

**ÖLÇEK DEĞİŞİMSİZ ÖZELLİK DÖNÜŞÜMÜ KULLANARAK
NESNE VE GÖRÜNTÜ TANIMA SÜRECİNDE EŞLEŞTİRME
YÖNTEMLERİNİ KARŞILAŞTIRMA VE HIZLANDIRMA**

**COMPARING AND SPEEDING-UP MATCHING METHODS IN THE
PROCESS OF OBJECT AND IMAGE RECOGNITION BY USING
SCALE INVARIANT FEATURE TRANSFORM**

MEHMET HAKAN AKŞİT

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
ELEKTRİK ve ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LISANS TEZİ
olarak hazırlanmıştır.

2012

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından ELEKTRİK ve ELEKTRONİK Mühendisliği ANABİLİM DALI'nda YÜKSEK LISANS TEZİ olarak kabul edilmiştir.

Başkan :
Prof.Dr. H. Selçuk GEÇİM

Üye (Danışman) :
Doç.Dr. Ali Ziya ALKAR

Üye :
Yrd.Doç.Dr. A. Semih BİNGÖL

Üye :
Yrd.Doç.Dr. Kayhan İMRE

Üye :
Yrd.Doç.Dr. Umut SEZEN

ONAY

Bu tez Hacettepe Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri uyarınca yukarıdaki jüri üyeleri tarafından/....../2012 tarihinde uygun görülmüş ve Enstitü Yönetim Kurulunca/....../2012 tarihinde kabul edilmiştir.

Prof.Dr. Fatma SEVİN DÜZ

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

ÖLÇEK DEĞİŞİMSİZ ÖZELLİK DÖNÜŞÜMÜ KULLANARAK NESNE VE GÖRÜNTÜ TANIMA SÜRECİNDE EŞLEŞTİRME YÖNTEMLERİNİ KARŞILAŞTIRMA VE HIZLANDIRMA

MEHMET HAKAN AKŞİT

ÖZ

Nesne tanıma; uzay araştırmalarından biyomedikal araştırmalarına, askeri kullanımdan günlük basit kullanıma kadar; görüntüleme sistemlerinin kullanıldığı hemen her alanda önemli bir yer tutmaktadır. Bu nedenle nesne ve örüntü tanıma amacıyla geçmişten bu yana bir çok araştırma yapılmış, yöntem geliştirilmiş ve halen geliştirilmektedir.

Nesne tanıma yöntemleri, kullanım ve araştırma amacına göre sınıflara ayrılmaktadır. Bir sınıflandırma da yöntemlerin karmaşıklık seviyesiyle yapılabilmektedir. Basit yöntemlerin genel olarak kullanımı görüntülemenin temiz olduğu, görüntü alınacak ortamın çok iyi hazırlandığı, hızın önemli olduğu sistemlerde uygun olmaktadır. Karmaşık yöntemler, görüntülemenin mükemmel olmadığı, nesnenin seçilmesinin zor olabildiği, görüntünün ortam koşullarının çok iyi hazırlanamadığı veya hızın öneminin azaldığı sistemlerde tercih edilmektedir.

Genellikle karmaşık yöntemler kararlı ve yüksek başarılı sonuçlar verdiği için bir çok sistemde kullanılmak istenmektedir. Araştırmalar göstermektedir ki; yapılan işlemlerin yoğunluğundan, bu yöntemlerin kullanımı kimi zaman sonuca hızlı ulaşılması gerektiği durumlarda uygun olamamaktadır.

Bu tez çalışmasında, karmaşık bir yöntem olan Ölçek Değişimsiz Özellik Dönüşümü (SIFT) yöntemi kullanılarak nesne tanıma için görüntüden tanımlayıcı vektörler çıkarılmıştır. Bu yöntemle nesne eşleştirmek için, tanımlayıcı vektörlerin birbirilerine olan Öklit uzaklığı bulunması gerekmektedir. Bunun için enyakın komşuları arama yöntemlerinden yaygın kullanılanlarından ikisi karşılaştırılmıştır. Basit Enyakın Komşu Araması ile Eniyi İlk Küme Aramasından hangisinin hızlı arama yapabilecek koşulda olduğu deneysel sonuçlarıyla gösterilmiştir. Daha sonra paralel işlemci kullanılarak hem yöntemler hızlandırılmış, hem de Basit Enyakın Komşu aramasının kullanılabilceği koşullar genişletilmiştir. Özellik tanımlayıcı oluşturan ve Öklit uzaklığına göre Enyakın Komşu aramasıyla eşleme yapılan hemen her yöntem için hız başarımı sağlayacak sonuçlar elde edilmiştir.

Anahtar Kelimeler: Nesne Tanıma, Örüntü Tanıma, Ölçek Değişimsiz Özellik Tespiti, Öklit Uzaklığı, Enyakın Komşu Araması, Eniyi İlk Küme.

Danışman: Doç.Dr. Ali Ziya ALKAR, Hacettepe Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü

COMPARING AND SPEEDING-UP MATCHING METHODS IN THE PROCESS OF OBJECT AND IMAGE RECOGNITION BY USING SCALE INVARIANT FEATURE TRANSFORM

MEHMET HAKAN AKŞİT

ABSTRACT

Object recognition has an important coverage on almost all systems where imaging takes place; from space researches to biomedical researches, from military use to daily civil use. For this purpose, there have been many researches and developments done in the area of object and pattern recognition and are still being done.

Object recognition methods are categorized according to the use and research purposes. One categorization type can be made according to the complexity of the methods. Simple methods are generally compatible with the systems in which imaging is clean, in which the imaging environment is well prepared and where speed has importance. Complex methods are preferred in the systems in which imaging is not perfect, in which objects in the image are not much clear to be spotted, in which the imaging environment is not well prepared or speed is not so much important.

Because of the fact that complex methods yield stable and high performance results, they are referred in many systems. As the researches show that, process intensity of these methods are so high, it may not be always feasible to use them.

In this thesis study, a complex method Scale Invariant Feature Transform (SIFT) is used for object recognition and descriptor vectors are extracted from the object image. In order to match the images, the Euclidean distance of the descriptor vectors must be found. For this purpose, two methods to find nearest neighbor are compared. Simple Nearest Neighbor Search and Best Bin First methods are simulated to show which one is faster according to different conditions. After that, with the use of parallel processing, both methods are speeded up and also the conditions that the Simple Nearest Neighbor Search can be used are extended. These results will contribute to the speed performance for almost all methods that generate feature descriptor vectors and use Euclidean distance to match these descriptors.

Keywords: Object Recognition, Pattern Recognition, Scale Invariant Feature Transform, Euclidean Distance, Nearest Neighbor Search, Best Bin First.

Advisor: Assoc.Prof. Ali Ziya ALKAR, Hacettepe University, Department of Electrical and Electronics Engineering

TEŐEKKÜR

Öncelikle, tez alıřmalarım boyunca bilgi ve deneyimleriyle bana yol gösteren tez danıřmanım Sayın Do. Dr. Ali Ziya ALKAR'a teőekkür ederim.

alıřmalarımda motivasyonumu kaybettiđim durumlarda her zaman yanımda olan, mesaide, gnlk hayatta desteđini esirgemeyen deđerli dostum Hıdır AŐKAR'a, alıřmalarımda yardımını esirgemeyen deđerli dostum Ömer HALİLOĐLU'na, bana gösterdikleri yakınlık ve yardımlarından dolayı tm Hacettepeli alıřma arkadaşlarıma teőekkür ederim.

Son olarak, sevgi ve destekleriyle bana daima g veren, maddi ve manevi hi bir desteđini esirgemeyen anneme, babama, ablalarım Ayten ve Farahnaz'a, kardeřlerim Emre ve Onur'a, motivasyonumu toparlamamda yardımcı olan kk yeđerlerim Yađız ve Dađhan ve de Efe'ye verdikleri tm desteklerden dolayı teőekkr ederim.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ	iv
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	x
SİMGELER VE KISALTMALAR DİZİNİ	xi
SÖZLÜK DİZİNİ	xii
1. GİRİŞ	1
2. RESİMLERDEN BİLGİ ÇIKARIMI VE YÖNTEMLERİ	5
2.1. Görünüm Tabanlı Özellik Tespitleme Yöntemleri	5
2.1.1. Kenar Tespiti	6
2.1.1.1. Roberts Çapraz Kenar Tespitleyicisi	6
2.1.1.2. Prewitt Kenar Tespitleyicisi	7
2.1.1.3. Sobel Kenar Tespitleyicisi	8
2.1.1.4. Canny Kenar Tespitleyicisi	9
2.1.2. Köşe Tespitleyicisi	10
2.1.2.1. Moravec Köşe Tespitleme Algoritması	11
2.1.2.2. Harris Stephens, Shi-Tomasi Köşe Tespitleyici Algoritmaları	12
2.1.2.3. Foerstner Köşe Tespitleyicisi	14
2.1.2.4. Wang ve Brady Köşe Tespitleyici Algoritması	15
2.1.2.5. SUSAN Köşe Tespitleyicisi	15
2.1.2.6. Trajkovic ve Hedley Köşe Tespitleyici	17
2.1.3. Damlacık Tespiti	17
2.1.3.1. Gauss'un Laplace İşleci (LoG)	17
2.1.3.2. Gaussların Farkı (DoG) Yaklaşımı	19

