n_x, j-n_y) \end{cases} \quad (3.1.3.6)$$

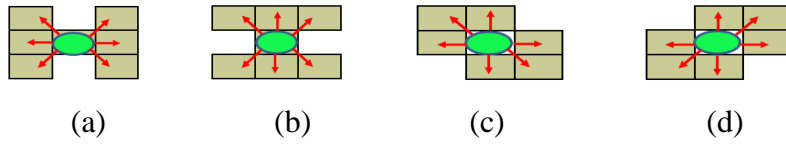
$$\forall (i, j), I(i, j) = \text{kenar}, \quad \text{değilse}$$

Burada  $-1 \leq n_i \leq 1$ ,  $i = x, y$  olarak verilmiştir. Ayrıca  $n_x$  ve  $n_y$  işaretleri verilen noktadaki kenarın yönüne bakılarak karar verilmiştir.

**Kenarların Birbirine Bağlanması:** Bu çalışmada tepe noktaların birbirine bağlamak için ED [62] yönteminde kullanılan temel fikir kullanılmıştır. Şekil

3.1.3.2’de gösterildiği gibi 4 yön temel alınmıştır. Önerilen yöntem eğer işlenen noktanın yönü;

- (a) doğu veya batı yönünde ise doğu-batı boyunca,
- (b) kuzey veya güney yönünde ise kuzey-güney boyunca,
- (c) güneydoğu veya kuzeybatı yönünde ise güneydoğu-kuzeybatı boyunca,
- (d) kuzeydoğu veya güneybatı yönünde ise kuzeydoğu-güneybatı boyunca noktalar birbirine bağlanarak ilerlemektedir.



Şekil 3.1.3.2 Kenarları bağlama yönleri

**DeneySEL Sonuçlar:** Bu çalışmanın performansı ROC [63, 64] veritabanı üzerinde deneyler yapılarak Ray’ın kenar tanıma yöntemi [65] ile karşılaştırılmıştır. Görsel sonuçlar Şekil 3.1.3.3’de verilmiştir.



Şekil 3.1.3.3 OVY tabanlı kenar çıkarma çıktıları

Verilen şekilde ilk satır ROC veritabanından alınmış orijinal resimlerdir. İkinci satırda ise bu resimlere ait *ground truth* çıktıları verilmiştir. Ground truth çıktılarının nasıl oluşturulduğunu öğrenmek için [63] de verilen çalışmaya bakılabilir. Üçüncü satırda Ray'ın kenar tanıma sonuçları ve dördüncü satırda önerilen yöntemin sonuçları sergilenmiştir. Ray için verilen sonuçlar referansında verilen çalışmadan alınmıştır.

Yukardaki görsel sonuçlara baktığımızda önerilen yöntem ile elde edilen kenarlar daha açık, kesintisiz ve anlamlı oldukları görülürken, Ray'ın yönteminde bazı detayların saklandığı görülmektedir. Bunun nedeni önerilen yöntemin diğer yöntemlerden farklı olarak ortak vektör yöntemini kullanarak vektörel olarak türev

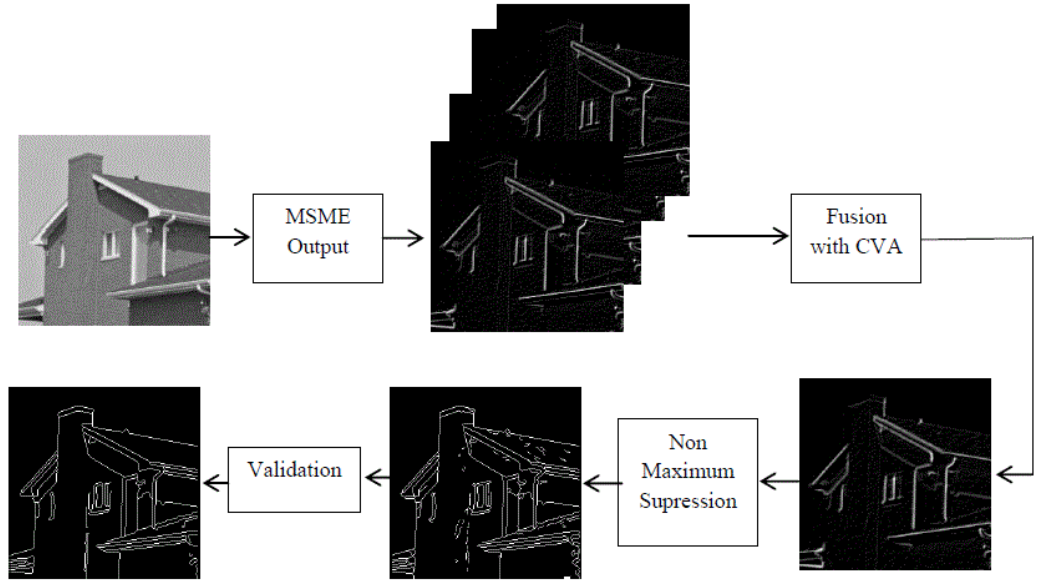
almasıdır. Yine belli bir uzunluktan küçük olan kenar parçaları elenerek önerilen yöntemin performansı artırılabilir. Ayrıca bu çalışmada kullanılan Otsu eşikleme yerine başka bir yöntem kullanılmasında performansın artacağı beklenmektedir.

### 3.1.4 Ortak Vektör Tabanlı Çok Ölçekli-Uzmanlı Kenar Tanıma

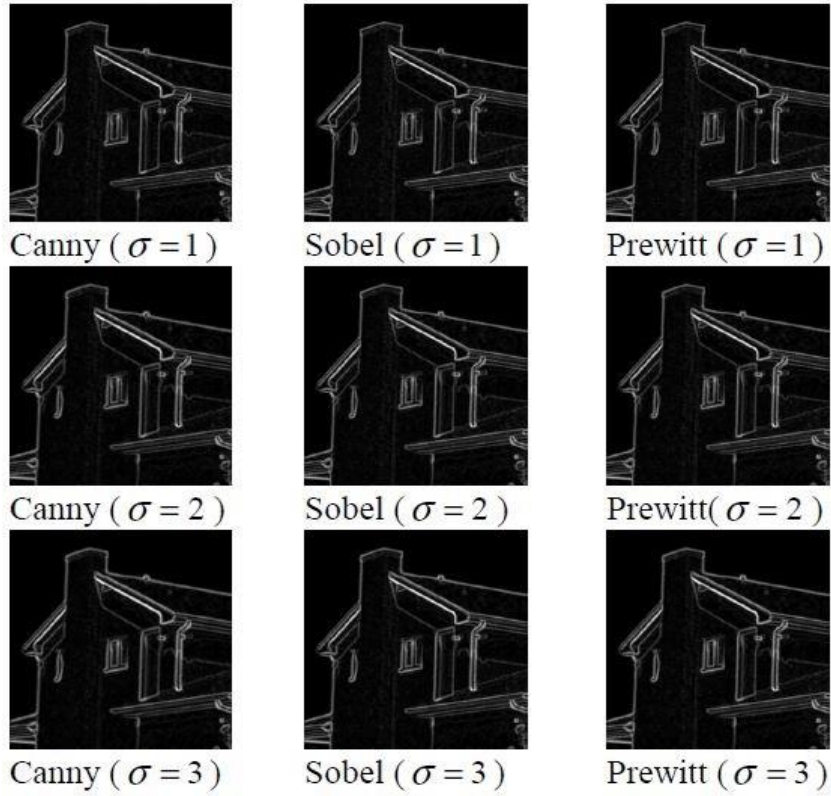
Bir diğer önerilen çalışma ise ortak vektör tabanlı çok-ölçekli ve çok-uzmanlı yeni bir kenar tanıma metodudur (MSME-CVAED) [66]. Yine her kenar tanıma yönteminde olduğu bu çalışmada amaç diğer çalışmalardan daha iyi kenar çıkartacak bir yöntem önermektir. Tek bir ölçek kullanılarak yeterli olmadığı varsayımından hareketle, çok ölçekli yöntemlerin amacı aynı görüntünün değişen ölçekli sürümlerini kullanarak kenar çıkarmaktır.

Önerilen yöntemin çalışma prensibi Şekil 3.1.4.1’de sergilenmektedir. İlk olarak farklı filtreler değişen ölçeklere uygulanarak birçok gradyen haritası çıkartılmaktadır. Bu gradyen haritaları ortak vektör yöntemi kullanılarak birleştirilmektedir (Fusion with CVA). Tek bir gradyen haritası çıkarıldıktan sonra baskın olmayanları bastırma (Non-Maximum Suppression) işlemi uygulanmaktadır. En son olarak belli bir uzunluk değerinden küçük olan kenarlar elenmektedir. Takip eden başlıklarda önerilen yöntemin tüm aşamaları detaylı olarak açıklanmıştır.

Şekil 3.1.4.2’de verildiği gibi çok uzmanlı için Canny, Prewitt, Sobel operatörleri ve çok ölçekli olması için varyans değerleri;  $\sigma = 1$ ,  $\sigma = 2$  ve  $\sigma = 3$  olmak üzere üç farklı Gauss yumuşatıcı filtresi kullanılmıştır. Farklı kombinasyonlar denenerek toplam olarak 9 farklı gradyen haritası elde edilmiştir. Bu gradyen haritaları OVY yöntemi kullanılarak tek bir gradyen haritasına dönüştürülmüştür.



Şekil 3.1.4.1 Çok ölçekli çok uzmanlı kenar tanıma modeli

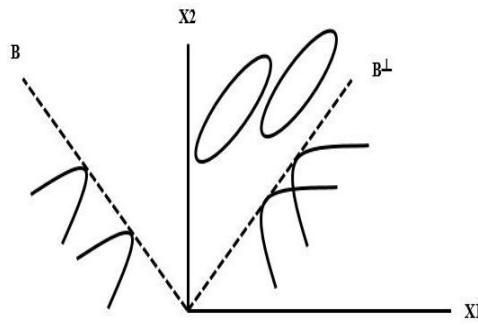


Şekil 3.1.4.2 Önerilen yöntem ile elde edilen 9 farklı gradyen haritası

**OVY Yetersiz Veri Durumu Kullanarak Birleştirme:** Çok-uzmanlı çok-ölçekli kenar tanıma yöntemlerinin altındaki temel amaç farklı gradyen haritalarındaki kenar bilgilerini tek bir gradyen haritasında toplamaktır. Bundan

dolayı bu çalışmada yine OVY yönteminin yetersiz veri durumunu gözönünde bulundurarak 9 farklı gradyen haritalarının ortak gradyen haritası çıkartılmıştır.

Yukarda da değindiğimiz gibi bir sınıfa ait seçilen öznelik vektörlerin boyutu  $n$ , vektörlerin sayılarından büyük ya da eşit ise yetersiz veri durumu oluşur. Yetersiz veri durumunda ( $n \geq m$ ) ortak vektör ve farklılık altuzayı Gram-Schmidt dikleştirme yaklaşımı veya altuzay teknikleri kullanılarak elde edilebilir. Altuzay tekniğinde ortak vektör ile ortak değışinti matrisinin (covariance matrix) özvektörleri arasındaki ilişkiden yararlanır.



Şekil 3.1.4.3 İki vektöre ait Farksızlık ve Farklılık altuzayları

Şekil 3.1.4.3'de  $X_1$  ile  $X_2$  iki ayrı vektör ve  $B$  ile  $B^\perp$  bu iki vektörün uzaydaki farklılık ve farklılık altuzayları verilmiştir.  $\mathcal{R}^n$  bir sınıfa ait öznelik vektörleri barındıran bir sonsuz uzay olsun ve  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m \in \mathcal{R}^n$  ile gösterilsin. Bu sınıfa ait ortak değışinti matrisi  $\Phi$  Eşitlik (1) deki gibi elde edilebilir.

$$\Phi = \sum_{i=1}^m (\mathbf{a}_i - \mathbf{a}_{ort}) (\mathbf{a}_i - \mathbf{a}_{ort})^T \quad (3.1.4.1)$$

Eşitlik (3.1.4.1)'de  $\mathbf{a}_{ort}$ , sınıf ortalama vektörünü göstermektedir. Bu eşitlikte verildiği gibi, OVY' de öznelik uzayı, fark altuzayı  $B$  ve farklılık altuzayı  $B^\perp$  olmak üzere birbirine dik iki altuzaya ayrılır. OVY yöntemine göre, farklılık altuzayı  $B^\perp$ , ortak değışinti matrisinin sıfır özdeğerlerine karşılık gelen özvektörler tarafından gerilir. Buna göre ortak vektör fark alt uzayındaki her vektöre dik olduğundan,  $B^\perp$  farklılık uzayında yer almalıdır. Yani ortak vektör ortak değışinti matrisinin sıfıra eşit özdeğerlerine karşılık gelen özvektörlerin doğrusal bileşimidir. Yetersiz veri durumu ( $n \geq m$ ) için ortak değışinti matrisi  $\Phi$ ,  $n-m+1$  adet sıfır

özdeğere sahip olacaktır. Bu özdeğerler büyükten küçüğe sıralandığında fark ve farksızlık altuzayları için özdeğerler aşağıdaki şekilde seçilir [67]:

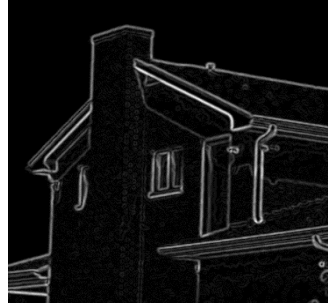
$$\lambda_n \geq \lambda_{n-1} \geq \dots \geq \lambda_{n-m+2} \geq \lambda_{n-m+1} \geq \dots \geq \lambda_2 \geq \lambda_1$$

$$\underbrace{\hspace{10em}}_{m-1} \quad \underbrace{\hspace{10em}}_{n-m+1}$$

$$\lambda_i \neq 0 \quad \lambda_i = 0$$

Fark Altuzayı                      Farksızlık Altuzayı

Bu durumda ortak vektör, sıfırdan farklı özdeğerlere karşılık gelen özvektörlerin fark alt uzayı üzerine izdüşümlerinin vektörün kendisinden çıkarılmasıyla da elde edilebilir. Şekil 3.1.4.4’de 9 gradyen haritasının tek gradyene dönüşümü gösterilmiştir.



Şekil 3.1.4.4 Tek bir gradyen haritası

**Baskın Olmayan Noktaları Bastırma ve İkili Resim:** Kenar çıkarmanın doğası olarak baskın olmayan noktaların yok edilmesi gerekmektedir. Bu yüzden bu çalışmada [65] verilen yeni bir eşiklendirme yöntemi kullanılmaktadır.

$$D_E = \frac{\text{Yerel baskılama sonucu kalan toplam piksel değeri}}{\text{İmgedeki toplam piksel değeri}} \quad (3.1.4.1)$$

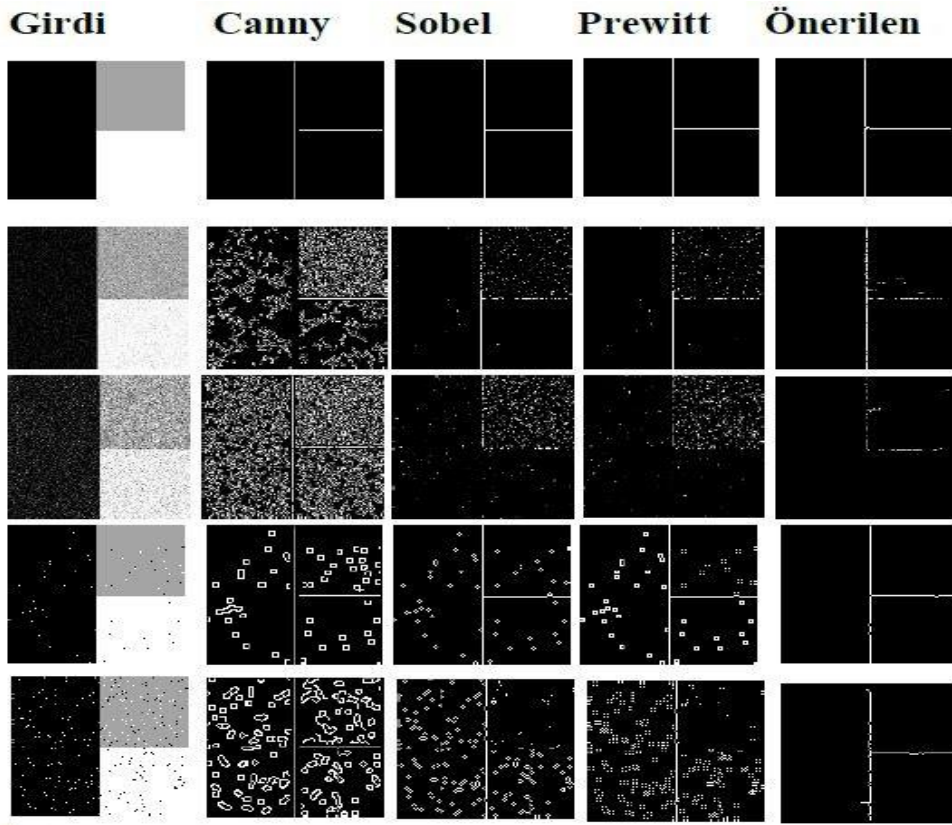
Tek bir eşik değerinin en iyilemesi yukardaki gibidir. Eşik değerinin en iyilenmesi için 8 bit resimlerde 0 dan başlayıp 255’e kadar eşik değerleri sıralı olarak uygulanmış ve  $D_E$  değerleri elde edilmiştir. Elde edilen  $D_E$  eşik değerleri aşağıda verilen formülle standart hale getirilmiştir.

$$Z = \frac{D_E - \mu}{\sigma} \quad (3.1.4.2)$$

Burada  $\mu$  ve  $\sigma$ , seçilen örneklerin ortalaması ve standart sapmasıdır. Yani normalize edilmiş eşik değeri ( $Z$ ) elde edilmiştir. Ortalama ve standart sapma

hesaplanırken 0 eşik değerine sahip olan  $D_E$  değerleri göz ardı edilmiştir. Elde edilen standartlaştırılmış verinin 0 geçiş noktası eşik değeri olarak seçilmiştir. Bu seçme yöntemi basit, hesaplaması kolay, imge boyutuna göre normalize edildiğinden herhangi bir boyuttaki imgelere uygulanabilmesi, bağlantı bilgisi ve kenar büyüklüğü bilgisi vb. gibi ekstra bilgilere göre modifiye edilebilecek olması bir avantajdır.