2.1.3.3. Hessian Determinantı	20
2.1.3.4. Melez Laplace İşleci ve Hessian Determinantı İşleci . .	20
2.1.4. Çıkıntı Tespitleyici	21
2.1.4.1. Sabit Ölçekte Çıkıntılarının ve Girintilerin Diferansiyel Geometrik Tanımı	22
2.1.4.2. Değişken Ölçekli Çıkıntılarının Hesaplanması	23
2.1.5. Hough Dönüşümü ile Şekil Tespiti	25
2.1.5.1. Teori ve Uygulama	25
2.1.5.2. Çizgi Tespiti	25
2.1.5.3. Daire Tespiti	27
2.1.5.4. Genel Hough Dönüşümü (GHT)	28
2.2. Özellik Tabanlı Tanımlayıcı Oluşturma Yöntemleri	28
2.2.1. Ölçek Değişimsiz Özellik Dönüşümü (SIFT)	29
2.2.1.1. Anahtar Noktaların Bulunması	29
2.2.1.2. Anahtar Noktalara Yönelim Atama	29
2.2.1.3. Tanımlayıcı Oluşturma	30
2.2.2. Hızlandırılmış Gürbüz Özellikler (SURF)	30
2.2.2.1. İlgi Noktası Tanımlayıcıları	32
2.2.3. Gradyan Konumu ve Yönelimi Histogramı (GLOH)	32
2.2.4. Yönlü Gradyanların Histogramı (HOG)	33
3. NESNE TANIMA İÇİN ÖZELLİK ÇIKARIMI VE TANIMLAYICI OLUŞTURMA	36
3.1. Ölçek Değişimsiz Özellik Dönüşümü (SIFT) Yöntemi ve Uygulamaları .	36
3.1.1. Kullanılan Donanım ve Yazılım Araçları	36
3.1.2. Ön İşlemler	36
3.2. Anahtar Noktaları Bulma	37
3.2.1. Gaussların Farkları Piramidinin Oluşturulması	37
3.2.2. Gauss Farkı Piramidlerinde Bölgesel Enuç Tespiti	41
3.2.3. Hassas Anahtar Noktaları Tespiti	43
3.2.4. Kenar Tepkilerinin ve Düşük Kontrastların Elenmesi	46

3.3. Yönelim Atama	49
3.4. Bölgesel Görüntü Tanımlayıcı	51
3.4.1. Tanımlayıcı Vektörü Oluşturma	51
4. ÖZELLİK TANIMLAYICILARI EŞLEME VE HIZLANDIRMA	53
4.1. Enyakın Komşu Araması (NNS)	64
4.2. KD Ağacı Algoritmasıyla Arama	64
4.3. Eniyi İlk Küme (BBF) Algoritmasıyla Arama	67
4.4. Yöntemlerin Karşılaştırılması ve Hızlandırma	67
5. SONUÇ	74
KAYNAKLAR DİZİNİ	76
ÖZGEÇMİŞ	80

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 2.1. İşlemlerin Uygulandığı Orijinal Görüntü	5
Şekil 2.2. Roberts Çapraz Yöntemiyle Tespit Edilmiş Kenar Görüntüsü	7
Şekil 2.3. Prewitt Yöntemiyle Tespit Edilmiş Kenar Görüntüsü	8
Şekil 2.4. Sobel Yöntemiyle Tespit Edilmiş Kenar Görüntüsü	9
Şekil 2.5. Canny Yöntemiyle Tespit Edilmiş Kenar Görüntüsü	10
Şekil 2.6. Moravec Yöntemiyle Tespit Edilmiş Köşe Noktaları	11
Şekil 2.7. Harris Yöntemiyle Tespit Edilmiş Köşe Noktaları	13
Şekil 2.8. Shi-Tomasi Yöntemiyle Tespit Edilmiş Köşe Noktaları	14
Şekil 2.9. SUSAN Yöntemiyle Tespit Edilmiş Köşe Noktaları	16
Şekil 2.10. Gaussların Farkıyla Damlacık Tespit Edilmiş Görüntü	19
Şekil 2.11. Hessian Determinantıyla Damlacık Tespit Edilmiş Görüntü	21
Şekil 2.12. Çıkıntı Tespit Edilmiş Görüntü	24
Şekil 2.13. Hough Dönüşümüyle Çizgi Tespiti	26
Şekil 2.14. 1. Gauss süzgecinin r yönünde ikinci kısmi türevi, 2. Gauss süzgecinin cr yönünde ikinci kısmi türevi, 3. Gauss süzgecinin r yönünde ikinci kısmi türevine SURF yaklaşımı, 4. Gauss süzgecinin cr yönünde ikinci kısmi türevine SURF yaklaşımı. Gri bölgeler sıfır olarak belirlenmiştir.	31
Şekil 2.15. SIFT Tanımlayıcı 1. Özellik tespiti yapılmış bölge, 2. Gradyan görüntüsünde ızgaralar, 3. Histogram boyutları, 4. 8 yönelim düzleminden 4'ü, 5.Kartezyan ve log-polar konumlama ızgaraları. Görüldüğü gibi log-polar ızgaraların gösterdiği gibi bölge 4 açısız olmak üzere toplamda 9 bölgeye bölünmüştür.	33
Şekil 3.1. Gauss ve Gaussların Farkı Piramitleri	39
Şekil 3.2. Gauss Piramiti Görüntüleri	40
Şekil 3.3. Enuç Değer Tespiti	41
Şekil 3.4. Görüntüdeki Aday Anahtar Noktaları	42
Şekil 3.5. Taylor Açılımı Uygulandığında Bulunan Enbüyük Değer	43
Şekil 3.6. Görüntüdeki Hassas Tespit Sonrası Kesin Anahtar Noktaları	45

Şekil 3.7. Görüntüdeki Kenar Tepkileri Elendikten Sonraki Kesin Anahtar Noktaları	48
Şekil 3.8. Görüntüdeki Anahtar Noktaların Gradyan Yönelimleri	50
Şekil 3.9. Tanımlayıcı Vektörün Oluşturulması	51
Şekil 4.1. Döndürülmüş, Ölçeklendirilmiş, Işıklandırması ve 3B Görüş Açısı Değiştirilmiş Nesnenin Eşleştirilmesi	54
Şekil 4.2. Kısmi Engelleme Olduğunda Nesne Tanıma	55
Şekil 4.3. Yönelimler Değiştiğinde Nesne Tanıma	55
Şekil 4.4. Bakış Açısı ve Işıklandırma Değiştiğinde ve Kısmi Engelleme Olduğunda Nesne Tanıma	56
Şekil 4.5. Nesne Sayısı Arttırıldığında ve Yönelim ve Işıklandırma Değiştirildiğinde Nesne Tanıma	57
Şekil 4.6. Nesne Sayısı Arttırıldığında ve Yönelim Değiştiğinde Nesne Tanıma	57
Şekil 4.7. Nesne Sayısı Arttırıldığında ve Bakış Açısı Değiştiğinde Nesne Tanıma	58
Şekil 4.8. Nesne Sayısı Arttırıldığında, Bakış Açısı ve Işıklandırma Değiştiğinde Nesne Tanıma	58
Şekil 4.9. Nesne Sayısı Arttırıldığında, Bakış Açısı ve Işıklandırma Değiştiğinde Nesne Tanıma	59
Şekil 4.10. Nesne Sayısı Arttırıldığında, Bakış Açısı ve Işıklandırma Değiştiğinde Nesne Tanıma	59
Şekil 4.11. Nesne Sayısı Arttırıldığında ve Işıklandırma Arttırıldığında Nesne Tanıma	60
Şekil 4.12. Nesne Sayısı Arttırıldığında ve Koşullar Değiştiğinde Nesne Tanıma	61
Şekil 4.13. Nesne Sayısı Arttırıldığında ve Işıklandırma ve Görüş Açısı Değiştiğinde Nesne Tanıma	61
Şekil 4.14. Nesne Sayısı Arttırıldığında ve Işıklandırma Değiştiğinde ve Kısmi Engelleme Olduğunda Nesne Tanıma	62
Şekil 4.15. Nesne Sayısı Arttırıldığında ve Kısmi Engelleme Olduğunda Nesne Tanıma	62
Şekil 4.16. KD Ağacı Yapısı, $K = 3$	66

Şekil 4.17. 300 ve 400 Tanımlayıcı Vektörlü Arama ve Eşleme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Karşılaştırması	69
Şekil 4.18. Tek İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması	70
Şekil 4.19. Paralel Çift İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması	71
Şekil 4.20. Paralel Üç İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması	71
Şekil 4.21. Paralel Dört İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması	72
Şekil 4.22. Farklı Sayıda Paralel İşlemcilerin Kullanımının Basit Enyakın Komşu Aramasında, Çeşitli Veri Tabanı Büyüklüklerinde Zaman Başarımına Etkisi	72
Şekil 4.23. Farklı Sayıda Paralel İşlemcilerin Kullanımının Eniyi İlk Küme Aramasında, Çeşitli Veri Tabanı Büyüklüklerinde Zaman Başarımına Etkisi	73

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 4.1. Eşleşme Başarımı Tablosu	63
Çizelge 4.2. KD Ağacı Örnek Veri Kümesi	65