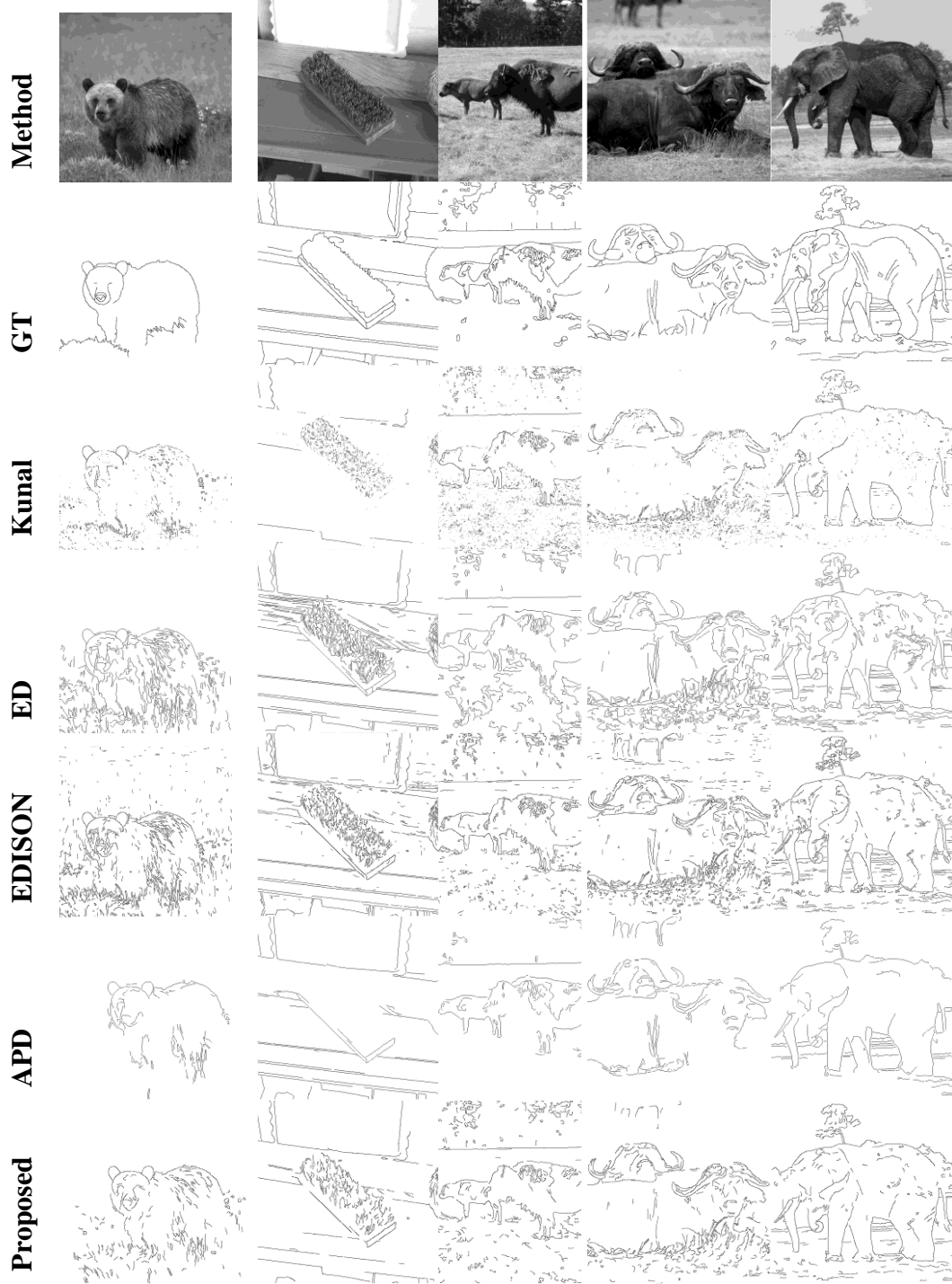
**Deneysel Sonuçlar:** Önerilen yöntem ile elde edilen sonuçlar farklı yöntemlerin sonuçları ile karşılaştırılmıştır. Karşılaştırma sonuçları Canny [56], Sobel [68], Prewitt [69] sonuçları ile karşılatırarak Şekil 3.1.4.5’de gösterilmiştir. Burada verilen resimlerden ilk satır: Gürültüsüz. İkinci satır: Gauss beyaz gürültü (ortalama 0, varyans 0.01). Üçüncü Satır: Gauss beyaz gürültü (ortalama 0, varyans 0.05). Dördüncü Satır: Tuz ve biber gürültüsü ( yoğunluğu 0.01 ). Beşinci Satır: Tuz ve biber gürültüsü (yoğunluğu 0.05).



Şekil 3.1.4.5 Gürültü resimlerde çok-ölçekli çok-uzmanlı kenar tanıma

Gürültülü resimlerde elde edilen sonuçlara bakıldığında Canny yönteminin Sobel ve Prewitten daha iyi olduğu görülmektedir. Gürültü değerlerinin artmasında

Canny yöntemi ile gürültülü olan yerlerde bulunan kenarların artığı gözlemlenmiştir. Bunun nedeni Canny yönteminin noktasal türev almasından kaynaklanmaktadır. Önerilen yöntemin başarısı ise en sağdaki sütunda net olarak görülmektedir.



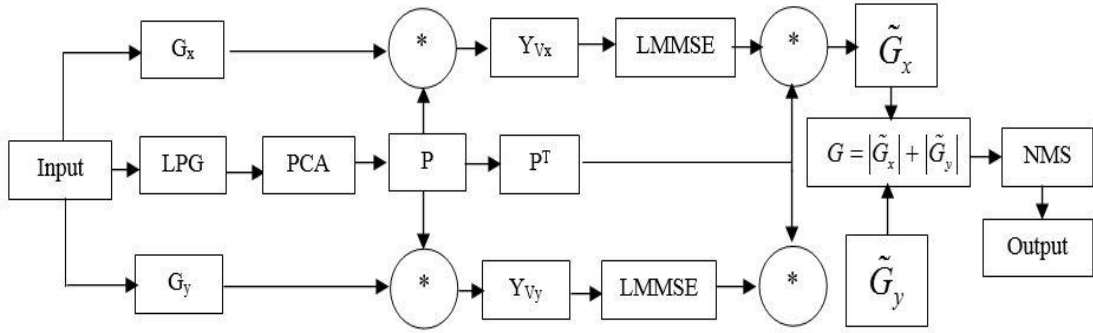
Şekil 3.1.4.6 Gürültüsüz resimlerde önerilen yöntemin performansı

Şekil 3.1.4.6 'da verildiği gürültüsüz resimlerde önerilen yöntemin performansı ölçülmüştür. Bunun için RUG veritabanı üzerinde denemeler yapılarak farklı çalışmaların sonuçları ile karşılaştırılmıştır. Önerilen yöntem ile elde kenar çıktıları APD [70], ED [62], KUNAL [65] ve EDISON [71] çalışmalarının sonuçları ile objektive olarak karşılaştırılmıştır. ED ve KUNAL'ın çıktıları hazır kodları kullanılarak, EDISON ve APD' çıktıları ise RUG veritabanının olduğu siteden [72] alınmıştır. Bütün yöntemleri varsayılan parametreleri kullanılmıştır.

Sonuçlara bakıldığında APD yöntemi ince ve anlamlı kenarlar bulmasına rağmen *fırçanın* olduğu resimde bazı detayları kaybetmiştir. Yine EDISON ve KUNAL yöntemlerinden elde edilen kenar segmentlerinin çentikli olduğunu görülmektedir. Önerilen yöntem ise ED ile benzer sonuçlar sergilemiştir.

### 3.1.5 Gürültülü Resimlerde Kenar Çıkarma: LPGPCA-ED

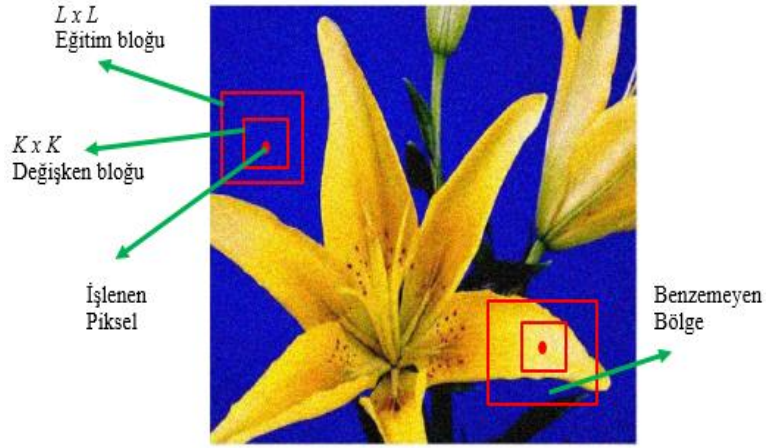
Bu çalışmada ise Gauss gürütüsü ile bozulmuş resimlerden kenar çıkarılmaya çalışılmaktadır. Temel bileşen analizinin yerel piksel gruplama üzerine uygulanarak imgeyi ayrıştırma ve gürültülü katsayıları yok etmek fikri bu çalışmada kenar tanıma için kullanılmaktadır. İlk olarak yatay ve dikey gradyenler çıkartılır ve bu gradyenlere LPGPCA (Principal Component Analysis based Local Pixel Grouping) yöntemi uygulanarak imge farklı bir alana dönüştürülmektedir. Daha sonra gürültülü alanlar lineer en az ortalama hata karelerinin tahmini (Linear Minimum Mean Square Error Estimation (LMMSE)) yöntemi ile gürültüden temizlenmektedir. Gürültü temizleme işleminden sonra imge TBA dönüşümün tersi kullanılarak yeniden eski alanına dönüştürülmektedir. Daha sonra gürütüden arındırılan gradyen haritası üzerinde baskın olmayan noktaların giderilmesi için [65] anlatılan yöntem kenar tanımda kullanılmaktadır. Önerilen yöntemin nasıl çalıştığı Şekil 3.1.5.1'de verilmiştir.



Şekil 3.1.5.1 LPGPCA-ED yöntemi

**Yerel Pksel Gruplama ile Eğitim Verisi Oluşturma:** Gürültü bir resim  $I_v = I + v$  şeklinde yazılabilmektedir. Burada  $I$  orijinal resmi ve  $v$  eklenen gürültüye ait değerleri tutmaktadır. Görüntüyü gürültüden arındırmadaki temel hedef gürültülü resimdeki bilgiyi ve gürültü karakteristik özelliği kullanılarak gürültüsüz bir resim elde etmektir. Gürültü arındırmada yerel piksel gruplama yapmak kenarlar ile bölgeleri birbirinden ayırmak için yapılmaktadır. Bir bölgeyi (patch) gürültüden arındırmaya çalışıldığında eğer başka bir bölgeden parçalarda eklenmiş ise bu parçalarda gürültü olarak kabul edilebilmektedir.

Gürültü ilişkisiz ve yeri değişmediğinden dolayı, benzer şekilde gürültü bir bölge  $\vec{a}_v = \vec{a} + v$  şeklinde yazılabilir. Şekil 3.1.5.2’de gösterildiği gibi işlenen pikseli gürültüden arındırmak için  $K \times K$  boyutunda pencere gürültü resim  $I_v$  üzerine oturtulur ve  $\vec{a}_v = [a_1 \dots a_n]^T$  şeklinde sütun vektörü halinde gösterilir. Burada  $n = K^2$  olacak şekildedir. Seçilen pencere  $L \times L$  boyutundaki bölge üzerinde gezdirilerek bu pencereye benzeyen bloklardan eğitim seti oluşturulur. Görüldüğü gibi burada  $L \times L$  bölgenin merkezi  $\vec{a}_v$  olacak şekilde alınmaktadır. Toplamda  $\vec{a}_v$  merkezli bir bölgeden  $(L - K + 1)^2$  eğitim bloğu elde edilir. Bir bölgeden benzer blokların elde edilmesi bir tür sınıflandırma işlemi olduğundan dolayı blok eşleştirme (block matching) veya k-ortalama algoritması (k-means algorithm) kullanılabilir. Bu çalışmada blok eşleştirme yöntemi kullanılmıştır.



Şekil 3.1.5.2 Gürültülü resimde sınıf seçmek

**LPG-PCA Kullanarak Gürültü Arındırma:** Bir önceki aşamada elde edilen eğitim blokları bu aşamada LPG-PCA kullanılarak gürültüden arındırma işlemi gerçekleştirilmektedir. Yukarıda anlatıldığı gibi eğer oluşturulan matrisde gürültü var ise Eşitlik 3.1.5.1'deki gibi yazılabilir.

$$X_v = X + V \quad (3.1.5.1)$$

Eğer amacımız gürültüden arındırma ise [73]'de verilen çalışmaya göre  $X_v$ 'nin merkezleştirilmesi gerektiği vurgulanmaktadır. Buna göre merkezleştirilen veri  $Z_c = X_v - \mu_{X_v}$  formunda ve  $Z_c = (X - \mu_x) + (V - \mu_v)$  şeklinde olabilmektedir. Eğer gürültü sıfır ortalamalı kabul edilirse verilen merkezleştirilen veri matris  $Z_c = X_c + V$  şeklinde olabilmektedir. Burada  $X_c$  merkezleştirilmiş gürültüsüz veriye denk gelmektedir.

Bu çalışmada amaç imgeyi gürültüden arındırmak yerine imgedeki kenarları bulmak olduğundan girdi verisini merkezleştirilmesine gerek duyulmamıştır. Bu amaç için  $x$  ait ortak değişinti matrisi  $\Omega_x$  şeklinde gösterilmekte ve TBA dönüşümü ortak değişinti matrisine bağlı olarak hesaplanmaktadır. Burada elde var olan  $X_v$  verisi gürültülü olduğundan dolayı  $\Omega_x$  verisi direk olarak hesaplanamamakta ve  $\Omega_{X_v}$  değerleri  $X_v$  verisi merkezleştirilmeden bulunması gerekmektedir. Buna göre  $\Omega_{X_v}$  aşağıdaki şekilde gösterildiği gibi formülize edilmelidir.

$$\Omega_{X_v} = \frac{1}{n} X_v X_v^T = \frac{1}{n} (XX^T + XV^T + VX^T + VV^T) \quad (3.1.5.2)$$

Burada X ve V ilişkisiz olduklarından  $XV^T$  ve  $VX^T$  değerleri sıfır olmaktadır.

$$\Omega_{X_v} \approx \frac{1}{n} (XX^T + VV^T) = \Omega_X + \Omega_V \quad (3.1.5.3)$$

Yukarda verilen eşitlikte  $\Omega_X = (1/n)XX^T$  ve  $\Omega_V = (1/n)VV^T$  olmaktadır.