SİMGELER VE KISALTMALAR DİZİNİ

BBF	: Best Bin First (Eniyi İlk Küme)
DoG	: Difference of Gaussians (Gaussların Farkı)
GHT	: Generalized Hough Transform (Genel Hough Dönüşümü)
GLOH	: Gradient Location of Oriented Histograms (Gradyan Konumu ve Yönelimi Histogramı)
HOG	: Histogram of Oriented Gradients (Gradyan Yönelimleri Histogramı)
KDT	: K-Dimensional Tree (K-Boyutlu Ağaç)
LoG	: Laplacian of Gaussian (Gauss'un Laplace İşleci)
NNS	: Nearest Neighbor Search (Enyakın Komşu Araması)
RGB	: Red Green Blue (Kırmızı Yeşil Mavi)
SIFT	: Scale Invariant Feature Transform (Ölçek Değişimsiz Özellik Dönüşümü)
SURF	: Speeded Up Robust Features (Hızlandırılmış Gürbüz Özellikler)
SUSAN	: Smallest Univalued Segment Assimilating Nucleus (Enküçük Tek Değerli Parça Özümleyen Çekirdek)
*	: Convolution (Evrşim)
<i>c</i>	: Column Coordinate Variable (Sütun Koordinat Değişkeni)
<i>r</i>	: Row Coordinate Variable (Satır Koordinat Değişkeni)

SÖZLÜK DİZİNİ

Arapiksel	: Sub-pixel
Asli Eğrilik	: Principal Curvature
Ayrık	: Difference
Başarım	: Performance
Çıkıntı	: Ridge
Dalgacık	: Wavelet
Damlacık	: Blob
Değişimsiz	: Invariant
Değiştirge	: Parameter
Enbüyük	: Maximum
Eniyi	: Best
Enküçük	: Minimum
Enuç	: Extrema
Enyakın	: Nearest
Görüşaçısı	: Viewpoint
İşleç	: Operator
Kendiliğinden	: Automatic
Küme	: Bin
Matrisin İzi	: Trace of A Matrix
Ortadeğer	: Median
Ölçek-normal	: Scale-normal
Ölçek-uzay	: Scale-space
Özdeğer	: Eigenvalue
Özellik	: Feature
Seviye	: Octave, Level
Süzgeçleme	: Filtering
Tanımlayıcı	: Descriptor
Tümlev	: Integrated
Yönelim	: Orientation

1. GİRİŞ

Uzun süredir bir çok araştırma ve geliştirme yapılan nesne tanıma, görüntü işleme alanının en geniş konularından biri olmakla birlikte halihazırda bir çok sistem ve araçta kullanılmaktadır. Sanayide kullanılan robotlar, robot kollar ve insansı robotlarda yapılan hareketleri belirlemek için nesne tanıma önemli bir süreçtir [1]. Savunma sanayinde, hareket izleme, hedef belirleme ve takip, güdüm amacıyla nesne tespiti ve tanıma önemli bir yer tutmaktadır [2]. Sivil güvenlik ve gizlilik söz konusu olduğunda yüz tanıma, retina tarama, parmak izi, yüz damar yapısı gibi bir çok kişiye özgü özelliklerin tespiti ve doğrulanması için görüntüden özellik çıkarıp eşleme önem taşımaktadır [3][4]. Bunların yanısıra yine sivil hayatta, trafik güvenliği, araç izleme ve takip sistemlerinde, araç plakası ve modelinin tespiti açısından önemlidir [5] [6]. Yine güvenlik amacıyla açık ve kapalı devre alan izleme ve güvenliği için kullanılmaktadır [7][8]. Kısaca nesne tanıma, günlük güvenlik teknolojilerinde ve görüntü işlemenin olduğu hemen her alanda kullanılmaktadır.

Nesne tanıma için çeşitli yöntemler kullanılmaktadır ve bu yöntemleri çeşitli sınıflara ayırabiliriz. Bu yöntemlerin bazıları nesnenin görünümü tabanlı olup köşe veya kenar tespiti, damlacık tespiti, çıkıntı tespiti gibi genellikle basit yöntemlerdir. Bu tür yöntemler nesnenin fotoğraftaki büyüklüğüne, duruşuna, ışıklandırmaya ve görüş açısına göre başarımlarını göstermektedir. Bu yöntemler genel olarak robot kolları üzerinde kullanılmaktadır. Bunun nedeni, robot kollarının görüntüyü almak için uygun pozisyonu alıp görüntüde oluşabilecek ölçek, konum, duruş ve ışıklandırma değişiklikleri gibi etkenlerden etkilenmemesini sağlamasıdır. Ayrıca yine bu yöntemler biyolojik görüntüleme ve haritalamada damar yollarının oluşturulması ve tespit edilmesi gibi ışıklandırmanın, konumlanmanın iyi yapıldığı durumlarda rahatlıkla kullanılabilir [9][10]. Haritalarda yolların ve rotaların çıkarılması da yine ışıklandırmadan etkilenmeden uzaydan çekilmiş standart fotoğraflar üzerinden kolayca yapılabilir. Özetle bu tip yöntemler, ölçeğin, açısal değişikliklerin ve ışıklandırmanın fazla değişmediği durumlarda istenen sonuçları verebilmektedir.

Nesnenin özelliklerinin taban alındığı yöntemler ise görüntüden tanımlayıcı oluşturma üzerinde yoğunlaşmış, karmaşık ve matematik yoğunluğu daha yüksek yöntemlerdir. Bu tür yöntemler genellikle, nesnenin özgün özellikleri çıkarıp, bunları diğer resimlerdeki nesne görüntüsündeki özelliklerle eşleştirmeye dayalıdır. Bu tek-

nikler daha çok ışıklandırmanın sabit olmadığı, kamera açısının düzgün ayarlanmadığı ve fotoğrafın boyutunun belli olmadığı durumlarda iyi sonuçlar vermektedir [11] [12] [13] [14] [15]. Diğer yöntemlere göre bu yöntemler daha güvenilir sonuçlar verdiği için diğer yöntemlerin kullanıldığı durumlarda da kullanılabilir. Bu yöntemlerden çıkarılan özellikler sadece nesne tanıma için kullanılmayıp, diğer görüntü işleme algoritmaları için de ortam hazırlamakta; yüz tanıma, parmak izi tanıma, imza tanıma, nesne takibi, görüntü birleştirme, 3B görüntü oluşturma, hareket tespiti ve bunlar gibi bir çok uygulamada da faydalanılmaktadır [3][16].

Görüntü üzerindeki belirli piksellerden özellik tanımlamak için bir çok yöntem vardır. Bu yöntemlerin en çok bilinenleri Ölçek Değişimsiz Özellik Dönüşümü (SIFT), Hızlandırılmış Gürbüz Özellikler (SURF), Gradyan Konumu ve Yönelimi Histogramı (GLOH) ve Gradyan Yönelimleri Histogramı (HOG)'dır. Yöntemler genel olarak görüntüdeki belirli piksellerin etrafında bir çerçeve çizerek bu alandan özellik tanımlayıcı oluşturmayı amaçlamaktadır. Bunlardan GLOH yöntemi SIFT yöntemiyle benzerlik taşıyıp, son aşamasına kadar farklı bir yöntem içermemektedir [12], ancak son aşamada tanımlayıcıyı oluştururken farklı bir yöntem izlenmektedir. HOG ise, diğer yöntemlerde uygulanan süzgeçlemeleri kısmen atlayıp, görüntüdeki piksellerin gradyanlarının çıkarımını esas almakta ve belirli piksellerde dairesel ve dikdörtgen kümelere yararlanarak tanımlayıcı oluşturmaktadır. HOG yönteminin geliştirilme amacı, bir görüntüdeki insanların hareketlerinin algılanması ve tanımlanmasıdır [13]. SURF yöntemi bunların içlerinde en yeni yöntem olup, tanımlayıcı oluştururken işlemlerin hızlandırılması için farklı yaklaşımlar kullanmaktadır [14]. Bu yöntemde tanımlayıcı vektörlerin özetleri alınmakta, işlem böylece hızlandırılmaktadır, ancak çok yeni bir yöntem olduğundan üzerinde fazla araştırma yapılmamış ve bu yüzden kullanımı tercih edilmemiştir. SIFT, bu yöntemlerin içerisinde en yaygın kullanılanlardan ve üzerinde en fazla araştırma ve geliştirme yapılanlardan birisidir [3] [11] [16] [17] [18] [19] [20] [21]. Aynı zamanda SIFT en kararlı özellik tanımlayıcılardan biridir. Bu açıdan, bu yöntemden elde edilen özellik tanımlayıcıları bir çok değişik uygulamada rahatlıkla kullanılabilir. Bu yüzden SIFT yöntemi üzerinde çalışmanın uygun olacağı öngörülmüştür.

Bu yöntemin her aşamasında görüntüye belirli işlemler uygulanır. Öncelikle çıkarılacak olan özellik noktalarında ölçeğe karşı değişimsizlik sağlanıp; sonrasında çıkartılmış özellik noktalarının döndürmeye değişimsizliği sağlanır. Daha sonra dön-

dürmeye deęişimsizlięi saęlanmış özellik noktalarından tanımlayıcı oluşturulurak bu tanımlayıcıların üç boyut duruş ve ışıklandırmadan etkilenmesi en aza indirgenir. Bu yöntem sırasıyla görüntünün filtrelenmesi, filtrelenmiş görüntülerden anahtar nokta adayları çıkarılması, olumsuz etkisi olan noktalarının elenip kesin anahtar noktalarının işaretlenmesi, anahtar noktalarının yöneliminin bulunması ve bu noktalarda tanımlayıcı vektörlerin oluşturulması aşamalarından meydana gelir [11].

Bulunan anahtar nokta tanımlayıcı vektörleri, başka bir görüntü veya daha önce görüntülerden oluşturulmuş veri tabanındaki vektörlerle karşılaştırılıp, Öklit uzaklığı en az olan noktalar bir elemeyden geçerek birbirleriyle eşlenmektedir. Eşlenmiş tanımlayıcılar üç veya daha fazla olduğunda nesnenin görüntüdeki varlığı ve yeri yüksek olasılıkla tespit edilmektedir [11] [22].