Burada  $\Omega_V(i, j)$  bileşenlerinin değerleri  $v_i$  ve  $v_j$  arasındaki ilişkiyi göstermektedirler. Burada  $i \neq j$  için  $v_i$  ve  $v_j$  ilişkisiz olduklarından dolayı,  $\Omega_V$ 'nin bir  $M \times M$  diyagonal ve diyagonal bileşenleri  $\sigma^2$  oluşturması beklenmektedir. Bundan dolayı  $I$  birim matrisi olmak üzere  $\Omega_V$  matrisi  $\sigma^2 I$  şeklinde yazılabilmektedir. Sezgisel olarak  $\Omega_X$  matrisi ile bağlantılı  $P_X$ 'e ait TBA dönüşümü hesaplamak  $\Omega_{X_v}$  ait TBA dönüşümü hesaplamak ile aynı olmaktadır. Bunun kanıtlanması takip eden denklemler ile açıklanmaktadır. Bundan dolayı  $\Omega_X$  şeklinde ayrıştırılabilir.

$$\Omega_X = \Phi_X \Lambda_X \Phi_X^T \quad (3.1.5.4)$$

Bu eşitlikte  $\Phi_X$  matrisi bir  $M \times M$  birimdik özvektör matrisi ve  $\Lambda_X$  ise diyagonal özdeğer matrisidir. Yine  $\Phi_X$  birimdik matrisi olduğundan dolayı  $\Omega_V$  aşağıdaki eşitliklerdeki gibi yazılabilir:

$$\Omega_V = (\sigma^2 I) \Phi_X \Phi_X^T = \Phi_X (\sigma^2 I) \Phi_X^T \quad (3.1.5.4)$$

$$\begin{aligned} \Omega_{X_v} &= \Omega_X + \Omega_V = \Phi_X \Lambda_X \Phi_X^T + \Phi_X (\sigma^2 I) \Phi_X^T \\ &= \Phi_X (\Lambda_X + \sigma^2 I) \Phi_X^T = \Phi_X \Lambda_{X_v} \Phi_X^T \end{aligned} \quad (3.1.5.5)$$

Burada  $\Lambda_{X_v} = \Lambda_X + \sigma^2 I$ . Eşitlik 3.1.5.4 gözlemlendiğinde  $\Omega_{X_v}$  ve  $\Omega_X$  aynı özvektöre  $\Phi_X$  sahip olduğu çıkarılabilmektedir. Bundan dolayı pratik olarak,  $\Omega_X$  yerine  $\Omega_{X_v}$  matrisini ayrıştırarak  $\Phi_X$  ortak değişinti matrisi elde edilebilir. Böylece X matrisi için birimdik TBA dönüşümü aşağıdaki gibi olmaktadır:

$$P_X = \Phi_X^T \quad (3.1.5.6)$$

Böylece  $P_X$  matrisini  $X_v$ 'ye uyguladığımızda aşağıdaki eşitlik oluşmaktadır.

$$Y_V = P_X X_V = P_X X + P_X V = Y + V_Y \quad (3.1.5.7)$$

Burada  $Y = P_X X$  değeri  $X$  matrisi için ayrışım veriseti ve  $V_Y = P_X V$  ise gürültülü verisetinin  $V$  matrisi için dönüştürülmüş formuna denk gelmektedir. Burada  $Y$  ve  $V_Y$  ilişkisiz olduklarından,  $Y_V$  'ye ait ortak değişinti matrisi aşağıdaki eşitlik ile elde edilebilir:

$$\Omega_{Y_V} = \frac{1}{n} Y_V Y_V^T = \Omega_Y + \Omega_{V_Y} \quad (3.1.5.8)$$

Burada  $\Omega_Y = \Lambda_X$  ise ayrıştırılmış veriseti olan  $Y$  'nın ortak değişinti matrisidir. Bundan dolayı gürültülü veri olan  $V_Y$  'nin ortak değişinti  $\Omega_{V_Y} = P_X \Omega_V P_X^T$  şeklinde yazılabilmektedir. Bu yol kullanılarak gürültüsüz veri olan  $Y$  'ye ait enerjinin birçoğunun TBA dönüşüm alanı olan  $Y_V$  üzerine yoğunlaştığı kabul edilmektedir. Ayrıca gürültülü veri olan  $V_Y$  'ye ait enerjinin ise daha fazla durumu ortaya çıkardığına inanılmaktadır. Bu durumda gürültüyü elemek için LMMSE yöntemi kullanılmaktadır.

**LPGPCA Tabanlı Kenar Tanıma:** Şekil 3.1.5.1 önerilen yöntemin çalışma stratejisini özetlemektedir. Önerilen yöntemi çalışma prensibi aşağıdaki aşamalardan oluşmaktadır:

- İlk olarak her bir gürültülü blok için LPG ile TBA dönüşümü uygulanmakta ve gürültülü bloklar TBA alanına çekilmektedir. Böylece izdüşüm (projection) matrisi ve izdüşüm matrisinin tersi elde edilmektedir.
- Böylece referans bloktan Canny yöntemi ile yatay ( $G_x$ ) ve dikey ( $G_y$ ) olmak üzere iki ayrı gradyen haritası elde edilir.
- Elde edilen  $G_x$  ve  $G_y$  gradyen değerleri izdüşüm matrisi ile çarpılarak TBA alanına dönüştürülür.
- TBA alanında dönüştürülen gradyen katsayıları LMMSE yöntemi ile gürültüden arındırılmakta ve arındırılan alan tekrar zaman alanına yani gürültüsüz alana döndürülmektedir.
- Sonra gradyen haritası  $G = |\tilde{G}_x| + |\tilde{G}_y|$  şeklinde oluşturulmaktadır.

- Son olarak baskın olmayan noktaların bastırılması, Ray'ın [65] çalışmasında önerilen optimal eşikleme yöntemi kullanılarak yapılmaktadır.

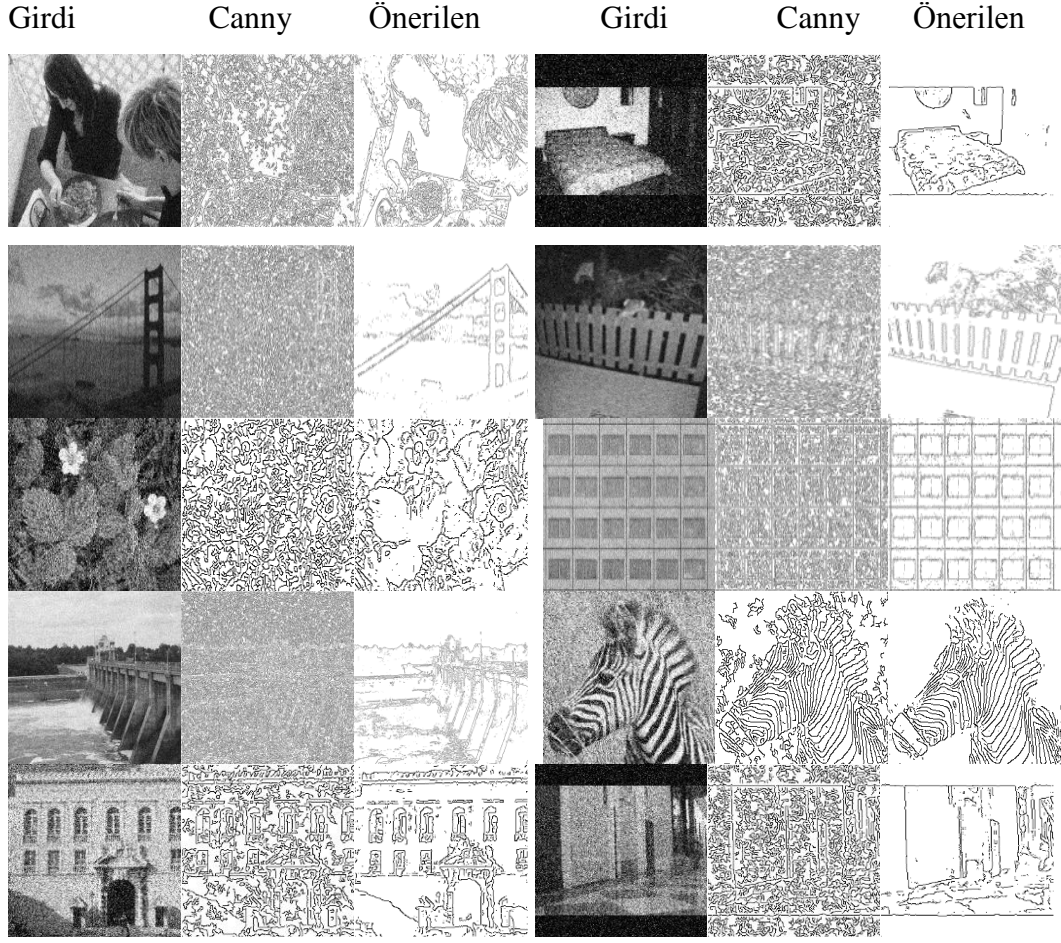
Bölüm 3.1.4'de anlatılan baskın olmayan noktaları bastırma ve ikili resim yazan başlık altında Ray'ın önerdiği eşiklendirme yöntemi detaylı olarak anlatılmıştır. Yine elde edilen eşik değeri  $NMS\_Threshold$  olmak üzere kenar (edge) noktaları aşağıdaki denklem ile elde edilmektedir.

$$\forall (i, j), I(i, j) = \text{kenar\_değil} \quad \text{eğer} \begin{cases} I(i, j) < NMS\_Eşikleme \\ I(i, j) < I(i + n_x, j + n_y) \\ I(i, j) < I(i - n_x, j - n_y) \end{cases} \quad (3.1.5.9)$$

$$\forall (i, j), I(i, j) = \text{kenar}, \quad \text{değilse}$$

Burada  $-1 \leq n_i \leq 1$ ,  $i = x, y$  olarak verilmektedir. Eşitlik 3.1.5.9'da görüldüğü gibi, önerilen yöntemin performansı seçilen eşik değerine bağlı olarak değişebilmektedir.

**Deneysel Sonuçlar:** Yine önerilen yöntem [74, 75]'den alınan resimler üzerinde test edilerek ve Şekil 3.1.5.3'de sonuçları sunulmuştur. Görüldüğü gibi önerilen yöntem Gauss 50db's gürültüsü ile bozulmuş resimlerde insan gözünün hoşuna giden kenar segmentleri bulabilmiştir. Tekrar hatırlatmak gerekirse bu kenar segmentleri daha sonra resimleri birbiriyle eşleştirmek yani hangisi hangisine benziyor onu tespit etmek için kullanılacaktır. Özellikler gürültülü resimlerde önerilen yöntemin performansı diğer öznelilik çıkarma yöntemlerini geçeceği hedeflenmektedir.



Şekil 3.1.5.3 LPGPCA ve Canny yöntemi ile elde edilen kenar segmentleri

### 3.2 Öznitelik Çıkarma

Öznitelik tabanlı imge çakıştırmaya göre resim geometrik konumları farklı olan birçok nokta kümesinden oluşmaktadır. Geometrik konumlarına ek olarak noktaların parlaklık değerleri ve yönleri ayrı birer özellik kabul edilir. Farklı dönüşümlere ve bozulmalara karşı dirençli noktaların seçilmesi görüntü çakıştırmaya veya eşleştirme işleminin performansını direkt etkilediği SIFT, SURF ve ORB gibi birçok öznitelik çıkarma yönteminde anlatılmıştır. Kısacası, bir çakıştırmaya işleminde önce bu öznitelikler arasındaki ilişki belirlenir, daha sonra bu ilişkiye göre dönüşüm parametreleri tahmin edilir.

Bu çalışmada bölüm 2 de verilen öznitelik çıkarma yöntemlerinin görüntü eşleştirme üzerindeki performansı ile önerilen eğri (curve) tabanlı öznitelik çıkarma yöntemi ile karşılaştırılmıştır. Eğri tabanlı görüntü eşleştirme yapılmasının nedeni; resmin ana iskeletinin bozulmalar karşısında değişmeyeceğinin varsayılmasıdır.

Resmin gri değerlerinin kullanılıp görüntün eşleştirmek zaman kaybına neden olmakta ayrıca şartlar değişti mi sonuçlar da değişmektedir. Örneğin, çok az bir gürültüde resmin içindeki piksellerin değerleri ve türevleri değişmektedir.

Daha önce anlatıldığı gibi SIFT yönteminde farklı ölçek ve dönüşümlerde, değişen parlaklıklarda, görünüşteki küçük açı değişimlerinde, gürültü eklenmiş, karışık ve yıpranmış resimlerde değişmeyen öznitelikleri bulmak hedeflenmiştir. Genel olarak önerilen yöntemler SIFT yöntemi ile performanslarını karşılaştırmışlardır. Bunun için SIFT yönteminden daha hızlı bir algoritma geliştirmek için farklı kombinasyonlar denenmiştir. Örneğin; ORB ve FAST yöntemlerinde olduğu gibi öznitelikleri bulma işlemi için aynı yöntem kullanılırken, bu özniteliklerin etrafında betimleyici bulmak için farklı yöntemler kullanmışlardır. Genel olarak, kapalı-sınır bölgeleri, kavşak ve köşe noktalar, doğru kesişim noktaları, konturlar, üzerindeki gradyen değeri yüksek olan noktalar ya da kenarlar birer öznitelik olarak seçebilir.

Bu çalışmada öznitelik olarak eğri noktalarının koordinatları kullanılmıştır. Bir önceki aşamadan elde edilen kenar haritasından eğriler çıkartılarak öznitelik seti oluşturulmuştur. Öznitelik seti oluşturulurken farklı eğrilerin türevleri hesaplanarak gradyen değerleri bulunulmuştur. Bu gradyenlerin temel yönü, en büyük özdeğere karşılık gelen özvektör üzerine izdüşüm alınması ile elde edilmiştir. Sonra izdüşüm katsayıları normalize edilmiştir. Bu normalize edilmiş katsayılar birinci öznitelik setini oluşturmaktadır.

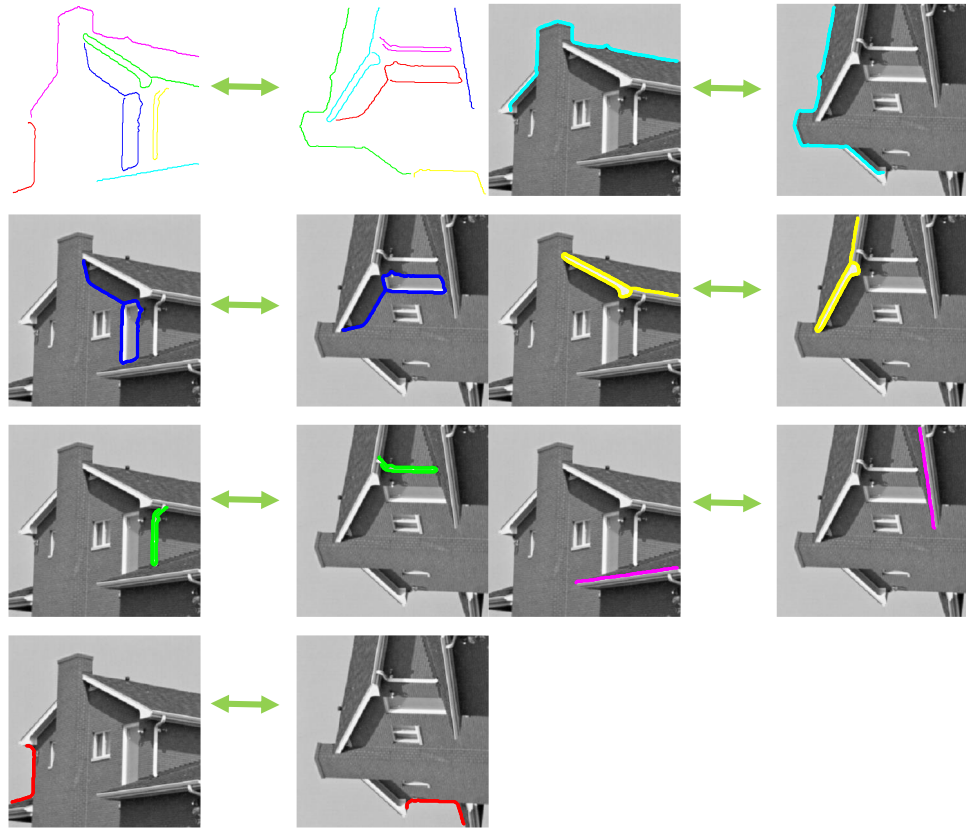
Ayrıca ikinci eşleştirme aşamasında imgelerin gradyenleri hesaplanır. Bu gradyen haritaları üzerindeki eğri noktalarından Harris öznitelikleri çıkartılarak ikinci öznitelik seti çıkartılmıştır.

### 3.3 Benzer Eğrilerin Bulunması

Bu çalışmada her aşamanın önemli bir katkısı olmasına rağmen, iki farklı resimde belirli bir yöntem ile benzer eğrilerin bulunması bu çalışmanın can damarıdır. Çünkü önerilen yöntem eğri yani öznitelik tabanlı bir yöntem olduğundan dolayı iki resmin birbirine benzerliklerinin doğru hesaplanması gerekmektedir. Bu amaç doğrultusunda, bir önceki aşamadan elde edilen öznitelik

setlerini kullanarak iki öznitelik set arasındaki uzaklık hesaplanır. Bu uzaklık eğrilerin birbirine benzeme oranını vermektedir.

İki eğri arasındaki uzaklığı hesaplamak için literatürde daha önce dönmüş noktaların karşılaştırılması için kullanılan Hausdorff uzaklığı [76-80] bu çalışmaya uyarlanmıştır. Hausdorff yöntemine göre en düşük uzaklık en iyi benzeyen anlamı taşımaktadır. Hausdorff yöntemine bağlı kalarak bir eğrinin hangi eğriye en çok benzediği tahmin edilmiştir. Aşağıda benzerliği bulunulan eğriler renkli bir şekilde gösterilmiştir. Hausdorff yönteminin çalışma prensibi takip eden başlık altında daha detaylı olarak anlatılmıştır.



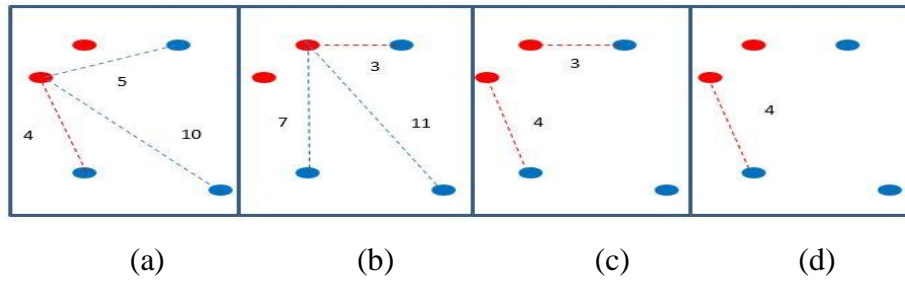
Şekil 3.3.1 Hausdorff uzaklığı ve ikinci eşleştirme ile benzeyen eğriler

Şekil 3.3.1 de gösterildiği gibi birbirine benzeyen eğriler Hausdorff uzaklığı ve ikinci eşleştirme kullanılarak bulunulmuştur. Soldaki resim referans resmi gösterirken, sağdaki resim ise referans resmin ilgin dönüşüm kullanılarak 90 derece döndürülmesini göstermektedir. Görüldüğü gibi her iki resimden de 6 tane eğri bulunulmuştur. 6 tane eğriden hepsi doğru eşleşmiştir.

Yukarda verilen eşleşen eğrileri elde etmek için ilk olarak bir öznitelik çıkarma aşamasında eğrilere ait öznitelikler elde edilmiştir. Birinci öznitelik seti eğrilerin TBA kullanılarak temel yön üzerinde izdüşümü alınıp normalize edilmesi oluşturulmuştur. İkinci öznitelik seti ise imgelerin gradyen haritası çıkartılıp ve gradyen haritasında eğri üzerindeki noktalardan Harris öznitelikleri çıkartılarak oluşturulmuştur. Birinci öznitelik seti kullanılarak iki eğri arasındaki Hausdorff uzaklığı tespit edilmekte sonra ikinci öznitelik kullanılarak eğriler eşleştirilmektedir. Böylece hiç eşleşmeyen eğriler ihmal edilerek kümede eğri sayısı düşürülmüştür. En son olarak kümede var olan eğrilerden en küçük Hausdorff uzaklığına göre birbiriyle en iyi eşleşen eğriler bulunulmuştur. Yukardaki şekilde önerilen yöntemin %100 doğru eşleşme oranı verdiği görülmektedir.

### 3.3.1 Hausdorff Uzaklığı

Adını 1868 ile 1942 yılları arasında yaşamış olan ve yöntemi öneren Felix Hausdorff'dan almıştır. Hausdorff uzaklığı iki nokta kümesini göz önünde tuttuğumuzda içinde birbirine yakın olan noktalar arasındaki en büyük uzaklıktır [76, 81]. Bu tanımın daha iyi anlaşılması için aşağıda Hausdorff uzaklık yönteminin nasıl çalıştığı bir örnek ile anlatılmıştır. Bu uzaklık daha önce obje karşılaştırma [82], iki eğri arasındaki uzaklığın ölçülmesinde [83], dönmeye karşın değişmeyen noktaların eşleştirilmesinde [76] ve resimlerde kontur çıkartılması gibi birçok çalışmada kullanılmıştır. Detaylı bir şekilde gösterimi aşağıda verilmiştir.



Şekil 3.3.1.1 Hausdorff uzaklık yönteminin çalışma prensibi

Yukarda gösterildiği gibi (a) safhasında birinci nokta ile diğer noktalar arasındaki en yakın uzaklığı hesaplar daha sonra (b) safhasında ikinci nokta ile diğer noktalar arasında en yakın uzaklığı hesaplar ve (c) safhasında en yakın uzaklıklar arasından maksimum olanını seçer. Yukarda verilen şeklin (d) safhasında iki nokta

seti arasındaki en yakın uzaklık 4 olarak seçilmiştir. Fakat bu sadece bir ön gösterimdir. Yani burada iki nokta arasındaki uzaklık için Öklid veya bir başka uzaklık metriği kullanılabilir.

Verilen referans çalışmalarda vurgulandığı gibi Hausdorff uzaklığı yönlü bir uzaklıktır. Simetrik olmayan bir uzaklıktır. Bu nedenden dolayı A nokta setinden B nokta setine uzaklık 4 iken, tersi yönde B'den A'ya farklı bir uzaklık elde edilir. Bu yüzden Hausdorff yöntemi için genel bir tanım önerilmektedir. Bu tanıma göre;

$$H(A, B) = \max(H(A, B), H(B, A)) . \quad (3.3.1.1)$$

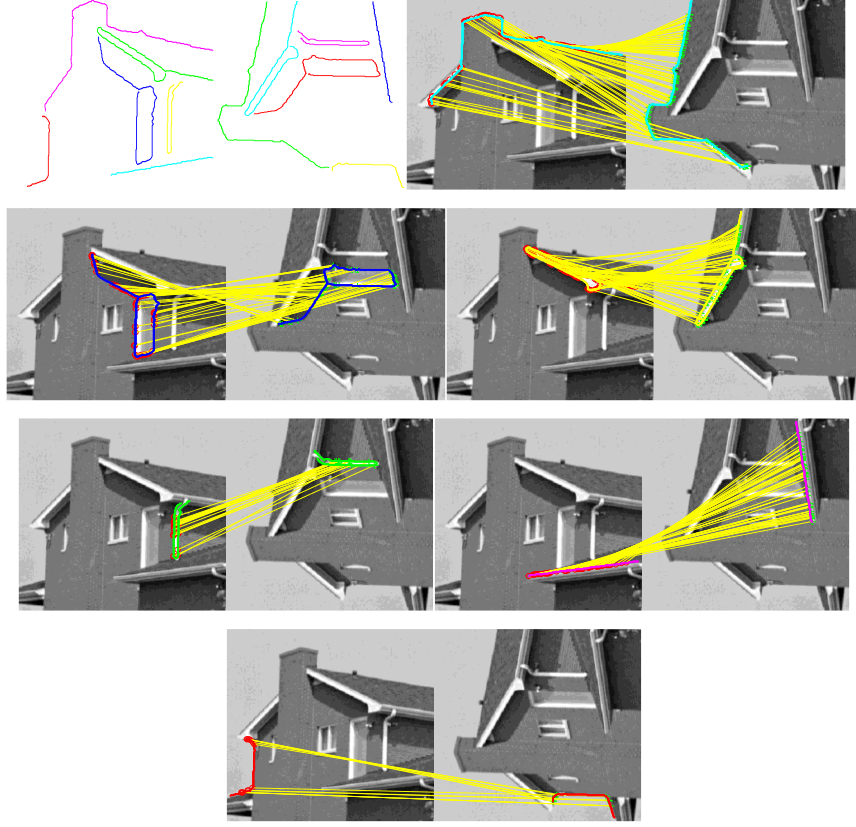
Yani A-B arası Hausdorff uzaklığı bulunur sonra B-A arası Hausdorff uzaklığı bulunarak bunlar arasından maksimum olanı seçilmektedir.

### 3.4 Öznitelik Eşleştirme

Bu aşamada amaç referans ve hedef imgede elde ettiğimiz öznitelikleri belli bir yöntem kullanarak karşılaştırıp, aynı noktaların doğru bir şekilde eşleşmesini gerçekleştirmektir. Bunu gerçekleştirmek için literatürde birçok yöntem önerilmiştir. Temel hedef dönüşüm matrisi hesaplamadan önce, eldeki özniteliklerin güçlü olup olmadığını tespit etmektir. Eğer doğru konuma sahip öznitelikler bulunursa, dönüşüm matrisini daha doğru hesaplamak kolaylaşacaktır. Bu aşamada temel hedef “t” anında alınan resimden elde edilen özniteliklerin, “t-1” anında elde edilen özniteliklerle belirli benzerlik ölçütleri kullanarak eşleştirilme işlemidir.

Şekil 3.4.1’de iki eğri üzerinde eşleşen noktaları görülmektedir. Bu eşleşen noktalar bizim çalışmamızın hayat sularıdır. Bu eşleşen noktalar sayesinde resmin ilgin dönüşümü hesaplanacaktır. Yani bu noktalar kullanılarak resmi döndürme işlemini yapacağız.

Şekilden de net olarak görüleceği gibi tüm eğrilerin doğru eşleştiği görülmektedir. Bu çalışmanın performansında Hausdorff metriği ve Harris noktalarının katkısı oldukça büyüktür. Ayrıca iki resimdeki ilgin dönüşümün hesaplanması için sadece 3 tane noktanın yeterli olacağı göz önünde tutulursa Hausdorff yönteminin başarısı takdir edilmeye değerdir.



**Şekil 3.4.1** İki farklı eğri üzerinde eşleşen öznitelikler

Yukardaki şekilde Hausdorff uzaklığı ve ikinci eşleştirme sonucu birbirine eşleşmiş eğriler sergilenmektedir. Bu çalışmada eşleştirme yapmak için imgelerin gradyen değerleri bulunur ve elde edilen gradyen haritaları üzerindeki eğri noktalarından Harris öznitelikleri çıkartılmaktadır. Sonra Matlab programının hazır fonksiyonları kullanılarak yukarda gösterildiği gibi hangi noktanın hangi noktayla eşleştiği gösterilmiştir.

### 3.5 Geometrik Dönüşümün Hesaplanması ve Görüntü Çakıştırma

Görüntü çakıştırma için gerekli olan en önemli aşama benzer görüntüler arasındaki dönüşüm matrisini bulmaktır. Birçok dönüşüm matrisi olmasına rağmen

bu çalışmanın ilgilendiği matris ilgin dönüşüm matrisidir. Elimizde eşleşen öznitelikler olduğundan dolayı iki resim arasındaki ilgin dönüşüm matrisinin bulunması kolay olmaktadır. Bir önceki aşamalardan en iyi eşleşen eğriler bulunulmuştur. Bu eğriler üzerindeki Harris öznitelikleri kullanılarak geometrik dönüşüm matris elde edilmektedir. Bunun için Matlab programında hazır bulunan *estimateGeometricTransform* fonksiyonu kullanılmıştır. Kullanılan metot RANSAC [84] metriği kullanmaktadır. Bu çalışmada *estimateGeometricTransform* fonksiyonun varsayılan parametreleri kullanılmaktadır.

### 3.5.1 İlgin Dönüşüm

İki resimden de eşleşen kontrol noktaları bulduktan sonra hedef resmin geometrik olarak ne kadar döndüğünün hesaplanması gerekir. Yani yaptığımız çalışmaya göre dönüşümün tipini belirlememiz gerekir. Genel olarak resmi döndürmek literatürde kullanılan dönüşümler [85] aşağıda verilmiştir:

- Benzerlik.
- İlgin.
- İzdüşümlü.
- Polinomsal.
- Doğrusal.
- Yerel ağırlıklandırılmış ortalama (YAO) .

Bu çalışmada ilgin dönüşüm kullanılarak görüntülerin iki görüntü arasındaki geometrik dönüşüm hesaplanmıştır. İlgin dönüşümde dönme (rotation), öteleme (translation), ölçeklenme (zoom ya da scaling) ve kayma (shear) olmak üzere sadece dört durum söz konusudur.

$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, S = \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & hx & 0 \\ hy & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5.1.1)$$

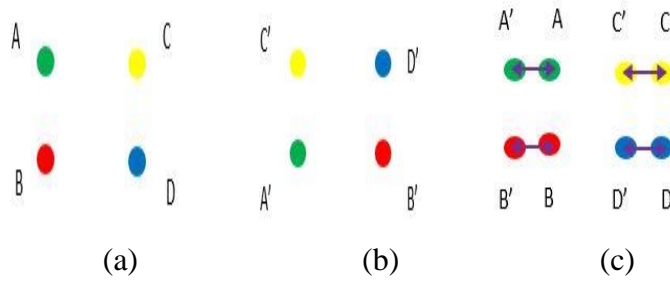
Yukardaki verilen matrislerden R dönmeyi (rotation), S ölçeklenmeyi (scaling), T kaymayı (translation) ve H kaymayı (shear) ifade etmektedir. Eğer sadece dönme ve öteleme varsa ilgin dönüşüm aşağıdaki gibi olur.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & tx \\ \sin \theta & \cos \theta & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.5.1.2)$$

Bu aşamada, iki tane eşleşen nokta setleri kullanılarak, iki resim arasındaki ilgin dönüşüm hesaplanmaya çalışıldı.

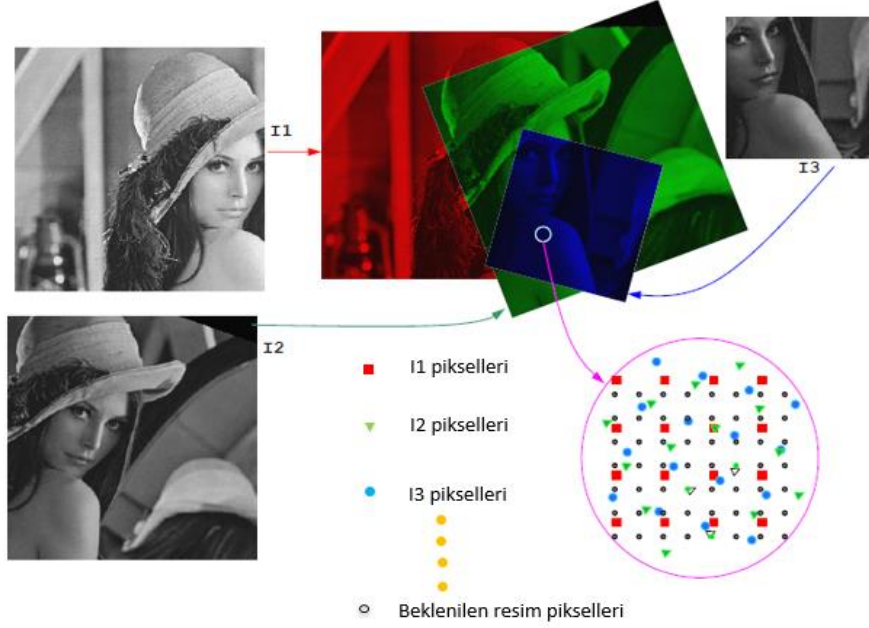
### 3.5.2 RANSAC

RANSAC yöntemi benzer nokta kümelerinin arasından en doğru eşleşenleri ortaya çıkaran bir yöntemdir. Örnek olarak; Şekil 3.5.2.1 de elimizde aşağıdaki gibi 4 tane nokta (a) ve bu noktaların 90 derecelik dönmüş koordinatları (b) verilsin. RANSAC yöntemi A, B ve C olarak noktasını A', B' ve C' noktaları ile belli bir metriğe bağlı olarak eşleştirir. Daha sonra bu noktalardan elde ettiği dönüşüm matrisini kullanarak resmi Şekil 3.5.2.1 (c)'de olduğu gibi üst üste getirir. Üst üste getirdikten sonra koordinat hatalarını yani gerçekten üst üste gelip gelmediğini kontrol eder. Bundan sonra rastgele diğer noktalar arasından üç tane nokta daha seçer ve başka hatalar elde eder. Tüm noktaları denedikten sonra en az hata veya en fazla doğruluk veren dönüşüm matrisini hesaplar. Böylece belli bir eşik değerine bağlı olarak aykırı noktalar elenmektedir.



Şekil 3.5.2.1 RANSAC çalışma prensibi

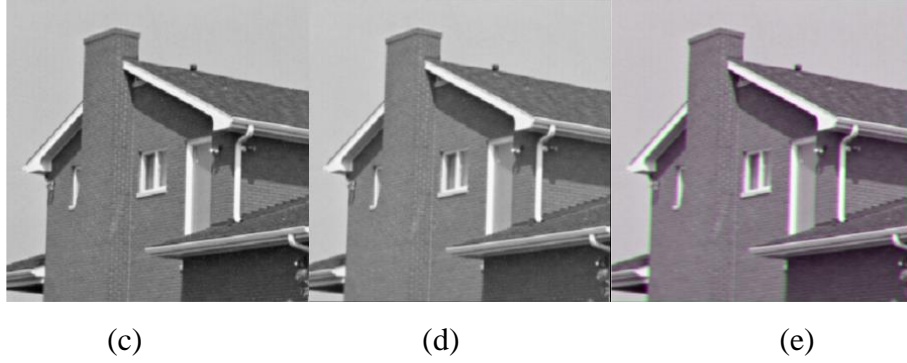
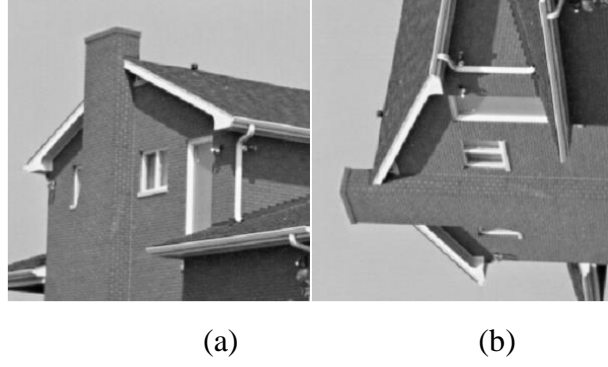
### 3.5.3 Görüntü Çakıştırma



Şekil 3.5.3.1 Görüntü çakıştırma genel görünüm

Şekil 3.5.3.1’de görüldüğü gibi bir görüntü referans olarak alınır ve diğer görüntülerin bu görüntüye göre dönüşümleri hesaplanır [16]. Dönüşüm yapıldıktan sonra referans görüntü koordinatları ile diğer görüntülerin koordinatlarının birbirine yakınlaştığı görülmektedir. Koordinatlar eğer tam üst üste gelir ise sıfır hata oranı ile görüntüleri çakıştırma işlemi gerçekleştirilmiş olacaktır.

Görüntü çakıştırma aslında görüntünün üst üste getirilmesi anlamına gelmektedir. Basit olarak üst üste gelen iki görüntünün piksellerin ortalamasını alınarak görüntüler çakıştırılabilir. Ayrıca ortak bilgi yöntemi, ortalama hata kareler veya normalize edilmiş çapraz korelasyon gibi metrikler kullanılabilir [86]. Ortak bilgi yönteminde bir imge diğeri üzerinde gezdirilmekte ve artan entropi değerine göre karar verilir. Bunun gibi yapılan çalışmalar [87] de verilen çalışmada mevcuttur.



Şekil 3.5.3.2 Döndürülmüş iki ev resminin çakıştırılması

Bu çalışmada ise Matlab programının hazır görüntü çakıştırma fonksiyonlarından yararlanarak görüntü çakıştırma işlemi gerçekleştirilmiştir. Matlab programında görüntü çakıştırma için 4 farklı metrik önermektedir. Bunlar ortak bilgi yöntemi (*mutual information*), ortalama hata kareleri (*Mean Square Errors*), düzenli adım gradyen iniş optimize edici (*Regular Step Gradient Descent Optimizer*) ve bir artı bir evrimsel optimize edici (*One-plus-one evolutionary optimizer*) olarak verilmektedir. Bu çalışmada en optimal metriği seçmek için *imregconfig* fonksiyonu kullanılmaktadır. Daha sonra hazır fonksiyonlar kullanılarak çakıştırma işlemi gerçekleştirilmiştir.