Bu teknik kullanılarak genel olarak eşleşme adına çok iyi başarımlar elde edilmektedir. Bu işlemler üzerinde çeşitli uygulamalar ve geliştirmeler yapılmıştır [17] [18] [19] [21] [23]. Bu geliştirmeler genel olarak eşleşme başarımını artırmak, farklı uygulamalarda kullanmak adınadır. Ancak bir çok sistemde eşleşme başarımı kadar eşleşme hızı da önemli olmaktadır. Eşleşme süresi, özellikle sanayide veya güvenlik sistemlerindeki gerçek zamanlı veya gerçek zamana yakın sonuçlar gerektiren uygulamalarda önem kazanmaktadır.

SIFT yönteminde tanımlayıcı vektörler büyük olduğundan, bu vektörlerin eşleştirilmesi için çeşitli yöntemler önerilmiştir [11] [20]. Bunlardan bazıları verimsiz, bazıları koşula baęlı olarak verimli eşleştirme yöntemleridir. Bu çalışmada eşleşme başarımından ödün vermeden farklı uygulama koşulları için farklı yöntemlerin hız başarımları karşılaştırılmış, bu yöntemlerin verimli olduğu koşullar gösterilmiştir. Ayrıca basit yöntemleri daha ön plana çıkaracak paralel işlemci kullanımıyla hız iyileştirmesi yapılmıştır.

Bu tez çalışmasında, Bölüm 2'de kısımda resimlerden bilgi çıkarımı yöntemleri hakkında ön bilgi verilmiştir, Bölüm 3'te SIFT yöntemi detaylı olarak açıklanmış, eşleşme için gerekli tanımlayıcı vektörlerin çıkarımı anlatılmıştır. Bölüm 4'te SIFT yöntemiyle çıkarılmış tanımlayıcı vektörlerin eşleştirmesi için kullanılan yöntemlerden bahsedilmiş, bu yöntemlerden çalışmalarda [11] [22] [23] kullanılan yöntem olan Eniyi İlk Küme (BBF) yöntemiyle, daha önce çalışmalarda üzerinde durulmayan Basit Enyakın Komşu Araması (NNS) yönteminin arama ve eşleme zamanları karşılaştırılarak

benzetimleri yapılmıştır. Bölüm 5'te ise, sonuçlar hakkında değerlendirmelerde bulunulmuştur.

2. RESİMLERDEN BİLGİ ÇIKARIMI VE YÖNTEMLERİ

Nesneleri tespit etmek ve tanımlamak için yöntemler iki ayrı ana başlık altında toplanmıştır. Görüntüden basit tanımlayıcılar çıkaran yöntemler ve görüntünün ilgi noktalarını bulup bunlardan tanımlayıcı oluşturan yöntemlerdir. Bu bölümde bu yöntemler kısaca tanıtılacak ve bunlar hakkında genel bilgiler verilecektir. Bu yöntemler ayrıca kullanılan yöntemler açısından ön bilgi olacak ve bu yöntemlerden bazıları kullanılan yönteme ek olarak kullanıldığında çeşitli geliştirmeler sağlayabilecektir.

2.1 Görünüm Tabanlı Özellik Tespitleme Yöntemleri

Özellik tespitleme, bir görüntüdeki şekillerin özel noktalarını veya özel noktalar kümelerinin ortaya çıkarılıp, karşılaştırılması için uygun işlemlerin yapılmasıdır. Bir görüntüde bir çok özellik tespiti yapılabilir ve bu özellikleri tespit etmek için bir çok farklı yöntem geliştirilmiştir. Bu bölümde yapılacak her türlü örnek sonuç için (Şekil 2.1) kullanılacaktır.



Şekil 2.1. İşlemlerin Uygulandığı Orijinal Görüntü

2.1.1 Kenar Tespiti

Bilgisayarla görme ve görüntü işlemede, kenar tespiti, gri ölçekteki görüntüdeki kaydadeğer değişikliklerinin yerini belirlemede ve bunlardan kaynaklı fiziksel olayların tanımlanmasıyla ilgilenmektedir. Bir görüntüye kenar tespiti uygulamak, görüntüde işlenmesi gereken önemli bilgileri korumakta, daha az önem arz eden bilgileri süzmekte ve işlem yükünü azaltmaktadır. Bu bilgiler 3B yapılamada, hareket, tanıma, görüntü büyütme ve yenileme, görüntü tescilleme, görüntü sıkıştırma gibi bir çok uygulamada çok işe yaramaktadır.

İdeal durumda, görüntüye uygulanan kenar tespitleyici algoritma, görüntüdeki her nesnenin ve nesnenin üstündeki sınır çizgilerini belirlemesi gerekir. Ancak her türlü durumda iyi başarımlar gösterecek ve kullanım alanına göre gelecek süreçlerin hepsi için genel bir kenar tespiti algoritması zordur. Bundan dolayı matematiksel ve algoritmik olarak farklı bir çok kenar tespiti yöntemi geliştirilmiştir. Bunlardan en yaygın kullanılanları Roberts, Prewitt, Sobel ve Canny kenar tespit yöntemleridir.

2.1.1.1 Roberts Çapraz Kenar Tespitleyicisi

Roberts Çapraz kenar tespitleyicisi ilk kenar tespitleyici yöntemlerden birisidir. Ayrık fark yöntemini çapraz noktalar üzerinde uygulamasıyla (2.1) görüntünün yaklaşık gradyanı alınmasıyla çalışır [24].

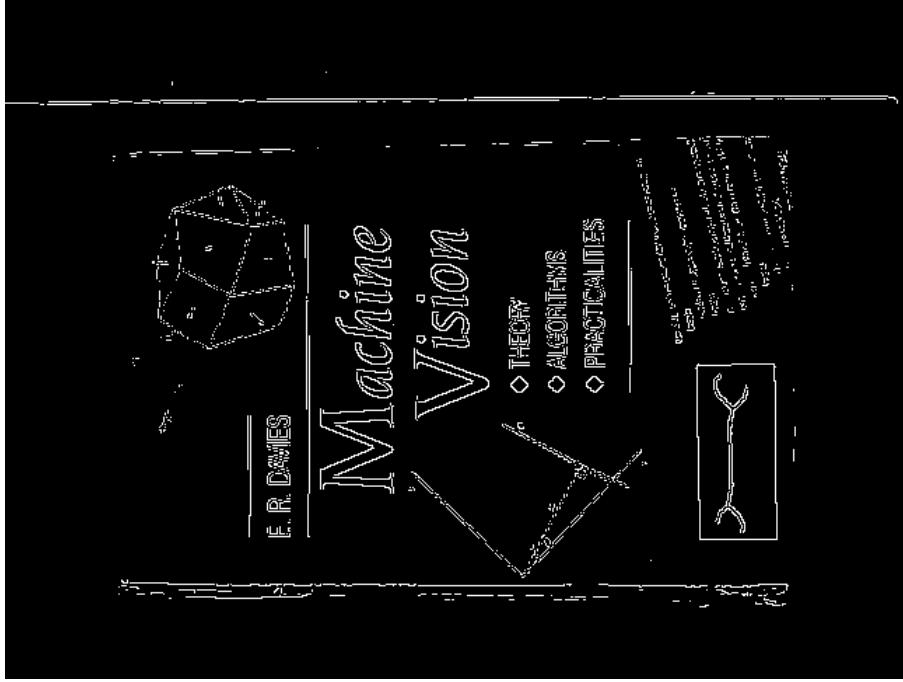
I görüntünün gri ölçekteki şiddeti, G_c ve G_r sırasıyla görüntünün yatay ve dikey gradyanları ve $*$ evrişim işlemi olmak üzere,

$$G_c = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * I \quad \text{ve} \quad G_r = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} * I \quad (2.1)$$

G_c ve G_r bulunur, bulduğumuz yatay ve dikey yöndeki gradyan görüntülerinin boyu bulunarak kenar tespitine uygun görüntü elde edilir(2.2).

$$\nabla I = G = \sqrt{G_c^2 + G_r^2} \quad (2.2)$$

Bulunan gradyan büyüklüğü görüntüsü Otsu yöntemiyle çıkarılan [25] eşik değerinden geçirilerek kenar tespiti yapılır (Şekil 2.2).



Şekil 2.2. Roberts Çapraz Yöntemiyle Tespit Edilmiş Kenar Görüntüsü

2.1.1.2 Prewitt Kenar Tespitleyicisi

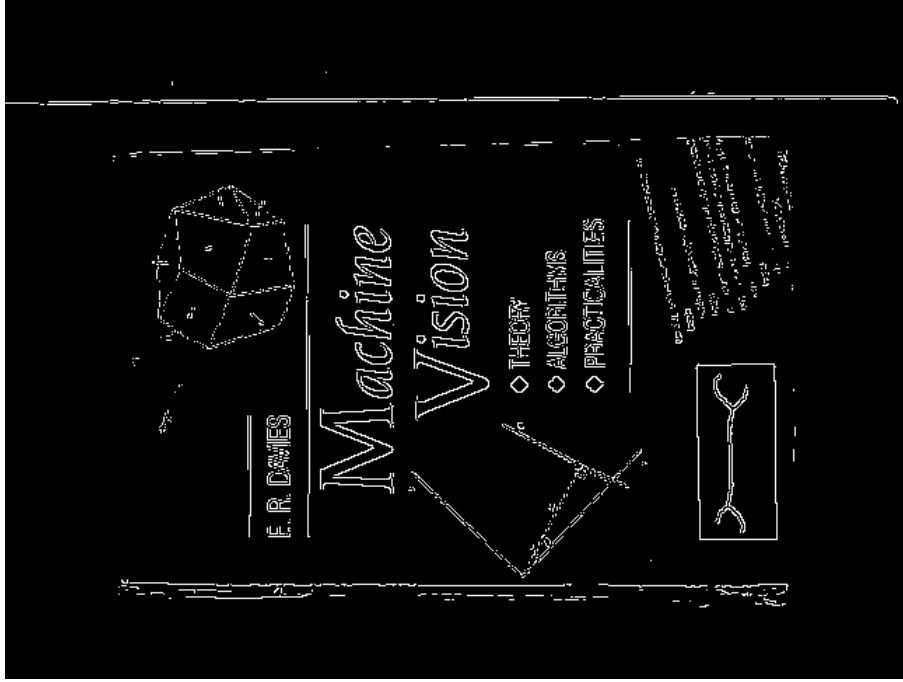
Prewitt, teknik olarak ayırık fark alma işlecidir; basit anlamda, görüntü şiddetinin her noktada yaklaşık gradyanını hesaplamaktadır (2.3). Bu da görüntü içerisinde açığa çıkaran koyuya en büyük değişimi vermektedir [26].

I gri ölçekteki görüntü, G_c ve G_r sırasıyla görüntünün yatay ve dikey gradyanları olmak üzere ,

$$G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * I \quad \text{ve} \quad G_c = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * I \quad (2.3)$$

$$G = \sqrt{G_c^2 + G_r^2} \quad (2.4)$$

Yatay ve dikey yöndeki gradyanların büyüklük toplamı (2.4) G , kenar tespiti için uygun görüntüyü oluşturur. Bu görüntü, görüntüden çıkarılan Otsu eşik değerinden [25] geçirilerek kenarlar tespit edilir (Şekil 2.3).



Şekil 2.3. Prewitt Yöntemiyle Tespit Edilmiş Kenar Görüntüsü

2.1.1.3 Sobel Kenar Tespitleyicisi

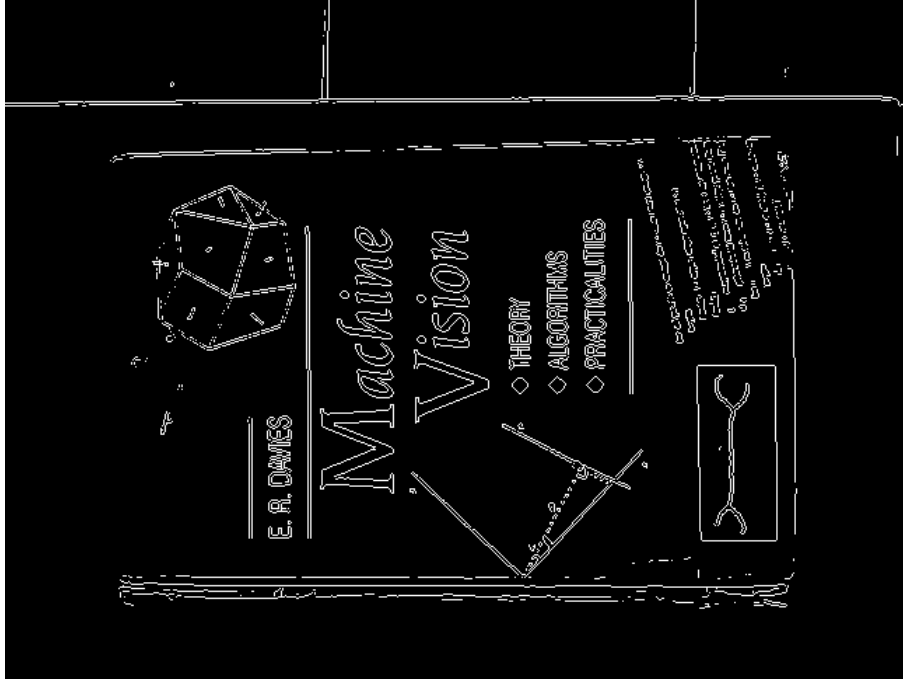
Sobel işleci de piksellerin ayırık farkını bulmak için kullanılır; görüntü şiddetinin her noktada yaklaşık zıt gradyanı hesaplanır (2.5). Sobel işleci görüntü şiddetindeki sürekli değişimlerde tepki vermeyip, sürekliliğin kesildiği noktalarda daha yüksek tepki verecektir [27].

I gri ölçekteki görüntü, G_c ve G_r sırasıyla görüntünün yatay ve dikey gradyanları olmak üzere,

$$G_c = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \quad \text{ve} \quad G_r = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \quad (2.5)$$

$$G = \sqrt{G_c^2 + G_r^2} \quad (2.6)$$

Prewitt tekniğinde olduğu gibi yatay ve dikey yöndeki gradyanların büyüklük toplamı (2.6) G , kenar tespiti için uygun görüntüyü oluşturur. Bu görüntü, görüntüden çıkarılan Otsu eşik değerinden [25] geçirilerek kenarlar tespit edilir (Şekil 2.4).



Şekil 2.4. Sobel Yöntemiyle Tespit Edilmiş Kenar Görüntüsü

2.1.1.4 Canny Kenar Tespitleyicisi

Canny kenar tespitleyicisi, çok aşamalı bir algoritma kullanarak görüntüde geniş aralıktaki kenarları tespit etmek için kullanılır [28].

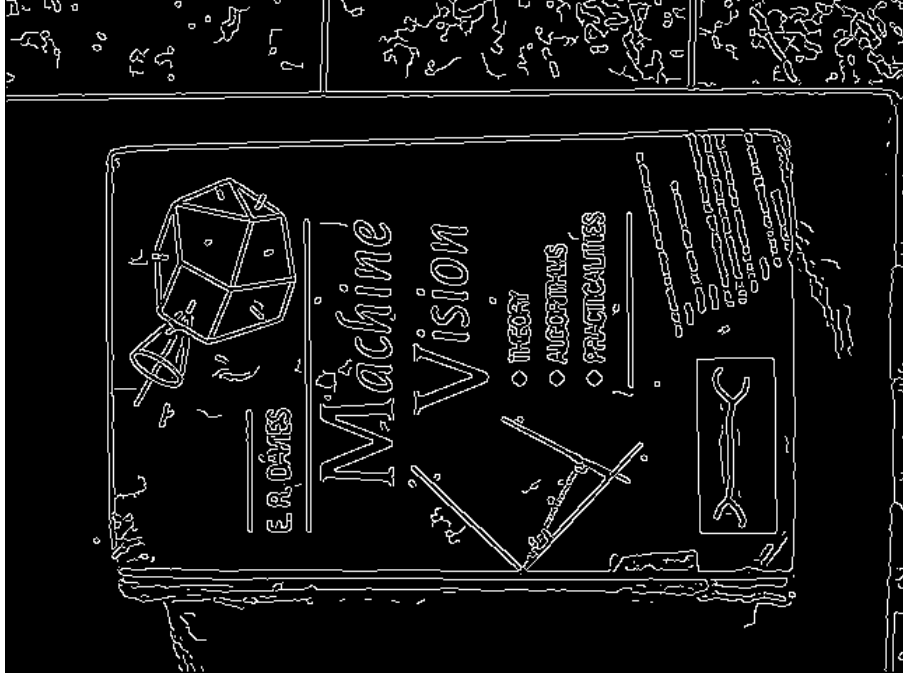
Canny öncelikle görüntüdeki gürültüyü azaltmak amacıyla görüntüyü orta derece ($\sigma = 1.4$) bir Gauss süzgecinden geçirir, böylece görüntü yumuşatılmış olur.

Sonra ayırık farkları almak için bilinen bir işleç kullanılır. Bulunan gradyandan, noktaların açısız yönelimi bulunarak (2.7) noktalar 4 kümeye ayrılır. Ayrılan kümelerde, kenarlar birbirini takip edecek şekilde tespit edilebilir. Bunun amacı tespit edilen sınır çizgilerinin daha ince olmasını sağlamaktır.

$$G = \arctan\left(\frac{G_r}{G_c}\right) \quad (2.7)$$

Daha sonra diğer yöntemlerden farklı olarak bulunan esas görüntünün histogramı çıkarılıp, bu histogramın ortadeğerinin 1.33 ve 0.66 katları veya bu değerlere yakın olan iki eşik değerinden geçirilmektedir. Bunun amacı birkaç gürültülü noktayı elemektir. Önce yüksek eşik değeri kullanılarak görüntüdeki gürültülü noktalar elen-

mekte ve bulunan kenarlar gradyanların açılal yönelimlerinin yardımıyla takip edil-mektedir. Sonrasında düşük eşik değeri kullanılarak takip edilen kenarlarda zayıf kalmış kısımlar güçlendirilerek kenar tespiti yapılmaktadır (Şekil 2.5).



Şekil 2.5. Canny Yöntemiyle Tespit Edilmiş Kenar Görüntüsü

2.1.2 Köşe Tespitleyicisi

Köşe tespitleyici, bir görüntünün belirli özelliklerini ve görüntü içeriğinden anlam çıkarımı için kullanılan bir yaklaşımdır. Köşe tespiti genellikle hareket algılama, görüntü kayıtlama, video takibi, görüntü mozaikleme, görüntü tümleştirme, 3B modelleme ve nesne tanıma için görüntü büyüklüğünün, yöneliminin, ışıklandırmasının ve diğer özelliklerin değişmediği durumlarda kullanılabilir.