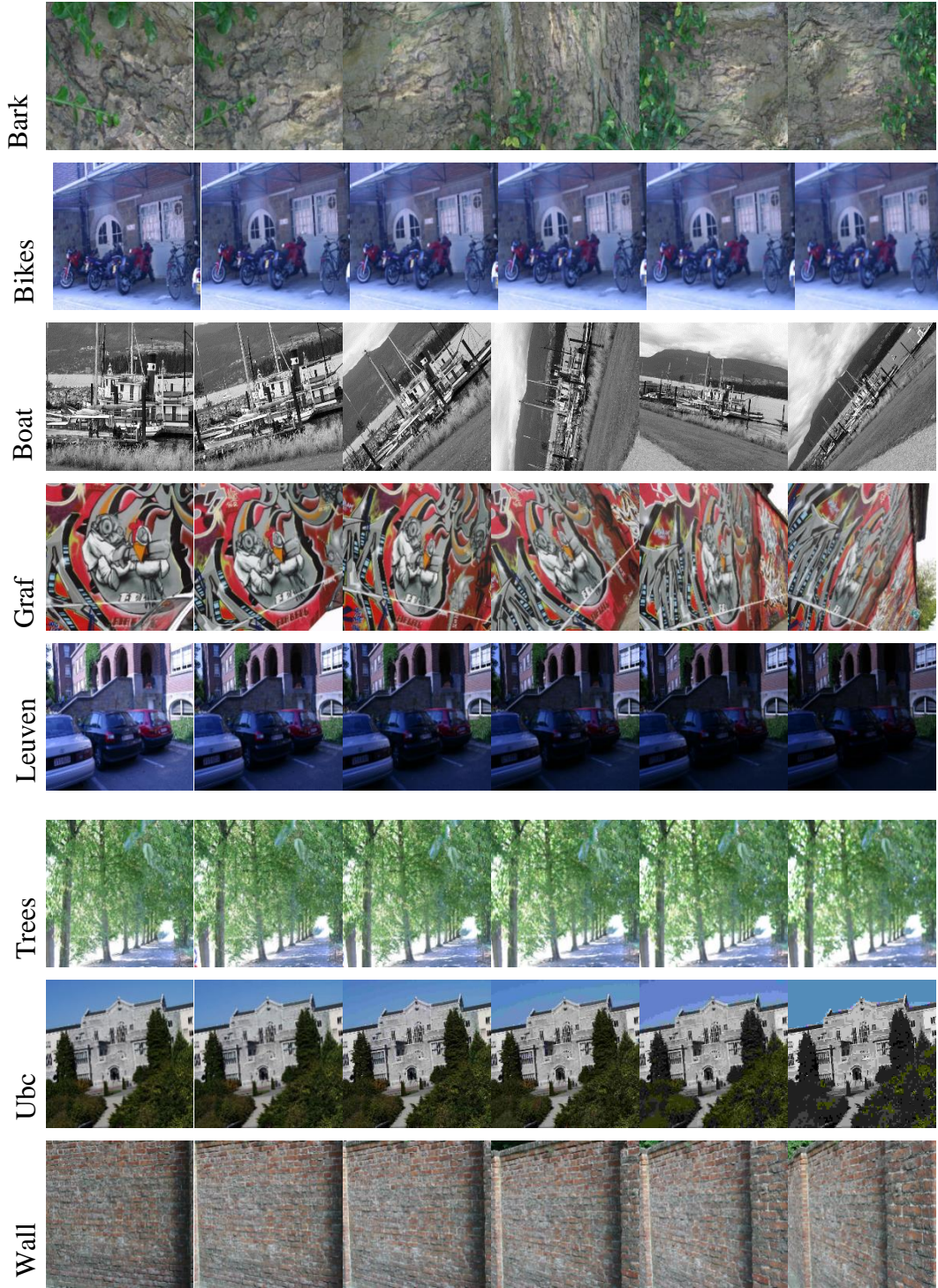
Şekil 3.5.3.2'de bir görüntü çakıştırmaya örnek verilmiştir. Görüldüğü gibi birinci resim 90 derece döndürülmüştür (a-b). Aradaki dönüşümü bulmak için nokta sayısı temel alınarak en iyi eşleşen iki eğri üzerindeki noktalar kullanılarak ilgin dönüşüm matrisi parametreleri hesaplanmıştır (c-d). Elde edilen ilgin dönüşüm matrisi dönen resmin (b) koordinatlarına uygulanarak (d)'de gösterildiği gibi yeniden döndürülmüştür. Böylece (e)'de gösterildiği gibi iki resmi üst üste rahatlıkla getirilebilir. Burada önemli olan nokta ise elde edilen ilgin dönüşüm matrisinin parametrelerinin doğruluğudur.

## 4 HESAPSAL SONUÇLAR

Bu tez kapsamında kendi yöntemimizin başarısını diğer yöntemler ile ölçmek için literatürde var olan ve en iyi bilinen öznitelik detektörlerin (feature detectors) ve öznitelik betimleyicilerin (feature descriptors) performansı ile karşılaştırılmıştır. SIFT ile SIFT, SURF ile SURF, MSER ile SIFT, BRISK ile FREAK, ORB ile ORB ve FAST ile BRIEF betimleyicileri kullanılarak iki farklı görüntünün eşleştirme performansı hesaplanmıştır. Çünkü yanlış eşleştirme yanlış çakıştırılma olasılığını artırmaktadır. Bundan dolayı çok iyi görüntü eşleştirme yöntemi aynı zamanda çok iyi bir görüntü çakıştırma yöntemi anlamına gelmektedir. Resmin ne kadar döndüğünü hesaplamak ise bütün yöntemler için RANSAC kullanılabilir.

### 4.1 Veritabanı

Öznitelik çıkarıcı ve betimleyicilerin performansını ölçmek için birçok veritabanı olmasına rağmen, Mikolajczyk ve Schmid tarafından önerilen ve en iyi bilinen veritabanını kullanılmaktadır [88]. Şekil 4.1.1’de verildiği gibi deney resimleri 8 sınıftan oluşmaktadır [89]. Bu sınıflar *bark*, *bikes*, *boat*, *graf*, *leuven*, *trees*, *ubc* ve *wall* olarak adlandırılmıştır. Her bir sınıf 6 resimden ve 5 tane Homograf (Homograph) matrisinden oluşmaktadır. Bu veritabanını seçmemizin nedeni birçok deformasyon türünü barındırıyor olmasıdır. Kullanılan veritabanı; rotasyon (rotation), zum yapma (zoom), bulanıklık (blur), ışık değişimleri (light changes or illumination), görüş açısı değişikliği (viewpoint changes) ve JPEG sıkıştırma gibi bozulmalar içermektedir.



Şekil 4.1.1 Deney resimleri

## 4.2 Karşılaştırma Metriği

Her bir yöntemin performansını ölçmek için [88]'de önerilen kesinlik (precision) ve duyarlılık (recall) ölçütleri kullanılmaktadır.

Bu çalışmada duyarlılık; iki resimde doğru eşleşen betimleyicilerin sayısının, eşleşen (doğru veya yanlış eşleşenlerin toplamı) betimleyicilerin sayısına oranıdır. Burada eşleşenler *correspondences* ve doğru eşleşenler *correct matches* şeklinde tanımlanırsa, duyarlılık (recall) değeri aşağıdaki gibi elde edilir.

$$Recall = \frac{\# correct matches}{\# correspondences} \quad (4.2.1)$$

Burada doğru eşleşenlerin sayısı (# correct matches), RANSAC çalıştırılıp aykırı betimleyicilerin yok edilip aykırı olmayan betimleyicilerin alınmasını ifade etmektedir. RANSAC yönteminden önce bir çapraz kontrol (cross check) işlemi uygulanmıştır. Yine eşleşenleri sayısı (# correspondences) referans resimde olan anahtar noktaların dönüşümden sonrada ikinci resimde halen var olan anahtar noktalarını ifade etmektedir. Bu çalışmada eşleşen sayısını bulmak için Brute-Force ve Flann-Based eşleştiriciler ile birlikte Hamming ve L-2 norm uzaklık metrikleri kullanıldı. SIFT ve SURF betimleyicileri için Flann-Based eşleştirici ile birlikte Hamming uzaklığı, ikili betimleyiciler olan ORB, BRISK ve FREAK için Brute-Force eşleştirici ile birlikte Hamming uzaklığı kullanılmıştır. Sadece BRIEF betimleyicisi için Brute-Force eşleştirici ile birlikte L-2 norm kullanılmıştır. Yine burada MSER ve FAST yöntemleri betimleyici çıkarmadıkları ve sadece anahtar nokta çıkararak yöntemler oldukları için SIFT ve SURF betimleyicileri ile beraber çalıştırıldı.

İlaveten kesinlik (precision) değeri ise RANSAC sonrası elde edilen eşleşen sayısının doğru eşleşen sayısına oranını ifade etmektedir. Burada en yüksek kesinlik değeri demek eşleşen noktaların birbiriyle ilişkisinin çok olduğunun göstergesidir.

$$Precision = \frac{\# correct matches}{\# total positive matches} \quad (4.2.2)$$

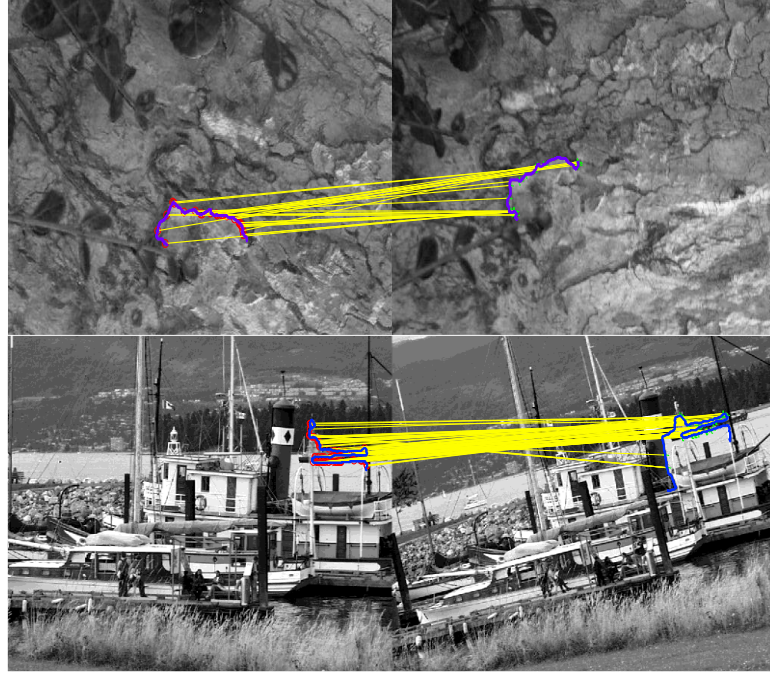
Burada dikkat edilmesi gereken nokta kesinlik ve duyarlılık değerleri seçilen eşleştirme yönteminin katılığına göre değişebilmektedir. Bu yüzde bizde adaletli olması için yukardaki gibi mümkün olduğunca her şeyi dengelemeye çalıştık.

Önerilen çalışmanın kesinlik ve duyarlılık değerlerini hesaplarırken nokta yerine eğri eşleştirme temel alınmaktadır. Buna göre doğru veya yanlış eşleşen eğrilerin toplamı *correspondences* sayısını, 0,6 eşik değerinden büyük olacak şekilde eşleşen eğrilerin sayısı *correctMatches* ve son olarak transform matrisi ile

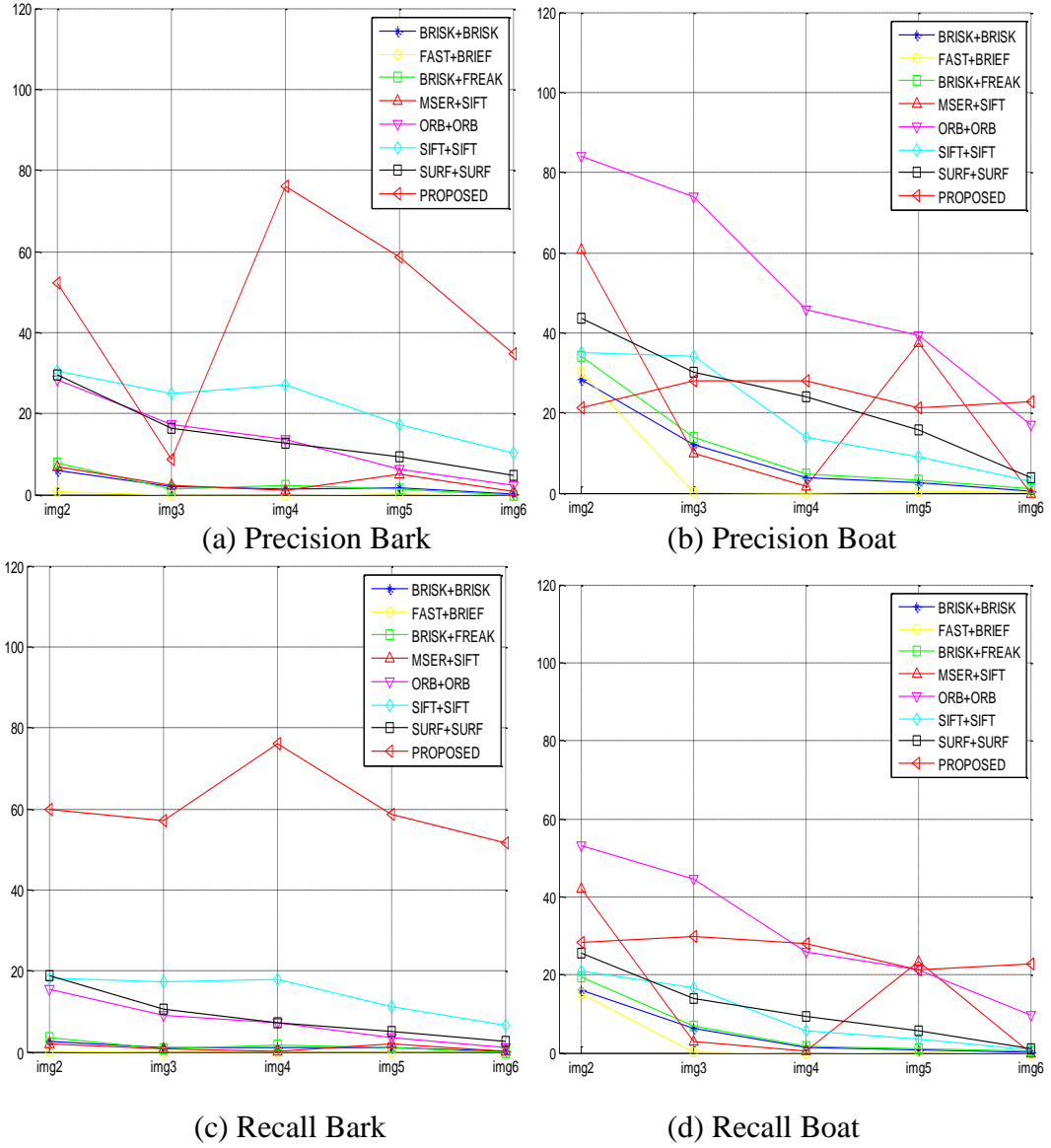
çarpıldığı halde resmin boyutunun dışına çıkmayan eğrilerin toplam sayısı ise bize *total positive matches* oranını vermektedir.

### 4.3 Rotasyon (Rotation) ve Zum (Zoom) Değişimi Etkileri

Önerilen yöntemin rotasyon ve zum değişimindeki performansını ölçmek için *bark* ve *boat* sınıfları üzerinde denemeler yapılarak kesinlik ve duyarlılık sonuçları Şekil 4.3.2 (a-d)'de gösterildi. Bu sınıflarda kullanılan resimler 30 ve 45 derecelik açılarla döndürülmüştür. Ayrıca resimler 25% oranında zum yapılmıştır. İlk bakışta görüleceği gibi, ORB, SIFT ve SURF birbiriyle yarışacak sonuçlar vermişlerdir. *Bark* resimleri için SIFT, ORB yöntemini geçerken, fakat *boat* resimleri için ORB'nin performansı SIFT'ten daha yüksektir. Kolayca ORB ve bizim (PROPOSED) yöntemlerinin diğerlerini geçtiğini söyleyebiliriz. Yine FAST ile BRIEF yönteminin en kötü olduğunu söyleyebiliriz. Görsel sonuçlar ise Şekil 4.3.1'de verilmiştir.