Bir görüntüde köşe farklı yönlerdeki iki baskın kenarın bir araya geldiği nokta olarak tanımlanabilir. Pratikte köşe tespitleyici yöntemler ilgi noktalarının bir çoğunu tespit ederler, bunun bir sonucu olarak köşeleri tespit etmek için bu noktaların hangilerinin gerçekten köşe olduğu bölgesel analizle bulunabilir.

Köşe tespit ediciler genellikle gürbüz olmayıp, iyi bir denetim gerektirir. Köşe tespitleyicinin kalite katsayılarından biri de değişik ışıklandırma, öteleme, döndürme ve diğer

dönüşümlere maruz kalmış benzer görüntülerde aynı köşeyi tespit edebilme yeteneğidir. Köşe tespitleme için Moravec, Harris-Stephens, Shi-Tomasi gibi bir çok değişik algoritma ve yöntem geliştirilmiştir [29] [30] [31].

2.1.2.1 Moravec Köşe Tespitleme Algoritması

Moravec en eski köşe tespitleyici algoritmalarından bir tanesidir. Algoritma bütün piksellerin tek tek test edilip, pikselin yakınında üst üste gelen büyük bölgelerden ne kadar benzer bir bölgenin merkezlendiğinin değerlendirilmesine dayanır. Benzerlik, iki bölge arasında farkların karelerinin toplamı alınarak ölçülür ve düşük değer daha çok benzerlik olduğunu gösterir.

Bir pikselin köşe olabilmesi, bölge ile komşu arasındaki en küçük farkların kareleri toplamıyla ölçülür. Eğer bu değer bölgesel olarak enbüyükse, ilgi noktası var demektir [29].



Şekil 2.6. Moravec Yöntemiyle Tespit Edilmiş Köşe Noktaları

2.1.2.2 Harris Stephens, Shi-Tomasi Köşe Tespitleyici Algoritmaları

Bu algoritma, Moravec köşe tespitleyicisinin, köşe değerlerini üst üste getirmek yerine, doğrudan yöne göre alınan farkı değerlendirerek Harris ve Stephens tarafından, geliştirilmiştir. Bu algortmada köşe değerleri görüntünün özilintisi olarak değerlendirilir [30].

I gri ölçekteki görüntümüz olmak üzere; görüntüdeki bir pencere $\omega(cu, ru)$ ve onun (c, r) kadar kaydırılmış hali arasında farkların karelerinin toplamı S olmak üzere,

$$S(c, r) = \sum_{cu} \sum_{ru} \omega(cu, ru) (I(cu + c, ru + r) - I(cu, ru))^2 \quad (2.8)$$

$I(cu+c, ru+r)$ Taylor açılımıyla şu halde gösterilebilir:

$$I(cu + c, ru + r) \approx I(cu, ru) + I_c(cu, ru)c + I_r(cu, ru)r \quad (2.9)$$

Bu yaklaşımla birlikte, S eşitliği (2.10) yaklaşımına dönüştürülebilir.

$$S(c, r) \approx \sum_{cu} \sum_{ru} \omega(cu, ru) (I_c(cu, ru)c + I_r(cu, ru)r)^2 \quad (2.10)$$

Eşitlik (2.12)'te \mathbf{A} yapı matrisi olmak üzere, S yaklaşımı matris formunda şöyle gösterilebilir:

$$S(c, r) \approx \begin{bmatrix} c & r \end{bmatrix} \mathbf{A} \begin{bmatrix} c \\ r \end{bmatrix}, \quad (2.11)$$

$$\mathbf{A} = \sum_{cu} \sum_{ru} \omega(cu, ru) \begin{bmatrix} I_c^2 & I_c I_r \\ I_c I_r & I_r^2 \end{bmatrix} = \begin{bmatrix} \langle I_c^2 \rangle & \langle I_c I_r \rangle \\ \langle I_c I_r \rangle & \langle I_r^2 \rangle \end{bmatrix} \quad (2.12)$$

Bulunan bir Harris matrisi olup, açılı parantezler ortalamayı göstermektedir. Eğer dairesel bir pencere kullanılırsa, tepki eşyönlü olur. Bir köşe genellikle (c, r) vektörünün her yönünde S 'in değişmesiyle karakterize edilir. \mathbf{A} matrisi analiz edilerek, bu işlem şöyle tanımlanabilir:

\mathbf{A} 'nın bir ilgi noktası olması için iki büyük özdeğere sahip olması gerekir. Özdeğerlerin, λ_1 ve λ_2 , büyüklüklerine bakılarak şu çıkarımlar yapılabilir:

1. Eğer $\lambda_1 \approx 0$ ve $\lambda_2 \approx 0$ ise (c,r) pikselinde bir ilgi noktası yoktur.
2. Eğer $\lambda_1 \approx 0$ ve λ_2 herhangi büyük bir değere sahipse, bir kenar bulunmuştur.
3. Eğer λ_1 ve λ_2 büyük değerlere sahipse, bir köşe tespit edilmiştir.

Özdeğerlerin tam hesaplanması zorluğu yüksek olduğu için Harris ve Stephens eşitlik (2.13)'deki yöntemi tavsiye etmişlerdir [30].

$$M_c = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 = \det(A) - \kappa \text{trace}^2(A) \quad (2.13)$$

Böylece, algoritma köşeleri bulmak için **A** matrisini ayrıştırmak yerine, **A** matrisinin belirteç ve kalıntıları kullanılabilir.

Shi-Tomasi köşe tespitleyicisi ise doğrudan λ_1, λ_2 'in en küçüğünü hesaplayarak daha kararlı sonuçlar bulmaktadır. κ değeri 0.04-0.15 arasında kullanılması deneysel olarak uygun bulunmuştur [31].



Şekil 2.7. Harris Yöntemiyle Tespit Edilmiş Köşe Noktaları



Şekil 2.8. Shi-Tomasi Yöntemiyle Tespit Edilmiş Köşe Noktaları

2.1.2.3 Foerstner Köşe Tespitleyicisi

Foerstner algoritması bir pencere içerisindeki köşenin bütün teğetlerini bularak, pencere içerisinde tespit edilmiş köşenin arapiksel duyarlılığında bulunmasını sağlamaktadır. Algoritma ideal köşenin sadece bir noktada olduğu olgusu üzerine geliştirilmiştir [32].

$\nabla I(\mathbf{x}') = [I_x, I_y]^T$, I görüntüsünün \mathbf{x}' 'deki gradyan vektörü olmak üzere, \mathbf{x}' pikselindeki teğet çizgisi denklemi şöyledir:

$$T_x(\mathbf{x}) = \nabla I(\mathbf{x}')^T (\mathbf{x} - \mathbf{x}') = 0 \quad (2.14)$$

N penceresinde bütün teğetlere en yakın \mathbf{x}_0 noktası eşitlik(2.15)'daki şekilde ifade edilir.

$$\mathbf{x}_0 = \arg \min_{\mathbf{x} \in \mathbb{R}^{2 \times 2}} \int_{\mathbf{x}' \in N} T_{\mathbf{x}'}(\mathbf{x})^2 d\mathbf{x}' \quad (2.15)$$

Teğetlerden \mathbf{x}_0 noktasına olan mesafe gradyan büyüklüğü ile ağırlıklandırılarak, kuvvetli gradyanlardan geçen teğetlere daha çok önem verilmesi sağlanır. Böylece \mathbf{x}_0

şu şekilde ifade edilebilir:

$$\mathbf{x}_0 = \arg \min_{\mathbf{x} \in \mathbb{R}^{2 \times 2}} (\mathbf{x}^T \mathbf{A} \mathbf{x} - 2 \mathbf{x}^T \mathbf{b} + c) \quad (2.16)$$

$\mathbf{A} \in \mathbb{R}^{2 \times 2}$, $\mathbf{b} \in \mathbb{R}^{2 \times 1}$ ve $c \in \mathbb{R}$ şöyle tanımlanmaktadır:

$$\begin{aligned} \mathbf{A} &= \int \nabla I(\mathbf{x}') \nabla I(\mathbf{x}')^T d\mathbf{x}' \\ \mathbf{b} &= \int \nabla I(\mathbf{x}') \nabla I(\mathbf{x}')^T \mathbf{x}' d\mathbf{x}' \\ c &= \int \mathbf{x}'^T \nabla I(\mathbf{x}') \nabla I(\mathbf{x}')^T \mathbf{x}' d\mathbf{x}' \end{aligned}$$

Eşitlik (2.16) denkleminin enküçük değeri \mathbf{x}' e göre türevinin 0 olduğu noktadır:

$$2\mathbf{A}\mathbf{x} - 2\mathbf{b} = 0 \quad (2.17)$$

Eşitlik (2.17)'nin çözümünün olması için \mathbf{A} terslenebilir olması gerekir, ve çözüldüğünde,

$\mathbf{x}_0 = \mathbf{A}^{-1} \mathbf{b}$ eşitliğinin var olması, N penceresinde gerçek bir köşenin varlığıyla gerçekleşir.

2.1.2.4 Wang ve Brady Köşe Tespitleyici Algoritması

Wang ve Brady tespitleyicisi, görüntüyü bir yüzey olarak dikkate almaktadır ve görüntü üzerinde uzun eğrilerin bulunduğu bölgelere bakıp bu eğrilerin ani yön değişikliği yaptığı yerler belirlenir [33]. Eşitlik (2.18) de c , tespitleyicinin kenarlara ne kadar tepki vereceğini belirler.

$$C = \nabla^2 I - c |\nabla I|^2 \quad (2.18)$$

2.1.2.5 SUSAN Köşe Tespitleyicisi

Köşe tespiti için SUSAN (Enküçük Tek Değerli Parça Özümleyen Çekirdek), test edilen pikselin etrafında dairesel bir maskeleme yapar. M maskeleme bölgesi olmak üzere, bölge içerisindeki piksel $m \in M$ olarak gösterilir. Çekirdek m_0 noktasındadır. Bölgedeki bütün pikseller çekirdekle karşılaştırılır (2.19).

$$c(\vec{m}) = e^{-\left(\frac{I(\vec{m}) - I(\vec{m}_0)}{t}\right)^6} \quad (2.19)$$

t yarıçapı ifade etmek üzere, e sayısının üssünün çarpanı deneysel olarak bulunmuştur. Bu fonksiyon yumuşatılmış dikdörtgen fonksiyon görünümündedir. SUSAN'ın alanı şöyle verilmektedir:

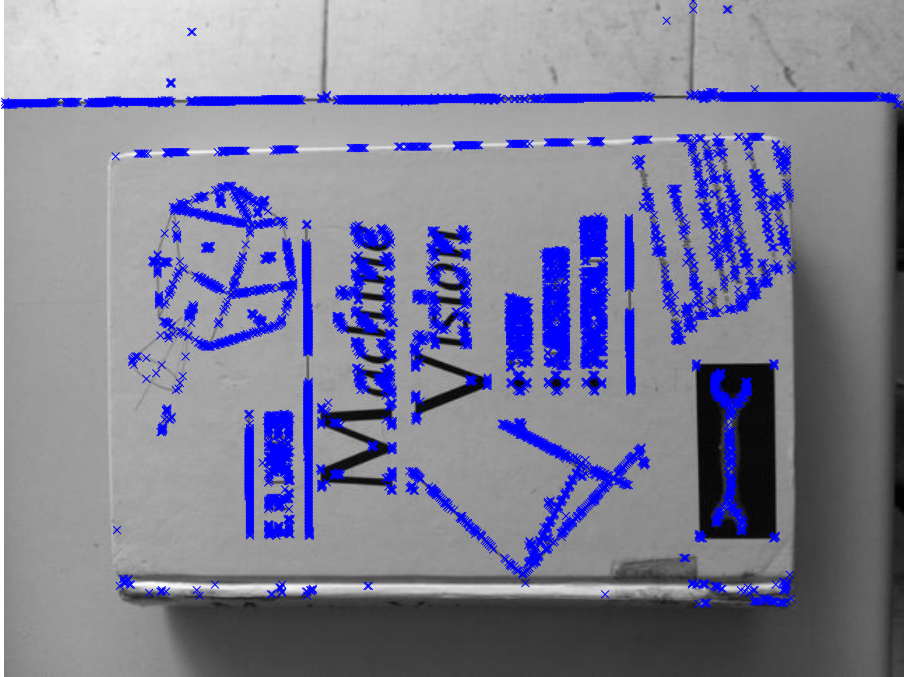
$$n(M) = \sum_{\vec{m} \in M} c(\vec{m})$$

Eğer c dikdörtgen fonksiyonsa, n , çekirdekten t kadar uzaklıktaki bölgedeki piksel sayısını verir. SUSAN işlecinin köşe tepkisi g geometrik eşik olmak üzere şöyle verilmektedir:

$$R(M) = \begin{cases} g - n(M), & \text{if } n(M) < g \\ 0 & , \text{ otherwise} \end{cases}$$

Eğer bölge yeterince küçük seçilirse, SUSAN işleci sadece pozitif sonuç verir ve bu da eksiksiz SUSAN işlecidir.

Köşe tespiti için iki aşama daha kullanılmaktadır. Önce, SUSAN bölgesinin merkezi bulunur; uygun bir köşenin çekirdekten uzak bir merkezi vardır. Sonra SUSAN bölgesinde, çekirdekten merkeze doğru olan bütün çizgiler ile maskenin kenarındaki tüm noktalar SUSAN'ın içinde sayılır ve köşe tespit edilebilir [34].



Şekil 2.9. SUSAN Yöntemiyle Tespit Edilmiş Köşe Noktaları

2.1.2.6 Trajkovic ve Hedley Köşe Tespitleyici

SUSAN'a benzer olarak, bir piksele bağlı bir bölgenin komşu piksellerine bakılarak kendi benzeri olduğu doğrudan test edilir. \vec{c} dikkate alınan piksel, $\vec{p} \in P$, \vec{c} merkezli P dairesindeki bir nokta, \vec{p}' çap boyunca \vec{p} 'nin karşısında olmak üzere, köşe tepkisi fonksiyonu şöyle tanımlanır:

$$r(\vec{c}) = \min_{\vec{p} \in P} ((I(\vec{p}) - I(\vec{c}))^2, (I(\vec{p}') - I(\vec{c}))^2)$$

Bu fonksiyon, merkez piksel çap boyunca kendine yakın olan iki piksele benzer olduğu durumda, bir yönelim oluşmuyorsa, büyük değer almaktadır [35].

2.1.3 Damlacık Tespiti

Damlacık tespiti, görüntüdeki noktaların veya bölgelerin, parlaklık ve renk değişimi gibi bazı özelliklerinin çevresindekilere göre farklılıklarını bulmayı amaçlayan görsel modüller olarak ifade edilir.

Damlacık tespitleyiciler, bazı bölgelerden kenar ve köşe tespitleyicilerinin çıkaramadığı, onları tamamlayan bilgiler çıkarmayı amaçlar. İlk çalışmalarda [36], damlacık tespitleyiciler daha ilerideki süreçlerde kullanılmak üzere ilgi bölgelerini çıkarmak için kullanılmıştır. Nesne tanıma ve takip yöntemlerine uygulandığında, bu bölgeler, bir nesnenin veya bir parçanın görüntü içerisinde varlığını işaret edebilmektedir. Görüntünün histogramı uygulamalarında, bölümlene uygulamasıyla birlikte, enuç tespiti için kullanılabilir. Doku analizi ve tanımda, ana yapıyı çıkarmak için kullanılmaktadır. Daha güncel çalışmalarda, görünüm tabanlı nesne tanıma için bilgi verici özelliklerin varlığını belirlemek için popüler bir kullanım alanı kazanmıştır.

Damlacık tespitleyicilerin iki ana türü vardır:

1. Türev ifadelerine dayanan diferansiyel yöntemler
2. Şiddet boyutunda bölgesel enuç noktalara dayanan yöntemler

2.1.3.1 Gauss'un Laplace İşleci (LoG)

İlk ve en çok kullanılan damlacık tespitleyiciler, Gauss'un Laplace işleci tabanlıdır.

Verilen $f(c, r)$ görüntüsü belirli t ölçeğindeki bir Gauss çekirdeğiyle(2.20) eşitlik(2.21)'de

gösterilen ölçek-uzayında evriştirilir [37].

$$g(c, r, t) = \frac{1}{2\pi t} e^{-(c^2+r^2)/2t} \quad (2.20)$$

$$L(c, r, t) = g(c, r, t) * f(c, r) \quad (2.21)$$

Daha sonra Laplace işleci (2.22) uygulanarak, $\sqrt{2t}$ boyutunda koyu damlalar için güçlü pozitif tepkiler, aynı boyutta parlak damlacıklar için güçlü negatif tepkiler alınır.

$$\nabla^2 L = L_{cc} + L_{rr} \quad (2.22)$$

Bu işleci tek ölçeğe uygularken en büyük problem, işleç tepkisinin görüntü alanındaki damlacık yapısının büyüklüğü ile Gauss çekirdeğinin büyüklüğü arasındaki ilişkiye kuvvetle bağımlı olmasıdır. Bunun için, bilinmeyen büyüklükteki damlacıkların kendiliğinden yakalanması için çok ölçekli yaklaşım gereklidir.

Kendiliğinden ölçek seçimli, çok ölçekli damlacık tespitleyici elde etmenin en anlaşılır yolu, ölçek-normal Laplace işlecini(2.23) hesaba katmaktır.

$$\nabla_{norm}^2 L = t(L_{cc} + L_{rr}) \quad (2.23)$$

Ölçek-uzay en uç noktaların tespiti için eşitlik(2.23) sonucunun hem ölçeğe göre, hem de uzaya göre en uç noktaları bulunmaktadır.

Sonuç olarak, verilen $f(c, r)$ iki boyutlu sayısal görüntüden, üç boyutlu sayısal ölçek-uzay hacmi $L(c, r, t)$ hesaplanır ve bir nokta kendi ölçeğindeki, bir üst ve bir alt ölçekteki komşu pikselleriyle, toplam 26 pikselle karşılaştırılarak, hepsinden büyüğe en parlak, hepsinden küçüğe en koyu nokta olarak sayılır. (\hat{c}, \hat{r}) ilgi noktaları ve \hat{t} ölçeği $(\hat{c}, \hat{r}, \hat{t}) = \arg \max \min local_{(c,r,t)}(\nabla_{norm}^2 L(c, r, t))$ eşitliğine göre bulunur.

Normalize Laplace işlecinin ölçek-uzay uçları olarak tanımlanan damlaların bazı temel özellikleri, öteleme, döndürme ve yeniden ölçeklendirmeye eşdeğişken olmasıdır. Bu yüzden, eğer (c_0, r_0, t_0) noktasında bir ölçek-uzay ucu varsayılırsa, s çarpımıyla yeniden ölçeklendirme yapıldığında yeni uç (sc_0, sr_0, s^2t_0) noktasında olur. Pratikte Laplace işleci damlacık tespiti dışında ölçek-uyarlamalı özellik takibi, ölçek bağımsız özellik dönüşümü, köşe tespitleme gibi başka uygulamalarda çok faydalıdır [37].