Şekil 4.3.1 Bark ve Baot resimleri üzerindeki sonuçlar



Şekil 4.3.2 Zum ve Rotasyon için Precision ve recall değerleri

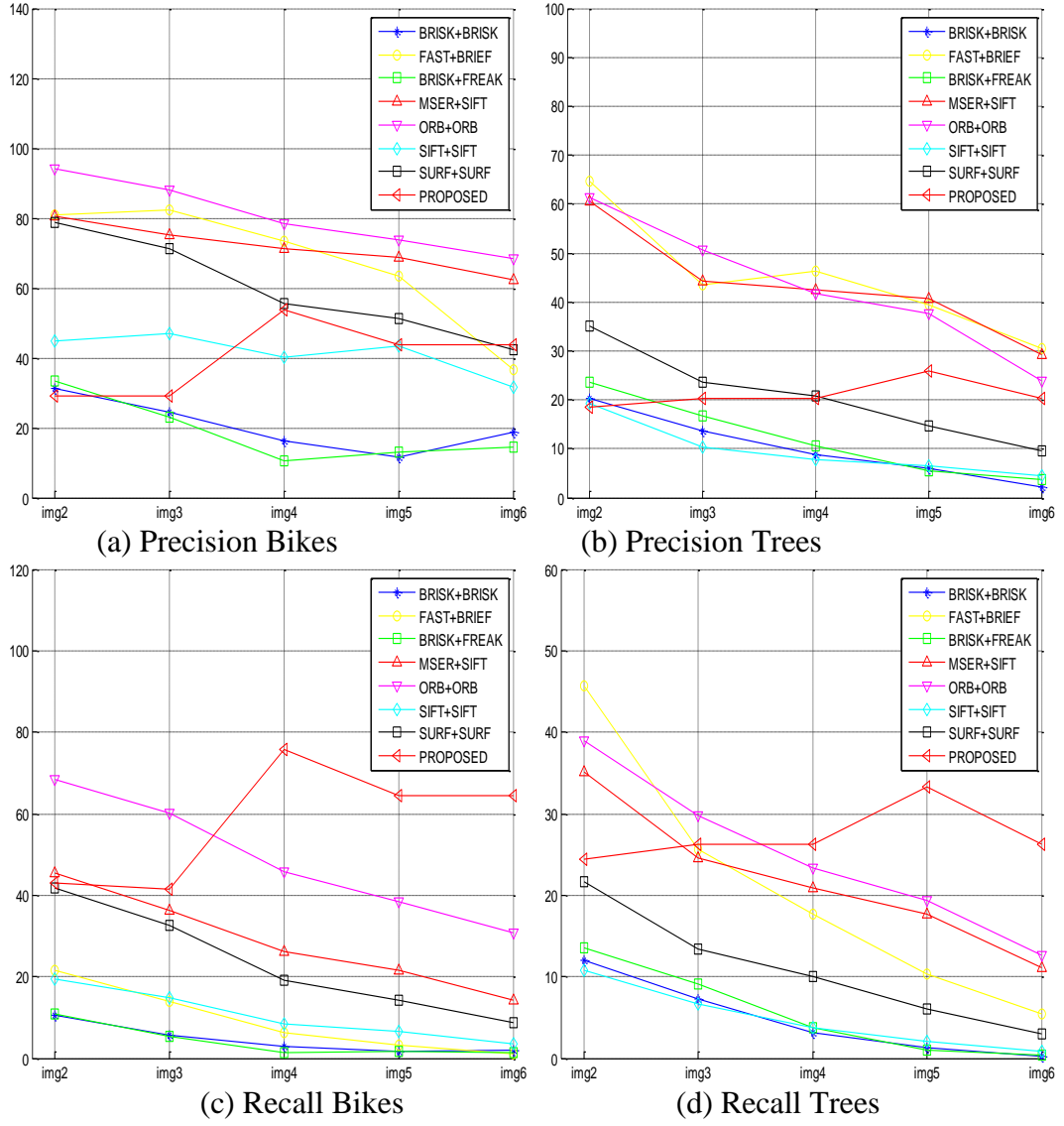
#### 4.4 Bulanıklık (Blur) Değişimi Etkileri

İkinci deney ise *bikes* ve *trees* deney setleri üzerinde yapılmıştır. Resimler yarıçapı 2-2,5 arasında değişecek şekilde bulanıklık ekleyerek farklı ölçekler elde edilmiştir. Artan bulanıklık seviyesine göre gözlemlediğimizde *bikes* resimlerinin *trees* resimlerine göre ana iskeletlerini korudukları gözlemlenmiştir. *Bikes*

resimlerinden elde edilen anahtar nokta sayısının *trees* resimlerinden daha fazla olduğu gözlemlendi. Şekil 4.4.2 (a-d) de verilen sonuçlara bakıldığında, biker resimleri üzerinde yapılan çalışmada ORB'nin MSER ile SIFT betimleyicileri ve FAST ile BRIEF betimleyicilerden daha iyi performans sergiledikleri görülecektir. Yine *trees* resimleri için MSER ile SIFT betimleyicileri ve FAST ile BRIEF betimleyicilerinin bulanıklık seviyesi artmasına rağmen iyi sonuçlar verdiği gözlemlenmiştir. İlginç bir gözlem olarak, SIFT her ne kadar rotasyon, ölçek ve parlaklık değişimi durumlarda iyi sonuçlar vermesine rağmen, yüksek bulanıklıkta krallığını başka yöntemlere kaptırmıştır. Bizim çalışma (PROPOSED) ise kesinlik değerleri için ortalama bir performans sergilerken, duyarlılık değerlerinde yüksek bulanıklık değerlerinde tüm yöntemlere göre iyi sonuç vermiştir. Yine Şekil 4.4.1'de görsel sonuçlar verilmiştir.



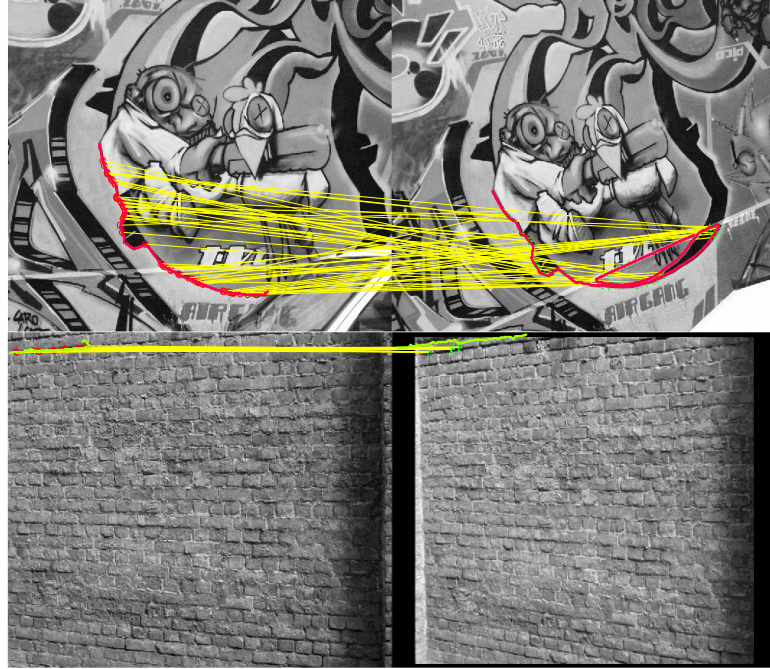
Şekil 4.4.1 Bikes ve Trees resimleri üzerindeki sonuçlar



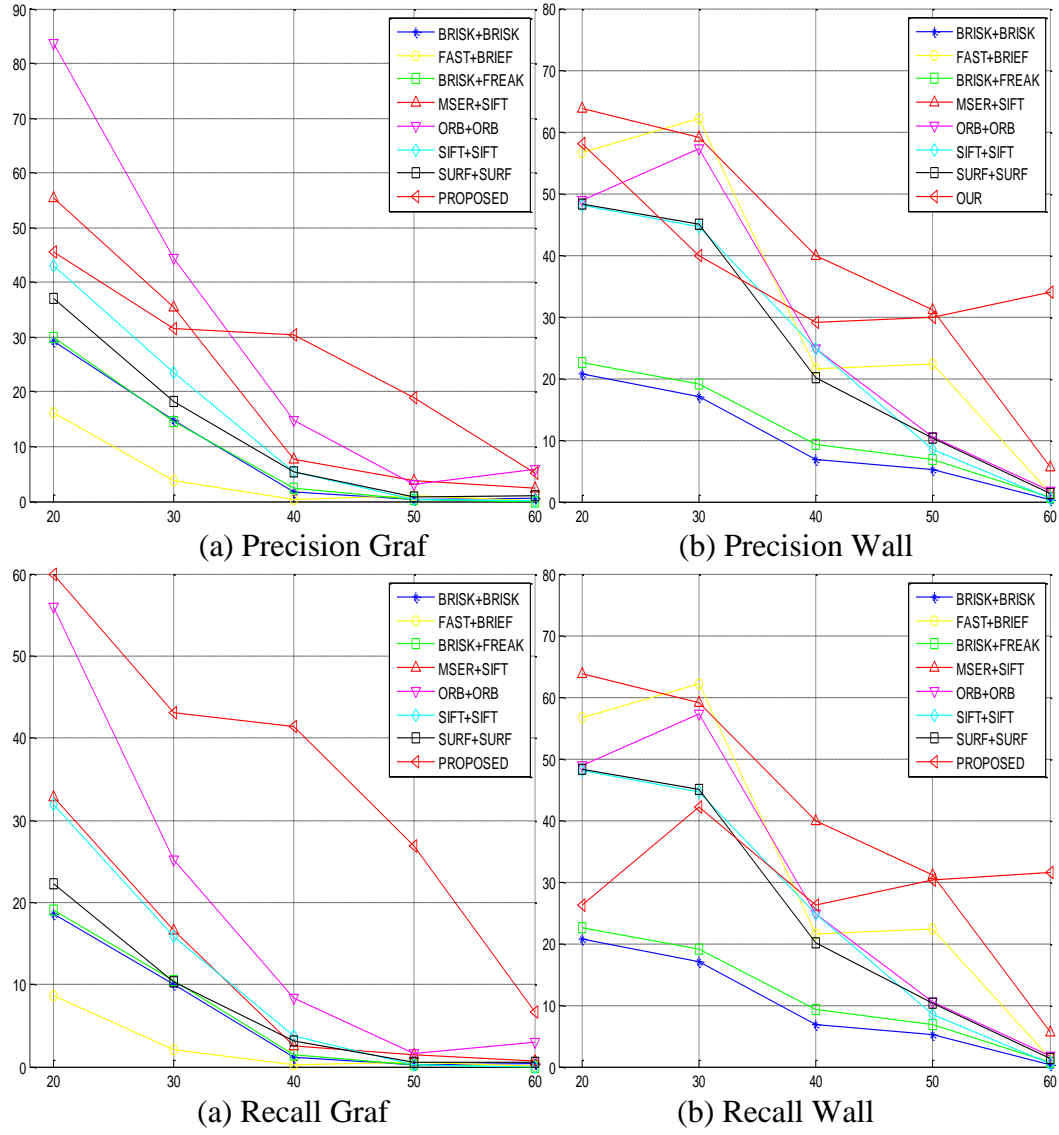
Şekil 4.4.2 Bulanıklık için precision ve recall değerleri

#### 4.5 Görüş Açısı (Viewpoint) Değişimi Etkileri

Üçüncü bir deney olarak, bizim yöntemin sonuçları ile diğer isimleri verilen yöntemlerin sonuçları görüş açının değiştirilmiş olduğu *Wall* ve *graf* sınıflarındaki resimler üzerinde denemeler yapılarak karşılaştırılmıştır. Görüş açısı değişimi vermek için *fronto-paralell* görüş yöntemi ile 20-60 dereceler küçültme yapılarak hesaplanmaktadır. Yapısal sahneler ve kenarlar bakımından zengin oldukları için *wall* ve *graf* veri tabanları ilgin dönüşümün analizi için çok büyük katkıya sahiptirler. Şekil 4.5.2’de gösterildiği gibi, MSER ile SIFT betimleyicileri, ORB, SIFT ve FAST ile BRIEF betimleyicilerinin kesinlik ve duyarlılık değerleri öne gelmektedir. Graf resimleri için ORB yöntemi iyi sonuçlar verirken wall resimleri için MSER ile SIFT betimleyicilerin daha baskın olduğu görülmektedir. Yine SIFT, FAST ile BRIEF betimleyicileri ve BRISK betimleyicilerin sonuçlarının birbirine karıştığı gözlenmiştir. Bizim yöntemimizin (PROPOSED) görüşü açısı değişiminden diğerlerine göre az etkilendiği açıkça görülmektedir. Şekil 4.5.1’de birkaç görsel sonuç verilmiştir.



Şekil 4.5.1 Graf ve Wall resimleri üzerindeki sonuçlar



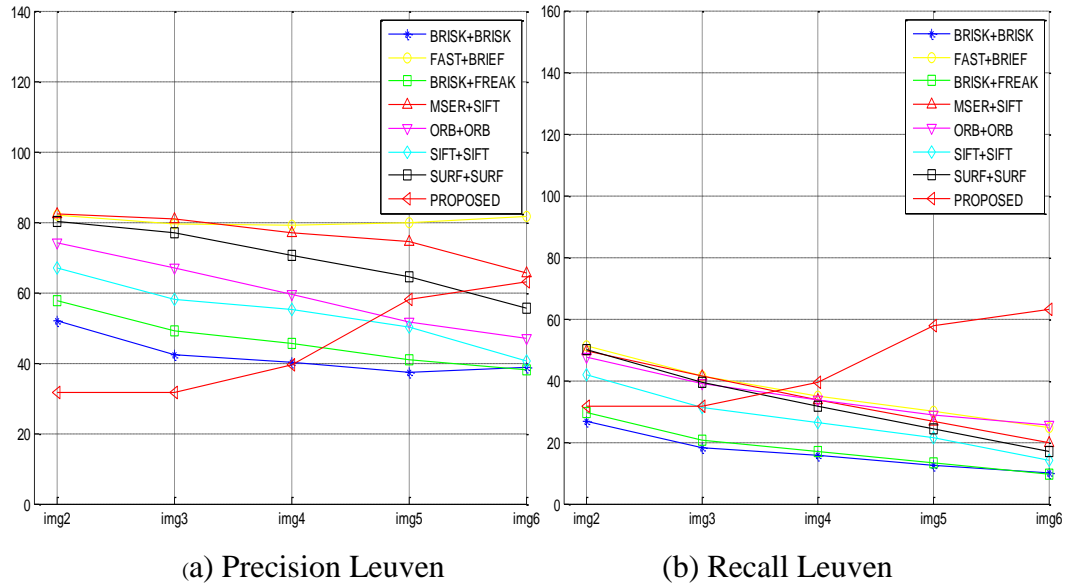
Şekil 4.5.2 Görüş açısı değişimi için precision ve recall değerleri

#### 4.6 Işık Şiddeti (Illumination) Değişimi Etkileri

Artan ışık şiddetinin değişimine göre çalışmamızın performansı nasıl etkilendiğini bulmak için *leuven* veri tabanı üzerinde denemeler yapılmıştır. Burada verilen resimler kamera deliğinin ayarı değiştirilerek oluşturulmuştur. Şekil 4.6.2 tüm çalışmaların performansını sergilemektedir. Sonuçlar göz önünü alındığında, FAST ile BRIEF betimleyicilerini kullanmak diğerlerine göre üstün sonuçlar vermiştir. Artan parlaklık seviyesine göre, SIFT ve BRISK betimleyicilerin performansında diğerlerine göre kötüleşme olduğu görülmüştür. Bunun nedeni SIFT ve BRISK betimleyicilerin karakteristik yapılarına verilebilir. Açık olarak görülüyor ki, artan parlaklık değerinde performanstaki azalma tüm yöntemler için benzer olmamaktadır. Bizim çalışmaya baktığımızda ise gittikçe artan bir performans sergilemiştir. Ayrıca Şekil 4.6.1 'de görsel sonuçlara bakıldığında önerilen yöntemin performansı daha açık olarak görülmektedir.



Şekil 4.6.1 Leuven resimleri üzerindeki sonuçlar

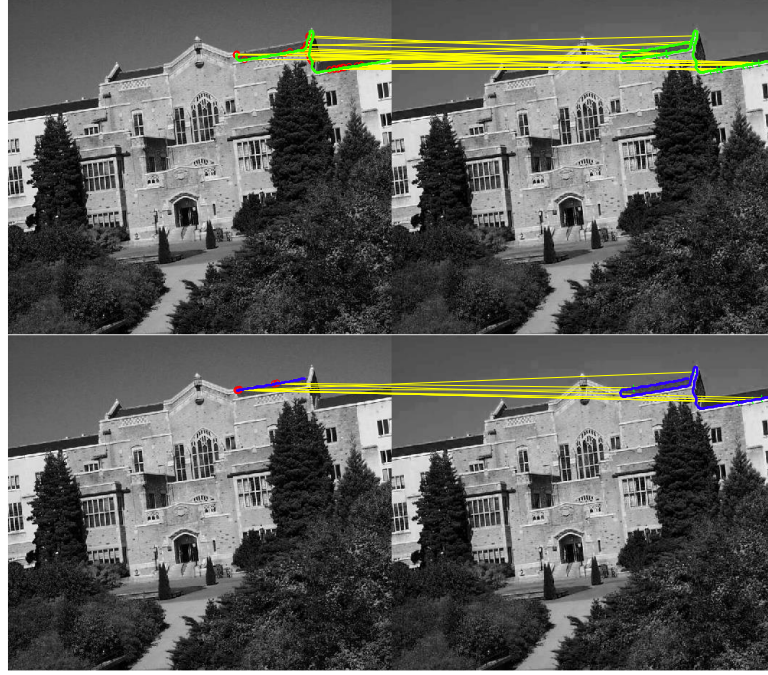


Şekil 4.6.2 Parlaklık değişimi için precision ve recall değerleri

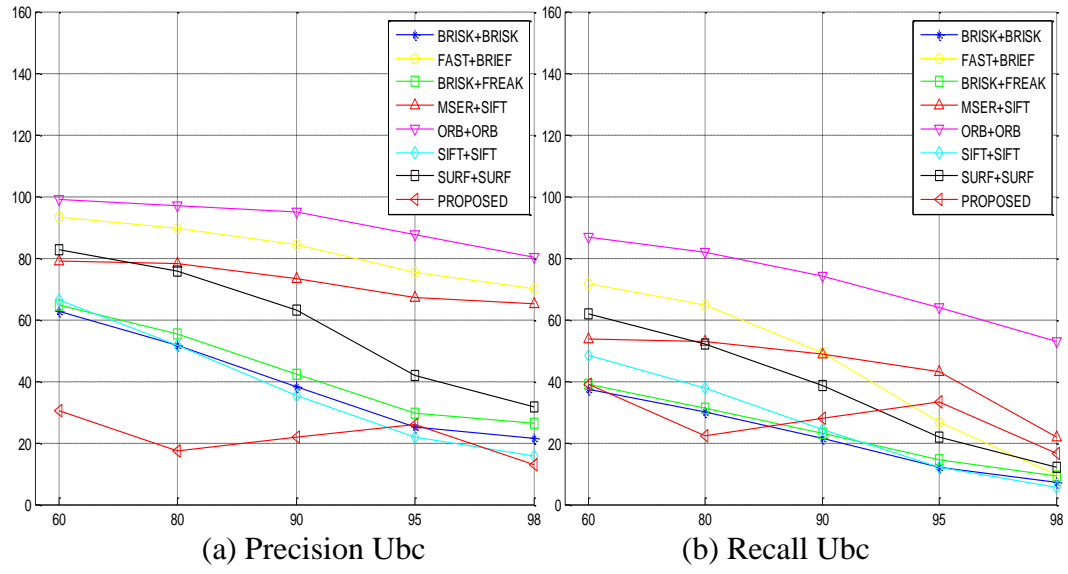
#### 4.7 JPEG Sıkıştırma (JPEG Compression) Değişimi Etkileri

Bu deneyde ise JPEG sıkıştırma yapılmış deney resimleri kullanılarak öznel ve betimleyici tabanlı görüntü eşleştiren yöntemler ile bizim yöntemin performansı karşılaştırıldı. Burada verilen resimler xv resimlerine JPEG sıkıştırma kullanılarak yapay dokuların eklenmesiyle resim kalitesi 40% ile 2% arasında

değişecek şekilde elde edilmiştir. Şekil 4.7.2 (a-b) verilen şekiller incelendiğinde, x-koordinatında 60 ile 98 arasında değişecek şekilde sıkıştırma yapıldığı görülecektir. ORB'nin artan sıkıştırma oranlarında çok yüksek sonuçlar verdiği gözükmemektedir. Bütün yöntemlerin performansı büyükten küçüğe sıralandığında FAST ile BRIEF yönteminin ikinci sırayı alacağı görülmektedir. Bizim yöntem eğri tabanlı çalıştığından dolayı performansının düştüğü görülmektedir. Örnek olarak birkaç görsel sonuç Şekil 4.7.1'de verilmiştir.



Şekil 4.7.1 Ubc resimleri üzerindeki sonuçlar

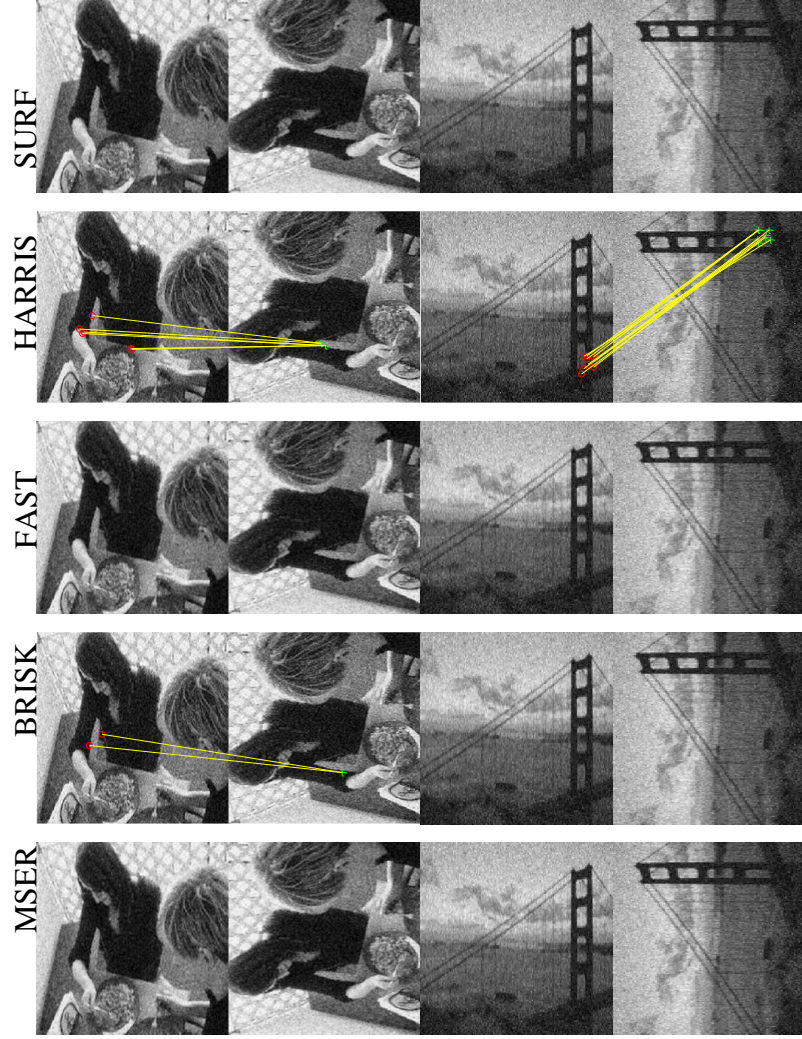


Şekil 4.7.2 JPEG sıkıştırma değişimi için precision ve recall değerleri

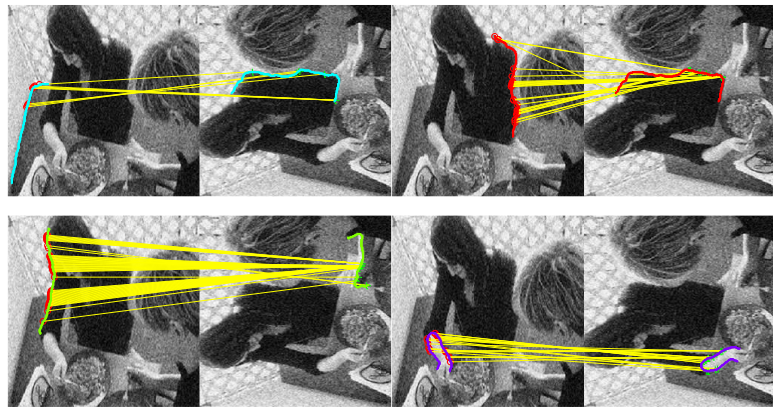
#### 4.8 Gürültülü (Noise) Değişimi Etkileri

Önerilen çalışmanın performansını diğer yöntemler ile karşılaştırmak için yapay olarak gürültü eklenmiş resimler oluşturulmuştur. Burada kullanılan resimler [75]'da verilen yayının web sitesinden indirilmiştir. Resimlere 50db's Gauss gürültüsü eklenmiştir. Şekil 4.8.1 de yüksek gürültüde eşleşmiş SURF, MSER, FAST, HARRIS, BRISK özneliklerinden eşleşenler ve bizim çalışmamızın eşleşmiş noktaları verilmiştir. Şekil 4.8.2'den Şekil 4.8.9'a kadar olan şekiller için en küçük kenar büyüklüğü 100 olarak seçilmiştir. Kenarlar bulunmadan önce Bölüm 3.1.1 de anlatılan sanatsal kenar ve köşe koruyarak yumuşatma operatörü kullanılarak resimler yumuşatılmıştır. Burada kullanılan sanatsal kenar ve köşe koruyarak yumuşatma operatörünün parametrelerinden yumuşatma için kullanılan  $\Sigma$  3, ilgi alanı boyutu için kullanılan *SectorNumber* değeri 4 olarak ve kenar keskinliği için kullanılan *EdgeSharpness* değeri 1 olarak alınmıştır. Daha sonra sırasıyla değişik kenar çıkarma yöntemleri kullanılarak görüntülerden kenarlar çıkartılmıştır.

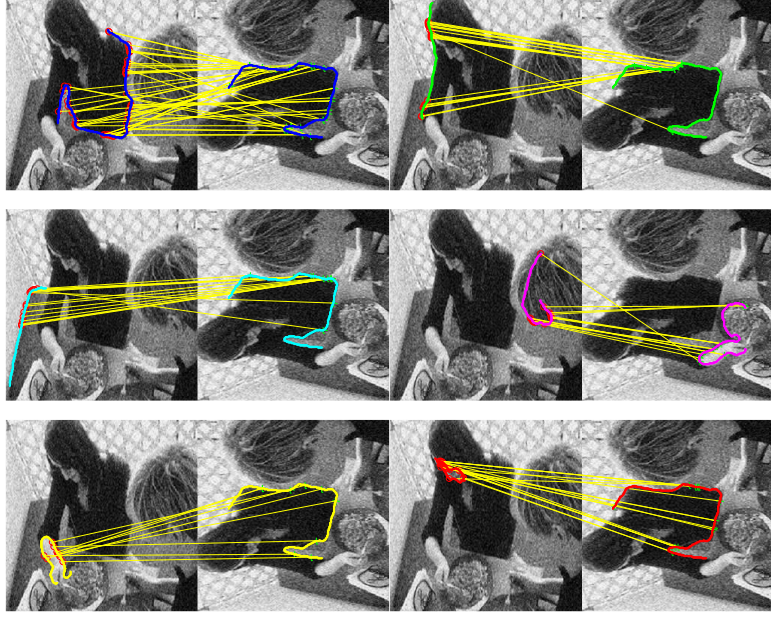
Şekil 4.8.2'de gösterildiği gibi SURF, MSER, FAST, HARRIS, BRISK yöntemleri kullanıldığında hiçbir noktanın eşleşmediği görülmektedir. Bunun yanında bizim kullandığımız kenar tanıma yöntemlerinin başarısı olarak en azından bir tane kenarın doğru eşleştiği görülmüştür. Çok yüksek gürültü durumlarında önerilen yöntem çok iyi performans sergilemiştir.



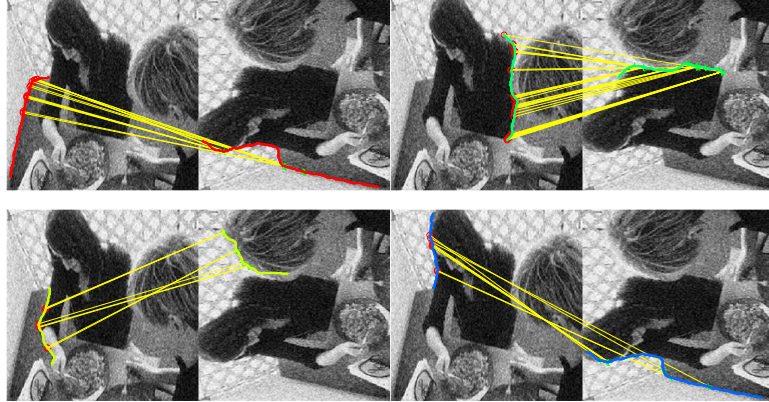
Şekil 4.8.1 SURF, MSER, FAST, HARRIS ve BRISK sonuçlar



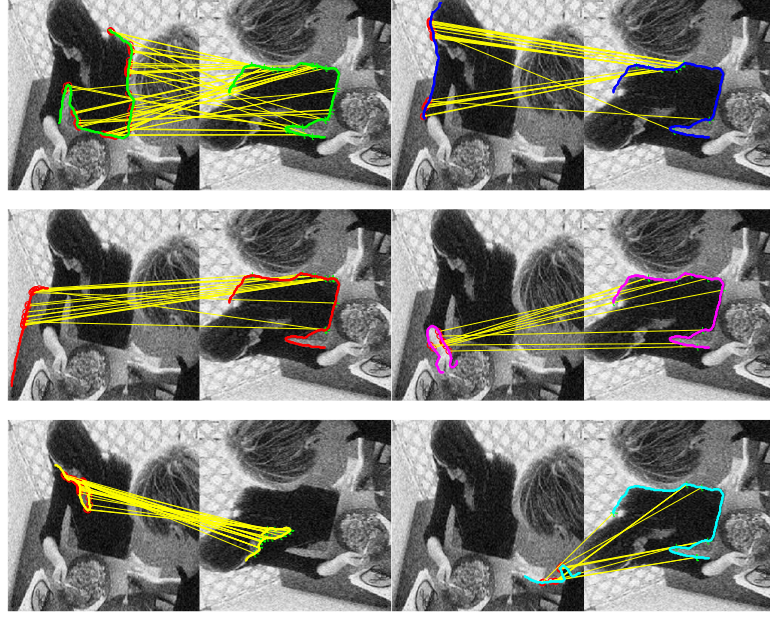
Şekil 4.8.2 Birinci resim üzerinde MSME-CVAED kenar çıkarıcı sonuçları



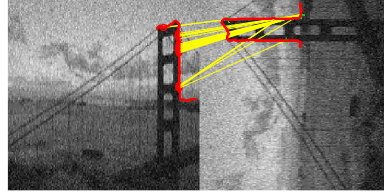
Şekil 4.8.3 Birinci resim üzerinde LPGPCA-ED kenar çıkarıcı sonuçları



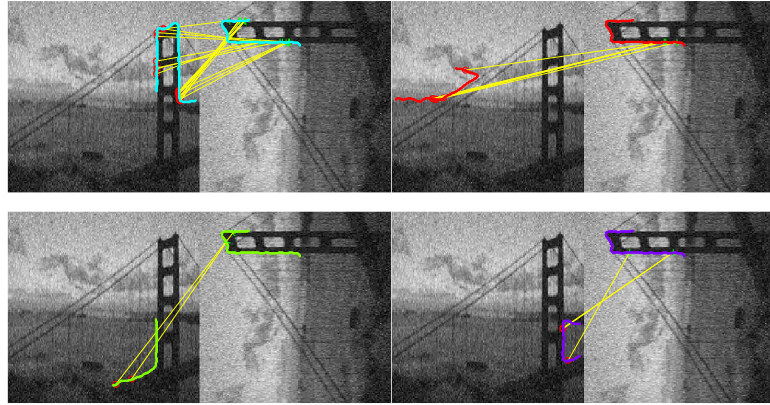
Şekil 4.8.4 Birinci resim üzerinde CVAED kenar çıkarıcı sonuçları



Şekil 4.8.5 Birinci resim üzerinde Canny kenar çıkarıcı sonuçları



Şekil 4.8.6 İkinci resim üzerinde MSME-CVAED kenar çıkarıcı sonuçları



Şekil 4.8.7 İkinci resim üzerinde LPGPCA-ED kenar çıkarıcı sonuçları



Şekil 4.8.8 İkinci resim üzerinde CVAED kenar çıkarıcı sonuçları



Şekil 4.8.9 İkinci resim üzerinde Canny kenar çıkarıcı sonuçları

#### 4.9 Çalışma Sürelerini Karşılaştırma

Bu çalışmada verilen yöntemlerin çalışma sürelerini birbiriyle karşılaştırmak için anahtar nokta başına düşen ortalama zaman ölçülmektedir. Elde edilen değerler Çizelge 4.9.1’de verilmektedir. Çizelgedeki sonuçlar için *bark* sınıfındaki birinci resim kullanılmıştır. Bu çalışmada elde edilen tüm sonuçlar 3,4 GHz ve 4 GB RAM özelliklerine sahip ve Windows 8 işletim sistemi bulunan bir bilgisayar kullanılmıştır. Önerilen yöntem için Matlab 2014a kullanılırken, diğer yöntemler için OPENCV 2.4.8 kullanılmıştır. Sadece anahtar nokta çıkarma için geçen süre ölçülmüştür. Verilen çizelgede *Time/Point* nokta başına düşen çalışma zamanını milli saniye (milliseconds) cinsinden ölçümünü, *Keypoints* elde edilen anahtar nokta sayısını ve *Time* ölçülen zamanı vermektedir. MSER ile SIFT betimleyicileri ve BRISK ile FREAK betimleyicileri diğerlerinden daha fazla zaman (bizimkisini hariç tutarsak) aldığı gözlemlenmiştir. Bunun yanında FAST ile BRIEF en hızlı çalışan denilebilir. Bizim çalışma (PROPOSED) ise aralarında en yavaş çalışan yöntemdir. Fakat eşleştirme için kullanılan programlama dilinin etkisi yazılan algorithmada göz önünde tutulursa önerdiğimiz yöntemin performansının iyi olduğu görülmektedir. Bunun yanında iki imgeyi eşleştirmek için en uzun eğrileri kullanıldığında önerilen hızının artması beklenmektedir. Yine değişen parametrelere göre sonuçlar değişebilmektedir.

Çizelge 4.9.1 Zamansal sonuçlar

<b>Method</b>	<b>Keypoints</b>	<b>Time (ms)</b>	<b>Time/Point</b>
<b>SIFT+SIFT</b>	3665	5.989	0.00163407
<b>SURF+SURF</b>	3634	1.083	0.00029768
<b>MSER+SIFT</b>	323	0.889	0.00275332
<b>BRISK+FREAK</b>	466	2.531	0.00543218
<b>BRISK+BRISK</b>	466	0.235	0.00050456
<b>ORB+ORB</b>	500	0.236	0.00047150
<b>FAST+BRIEF</b>	11880	0.083	0.00000698
<b>PROPOSED</b>	180	12.934	0.71855560

## 5 SONUÇLAR ve ÖNERİLER

Bu çalışmada eğri tabanlı imge eşleştirme ve çakıştırma üzerinde denemeler yapılmıştır. Eğrileri elde etmek için 4 farklı kenar tanıma yöntemi kullanılmıştır. Bunlar CVAED, MSMER-CVAED, LPGPCA-ED ve Canny yöntemleridir. Daha sonra elde edilen eğrilere Hausdorff yöntemi kullanılarak eşleştirme yapılmıştır. İkinci bir eşleşme kriteri olarak gradyen imgeler üzerindeki eğri noktalarından Harris öznitelikleri çıkartılarak kümedeki hiç benzemeyen eğriler ihmal edilmiştir. En küçük Hausdorff uzaklığı temel alınarak en iyi eşleşen eğriler bulunulmuştur. Son olarak en iyi eşleşen özniteliklerden ilgin dönüşüm hesaplanmış ve görüntüler döndürülerek çakıştırılmıştır. Ayrıca diğer öznitelik çıkarma yöntemlerinin performansı ile önerilen yöntemin performansı karşılaştırılmıştır. Performans karşılaştırma kriteri olarak kesinlik ve duyarlılık kriterleri kullanılmıştır.

Bu çalışmadaki deneysel çalışmalardan elde edilen kalite ölçümlerinden görülüyor ki, bütün şartlar altında önerilen çalışma tatmin edici sonuçlar verebilmiştir. Özellikle gürültü durumlarında önerilen yöntemin performansı başı çekmektedir. Bundan dolayı görüntü eşleştirme veya çakıştırma işlemleri için kullanılabilmesine inanılmaktadır.

Deneysel sonuçlar üzerinde yapılan gözlemlerde, resimdeki bulanıklığın değişmesi durumunda ORB, MSER ile SIFT betimleyicileri ya da FAST ile BRIEF betimleyicileri en iyi performansları verdikleri görülmektedir. Yine görüş açısı değişimi için MSER ile SIFT betimleyicileri ya da SIFT arasından bir yöntem tercih edilebilir. Resimdeki parlaklık ya da ışık şiddetinin değişimi gibi durumlarda, FAST ile BRIEF betimleyicileri kullanılabilir. Eğer sıkıştırılmaya karşı güçlü ve duyarlı olmayan bir öznitelik çıkarma yöntemi kullanmak istiyorsak, ORB yöntemi seçilebilir. İlaveten yapılan çalışma için en önemli şey hız ise FAST ile BRIEF betimleyicileri bizim için vazgeçilmez olacaktır. Önerilen yöntemin ise bazı durumlarda orta seviye bazı durumlarda ise imgedeki bozulmalar artmasına rağmen en yüksek sonuçları sergilediği görülmektedir.

Önemli bir nokta olarak vurgulamak gerekirse, her bir metodun başarılı olacağı görevler farklıdır. Ama bunlar arasında en az zamanda çalışan ve en iyi sonuç verecek birini seçmekte fayda vardır. Bu çalışmanın bir amacı olarak hangi yöntemin görüntü eşleştirme ve çakıştırma daha iyi performans verdiğini

belirlemektir. Bu çalışmanın sonuçlarına bakılarak hangi yöntemin bir para eşleştirme veya yüz eşleştirme gibi benzer eşleştirme çalışmaları ve görüntü çakıştırma çalışmaları için faydalı olduğuna karar verebilir. Bu çalışmanın bir devamı 3B (3 boyutlu) görüntü eşleştirme ve çakıştırma yapılması hedeflenmektedir.

## KAYNAKÇA

- [1] H. Sarnel, "İzdüşümlere Kepstrum Tekniği Uygulanması ile Görüntü Çakıştırma," 2000.
- [2] B. Glocker, A. Sotiras, N. Komodakis, ve N. Paragios, "Deformable medical image registration: Setting the state of the art with discrete methods," *Annual review of biomedical engineering*, Sayı 13, 219-244, 2011.
- [3] S. Klein, M. Staring, K. Murphy, M. A. Viergever, ve J. P. Pluim, "Elastix: a toolbox for intensity-based medical image registration," *Medical Imaging, IEEE Transactions on*, Sayı 29, 196-205, 2010.
- [4] R. Shams, P. Sadeghi, R. A. Kennedy, ve R. I. Hartley, "A survey of medical image registration on multicore and the GPU," *Signal Processing Magazine, IEEE*, Sayı 27, 50-60, 2010.
- [5] E. Jayabalan ve A. Krishnan, "Target tracking in aerial videos using image registration techniques," in *Computer Networks and Information Technologies*, ed: Springer, 475-477, 2011.
- [6] F. Chatelain, J.-Y. Tournet, J. Inglada, ve A. Ferrari, "Bivariate gamma distributions for image registration and change detection," *Image Processing, IEEE Transactions on*, Sayı 16, 1796-1806, 2007.
- [7] X. Dai ve S. Khorram, "The effects of image misregistration on the accuracy of remotely sensed change detection," *Geoscience and Remote Sensing, IEEE Transactions on*, Sayı 36, 1566-1577, 1998.
- [8] H.-Y. Shum ve R. Szeliski, "Construction of panoramic image mosaics with global and local alignment," in *Panoramic vision*, ed: Springer, 227-268, 2001.
- [9] W. W. Yue ve W. Yun-dong, "Free image registration and mosaicing based on tin and improved szeliski algorithm," in *Proceedings of ISPRS Congress*, 2008.
- [10] X. Fang, B. Luo, J. Tang, H. Zhao, B. He, ve H. Wu, "Registration of blurred images for image mosaic," in *Computer-Aided Design and Computer Graphics, CAD/Graphics' 09. 11th IEEE International Conference on*, 184-189, 2009.
- [11] M. Hild ve G. Umeda, "Image registration in stereo-based multi-modal imaging systems," in *Image and Signal Processing and Analysis, ISPA 2005. Proceedings of the 4th International Symposium on*, 70-75, 2005.
- [12] N. Ritter, R. Owens, J. Cooper, R. H. Eikelboom, ve P. P. Van Saarloos, "Registration of stereo and temporal images of the retina," *Medical Imaging, IEEE Transactions on*, Sayı 18, 404-418, 1999.
- [13] P. Chalermwat, "High performance automatic image registration for remote sensing," George Mason University, 1999.
- [14] H.-M. Chen, P. K. Varshney, ve M. K. Arora, "Performance of mutual information similarity measure for registration of multitemporal remote sensing images," *Geoscience and Remote Sensing, IEEE Transactions on*, Sayı 41, 2445-2454, 2003.
- [15] Q. Li, G. Wang, J. Liu, ve S. Chen, "Robust scale-invariant feature matching for remote sensing image registration," *Geoscience and Remote Sensing Letters, IEEE*, Sayı 6, 287-291, 2009.

- [16] E. Seke. *Image Restoration Course Notes* Available: <http://eseke.ogu.edu.tr/imr/imr.html>
- [17] B. S. Manjunath. *Vision Research Lab, Image Registration*. Available: <http://vision.ece.ucsb.edu/registration/demo/examples.shtml>
- [18] L. G. Brown, "A survey of image registration techniques," *ACM computing surveys (CSUR)*, Sayı 24, 325-376, 1992.
- [19] J. Maintz ve M. A. Viergever, "A survey of medical image registration," *Medical image analysis*, Sayı 2, 1-36, 1998.
- [20] P. A. Van den Elsen, E.-J. Pol, ve M. A. Viergever, "Medical image matching-a review with classification," *Engineering in Medicine and Biology Magazine, IEEE*, Sayı 12, 26-39, 1993.
- [21] B. Zitova ve J. Flusser, "Image registration methods: a survey," *Image and vision computing*, Sayı 21, 977-1000, 2003.
- [22] J. M. Fitzpatrick, D. L. Hill, ve C. R. Maurer Jr, "Image registration."
- [23] S. Boda, "Feature-based Image Registration," 2009.
- [24] C. Harris ve M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, 50, 1988.
- [25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, Sayı 60, 91-110, 2004.
- [26] G. Bebis. *Computer Vision Lecture Notes, Interest Point Descriptors Matching*. Available: <http://www.cse.unr.edu/~bebis/CS485/Lectures/InterestPointDescriptorsMatching.ppt>
- [27] M. D. Mura. *Image Analysis Lecture Notes*. Available: [http://www.gipsa-lab.grenoble\\_inp.fr/~mauro.dallamura/teaching/InternshipESRF\\_PaleoPierre\\_SICOM\\_STIC\\_Promotion2014.pdf](http://www.gipsa-lab.grenoble_inp.fr/~mauro.dallamura/teaching/InternshipESRF_PaleoPierre_SICOM_STIC_Promotion2014.pdf)
- [28] Y. Ke ve R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, Sayı 2, 506-513, 2004.
- [29] H. Bay, T. Tuytelaars, ve L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, ed: Springer, 404-417, 2006.
- [30] P. Panchal, S. Panchal, ve S. Shah, "A Comparison of SIFT and SURF," *International Journal of Innovative Research in Computer and Communication Engineering*, Sayı 1, 97-98, 2013.
- [31] P. Viola ve M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Sayı 1, 511-518, 2001
- [32] E. Rosten, R. Porter, ve T. Drummond, "Faster and better: A machine learning approach to corner detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Sayı 32, 105-119, 2010.
- [33] E. Rosten ve T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*, ed: Springer, 2006, pp. 430-443.
- [34] E. Rosten. *Akademik Notes*. Available: <http://www.edwardrosten.com/work/fast.html>

- [35] S. Leutenegger, M. Chli, ve R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2548-2555, 2011.
- [36] J. Matas, O. Chum, M. Urban, ve T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, Sayı 22, 761-767, 2004.
- [37] D. Obdržálek, S. Basovnik, L. Mach, ve A. Mikulik, "Detecting scene elements using maximally stable colour regions," in *Research and Education in Robotics-EUROBOT 2009*, ed: Springer, 107-115, 2010.
- [38] D. Nistér ve H. Stewenius, "Linear time maximally stable extremal regions," in *Computer Vision-ECCV 2008*, ed: Springer, 183-196, 2008.
- [39] V. Lepetit ve P. Fua, "Keypoint recognition using randomized trees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Sayı 28, 1465-1479, 2006.
- [40] M. Ozuysal, M. Calonder, V. Lepetit, ve P. Fua, "Fast keypoint recognition using random ferns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Sayı 32, 448-461, 2010.
- [41] M. Calonder, V. Lepetit, C. Strecha, ve P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision-ECCV 2010*, ed: Springer, 778-792, 2010.
- [42] E. Rublee, V. Rabaud, K. Konolige, ve G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2564-2571, 2011.
- [43] A. Alahi, R. Ortiz, ve P. Vanderghenst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 510-517, 2012.
- [44] G. Papari, N. Petkov, ve P. Campisi, "Artistic Edge and Corner Preserving Smoothing," *Image Processing, IEEE Transactions on*, Sayı 16, 2449-2462, 2007.
- [45] G. Papari, N. Petkov, ve P. Campisi "Artistic Edge and Corner Preserving Smoothing" Available: [cs.rug.nl/~imaging/artisticsmoothing/introduction.html](http://cs.rug.nl/~imaging/artisticsmoothing/introduction.html)
- [46] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita, "Processing of RI-angiocardigraphic images," in *Digital processing of biomedical images*, ed: Springer, 187-202, 1976.
- [47] R. Van den Boomgaard, "Decomposition of the Kuwahara-Nagao Operator In Terms Of Linear Smoothing And Morphological Sharpening," in *Mathematical Morphology: Proc. Sixth International Symposium*, 283-292, 2002.
- [48] H. J. Heijmans, "Connected morphological operators for binary images," *Computer Vision and Image Understanding*, Sayı 73, 99-120, 1999.
- [49] A. Meijster ve M. H. Wilkinson, "A comparison of algorithms for connected set openings and closings," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Sayı 24, 484-494, 2002.
- [50] C. Tomasi ve R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, 1998, pp. 839-846.
- [51] W. K. Pratt, "Digital image processing," *New York*, Sayı 242, 1978.

- [52] M. Basu, "Gaussian-based edge detection methods a survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, Sayı 32, 252-260, 2002.
- [53] C. Akinlar ve C. Topal, "EDPF: A Real-time Parameter-free Edge Segment Detector with a False Detection Control," *International Journal of Pattern Recognition and Artificial Intelligence*, Sayı 26, 2012.
- [54] F. Catté, P.-L. Lions, J.-M. Morel, ve T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM Journal on Numerical analysis*, Sayı 29, 182-193, 1992.
- [55] B. Sumengen ve B. Manjunath, "Multi-scale edge detection and image segmentation," in *European signal processing conference (EUSIPCO)*, 2005.
- [56] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 679-698, 1986.
- [57] D. Bisaccia. (2014). *Ders Notları, Edge Detection*. Available: <http://www.cs.ucf.edu/courses/cot4810/spr2008/EdgeDetection.ppt>
- [58] M. Koc ve A. Barkana, "Feature selection method with common vector and discriminative common vector approaches," in *Signal Processing and Communications Applications 2011 IEEE 19th Conference on*, 98-101, 2011.
- [59] K. Delac, M. Grgic, ve S. Grgic, "Independent comparative study of PCA, ICA, and LDA on the FERET data set," *International Journal of Imaging Systems and Technology*, Sayı 15, 252-260, 2005.
- [60] M. B. Gulmezoglu, V. Dzhafarov, M. Keskin, ve A. Barkana, "A novel approach to isolated word recognition," *Speech and Audio Processing, IEEE Transactions on*, Sayı 7, 620-628, 1999.
- [61] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, Sayı 11, 23-27, 1975.
- [62] C. Topal ve C. Akinlar, "Edge Drawing: A combined real-time edge and segment detector," *Journal of Visual Communication and Image Representation*, Sayı 23, 862-872, 2012.
- [63] K. Bowyer, C. Kranenburg, ve S. Dougherty, "Edge detector evaluation using empirical ROC curves," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*, 1999.
- [64] ROC Veritabanı. Available: <http://figment.csee.usf.edu/edge/roc/>
- [65] K. Ray, "Unsupervised edge detection and noise detection from a single image," *Pattern Recognition*, Sayı 46, 2067-2077, 2013.
- [66] K. Özkan. Ş. Işık, "A Novel Multi-Scale And Multi-Expert Edge Detection Method Based On Common Vector Approach," presented at the Signal Processing and Communications Applications Conference (SIU), 2014 22nd, Trabzon, 2014.
- [67] R. Edizkan, B. Tiryaki, T. Büyükcan, ve İ. Uzun, "Ses Komut Tanıma ile Gezgin Araç Kontrolü."
- [68] I. Sobel ve G. Feldman, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in*, 271-272, 1968.
- [69] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics*, Sayı 10, 15-19, 1970.

- [70] G. Papari ve N. Petkov, "Adaptive pseudo dilation for gestalt edge grouping and contour detection," *Image Processing, IEEE Transactions on*, Sayı 17, 1950-1962, 2008.
- [71] P. Meer ve B. Georgescu, "Edge detection with embedded confidence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Sayı 23, 351-1365, 2001.
- [72] G. Papari ve N. Petkov. *RUG Database*. Available: <http://www.cs.rug.nl/~imaging/APD/rug/rug.html>
- [73] L. Zhang, W. Dong, D. Zhang, ve G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, Sayı 43, 1531-1549, 2010.
- [74] J. Jakubowicz, R. Grompone von Gioi, J. Morel, and G. Randall. *Image Processing Online*. Available: [http://demo.ipol.im/demo/gjmr\\_line\\_segment\\_detector/archive](http://demo.ipol.im/demo/gjmr_line_segment_detector/archive)
- [75] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a line segment detector," *Image Processing On Line*, 2012.
- [76] C. Gope ve N. Kehtarnavaz, "Affine invariant comparison of point-sets using convex hulls and hausdorff distances," *Pattern Recognition*, Sayı 40, 309-320, 2007.
- [77] D. P. Huttenlocher, G. A. Klanderman, ve W. J. Rucklidge, "Comparing images using the Hausdorff distance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Sayı 15, 850-863, 1993.
- [78] O. Jesorsky, K. J. Kirchberg, ve R. W. Frischholz, "Robust face detection using the hausdorff distance," in *Audio-and video-based biometric person authentication*, 90-95, 2001.
- [79] W. Rucklidge, *Efficient visual recognition using the Hausdorff distance*: Springer-Verlag New York, Inc., 1996.
- [80] G. T. Toussaint, "Hausdorff Distance," ed.
- [81] G. Rote, "Computing the minimum Hausdorff distance between two point sets on a line under translation," *Information Processing Letters*, Sayı 38, 123-127, 1991.
- [82] M.-P. Dubuisson ve A. K. Jain, "A modified Hausdorff distance for object matching," in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, 566-568, 1994.
- [83] E. Belogay, C. Cabrelli, U. Molter, ve R. Shonkwiler, "Calculating the Hausdorff distance between curves," *Information Processing Letters*, Sayı 64, 17-22, 1997.
- [84] M. A. Fischler ve R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Sayı 24, 381-395, 1981.
- [85] E. Bendeş, V. Aslantaş, R. Kurban ve A. N. Toprak, "Çoklu Algılayıcılardan Alınan Görüntülerde Eşleştirme Yöntemlerinin Karşılaştırılması," presented at the Elektrik-Elektronik ve Bilgisayar Sempozyumu.
- [86] D. C. Paquin, D. Levy, E. Schreibmann, ve L. Xing, "Multiscale image registration," *Mathematics*, p. 5, 2006.

- [87] J. P. Pluim, J. A. Maintz, ve M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *Medical Imaging, IEEE Transactions on*, Sayı 22, 986-1004, 2003.
- [88] K. Mikolajczyk ve C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Sayı 27, 1615-1630, 2005.
- [89] V. G. Group. *Department of Engineering Science, University of Oxford*. Available: <http://www.robots.ox.ac.uk/~vgg/data/data-aff.html>