2.1.3.2 Gaussların Farkı (DoG) Yaklaşımı

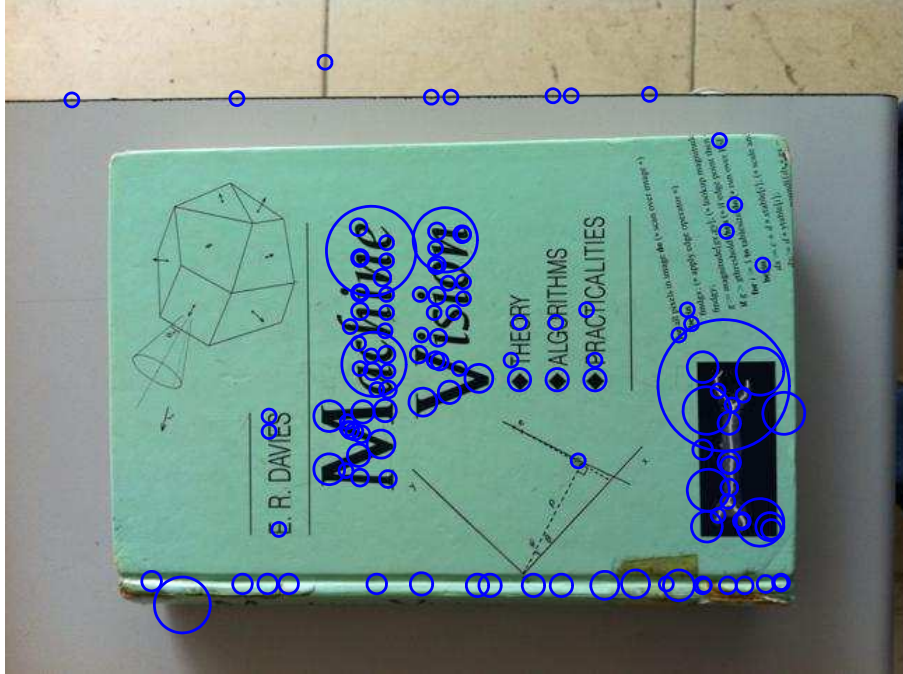
Ölçek-uzay gösterimi $L(c, r, t)$ yayılım denklemini (2.24) sağladığından, Gauss'un Laplace işleci (2.22) Gauss süzgecinden geçirilmiş iki görüntünün limit durumundaki farkları (2.25) olarak da hesaplanabilir [38].

$$\partial_t L = \frac{1}{2} \nabla^2 L \quad (2.24)$$

$$\nabla_{norm}^2 L(c, r, t) \approx \frac{t}{\Delta t} (L(c, r, t + \Delta t) - L(c, r, t - \Delta t)) = \lim_{k \rightarrow 1} \frac{L(c, r, k\sigma) - L(c, r, \sigma)}{k - 1} \quad (2.25)$$

Literatürde bu yaklaşım Gaussların farkı yaklaşımı olarak ifade edilir. Küçük teknik ayrıntılar dışında, bu işlem Laplace işleci yöntemine temel olarak çok yakındır. Bu yaklaşım, Gauss'un Laplace işleci kullanılması gereken, işlem yükünün azaltılmasının amaçlandığı, ölçek değişimsizlik gerektiren hemen her yöntem içerisinde fazla fark yaratmadan kullanılabilir [38].

2% → 20% , 22 steps



Şekil 2.10. Gaussların Farkıyla Damlacık Tespit Edilmiş Görüntü

2.1.3.3 Hessian Determinantı

Basitleştirilmiş biçimde, Haar dalgacıklarından oluşturulmuş ölçek-normal Hessian determinantı, SURF özellik tespitleyicide ana ilgi nokta işleci olarak kullanılmaktadır [14].

L_{cc} , L_{rr} ve L_{cr} Hessian matrisinde görüntünün belirli noktalarında ikinci dereceden türevleri temsil etmek üzere, ölçek-normal Hessian determinantı (2.26) hesaplanıp, sonra ölçek-uzay uç noktası bulunarak (2.27) kendiliğinden ölçek-seçmeli başka bir diferansiyel damlacık tespitleyici elde edilebilir [37].

$$\det HL(c, r, t) = t^2(L_{cc}L_{rr} - L_{cr}^2) \quad (2.26)$$

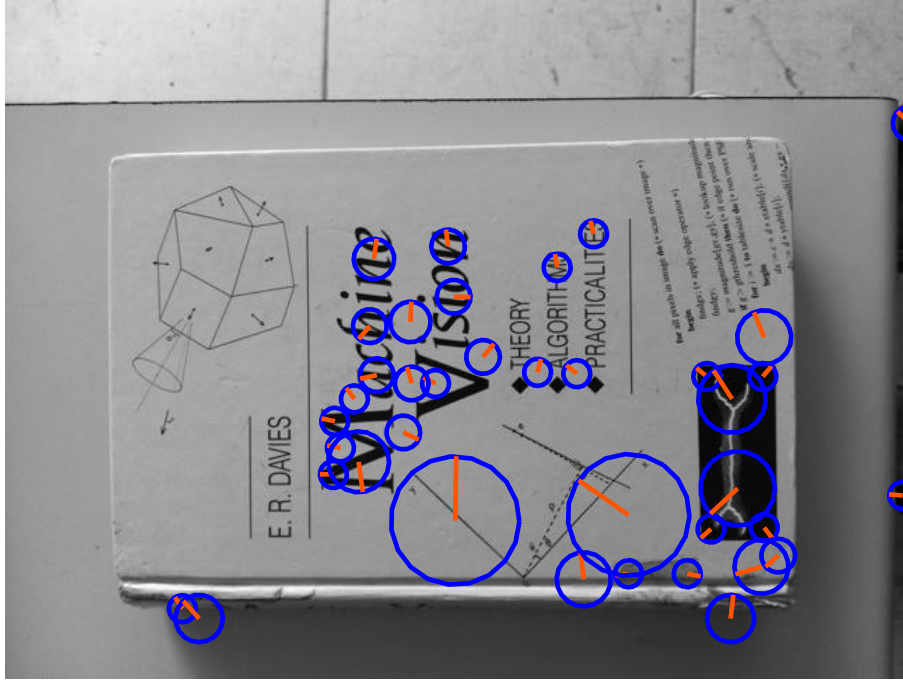
$$(\hat{c}, \hat{r}, \hat{t}) = \arg \max_{local_{(c,r,t)}}(\det HL(c, r, t)) \quad (2.27)$$

2.1.3.4 Melez Laplace İşleci ve Hessian Determinantı İşleci

Uzaysal seçimin Hessian determinantı (2.28) tarafından, ölçeksel seçimin ise Laplace işleci (2.29) tarafından yapıldığı melez bir işleçle damlacık tespiti yapılmaktadır [39].

$$(\hat{c}, \hat{r}) = \arg \max_{local_{(c,r)}}(\det HL(c, r, t)) \quad (2.28)$$

$$\hat{t} = \arg \max \min_{local_t}(\nabla_{norm}^2 L(\hat{x}, \hat{y}, t)) \quad (2.29)$$



Şekil 2.11. Hessian Determinantıyla Damlacık Tespit Edilmiş Görüntü

2.1.4 Çıkıntı Tespitleyici

Bir görüntünün çıkıntısı veya çıkıntı kümesi, görüntü fonksiyonunun en az bir boyutta bölgesel enbüyük noktaların oluşturduğu eğriler kümesidir. Daha genel olarak, N değişkenli bir fonksiyonun çıkıntısı, o fonksiyonun $N-1$ boyutta bölgesel enbüyüklerinin oluşturduğu eğriler kümesidir. Bu bakımdan, çıkıntılar kavramı bölgesel enbüyükler yaklaşımına genişletilir. Paralel olarak, girintiler kavramı da bölgesel enküçükler yaklaşımı olarak tanımlanabilir [9] [10].

Çıkıntılar ve girintiler kümelerinin birleşimi, ilişkili noktalar kümesiyle birlikte, fonksiyonun kritik noktalarında birleşen veya keşisen eğriler kümesi veya birleştirici küme olarak tanımlanır. Bu kümelerin birleşimi de, fonksiyonun göreceli kritik kümesi olarak tanımlanır. Girintiler kümesi, çıkıntılar kümesi ve göreceli kritik kümeler fonksiyona özgü önemli geometrik bilgiler verir. Girinti ve çıkıntıları tespit etme, görüntüdeki uzatılmış nesnelerin içkısımlarını çıkarmayı amaçlamaktadır.

Ölçek-uzay hesaplamaları Gauss çekirdeğiyle evrişim içerdiğinden, çok-ölçekli çıkıntı, girinti ve kritik noktalar görüntüde nesnelerin daha gürbüz tanımlanmalarına

izin verir. Böylelikle tek ölçekli görüntüdeki gürültüler azaltılabilmektedir.

Girinti ve çıkıntılar doğal ilgi noktalarının veya bölgesel uç noktalarının tamamlayıcısı olarak görülebilir. Uygun tanımlanan kavramlarla, girinti ve çıkıntılar uzaysal sınırları düzenlemek için ölçek bağımsız bir iskelet oluşturabilir. Girinti ve çıkıntılar genelde harita görüntülerinde yolları, retina görüntülerinde veya 3B manyetik rezonans görüntülerinde damarları tespit etmek için kullanılmaktadır.

2.1.4.1 Sabit Ölçekte Çıkıntılarının ve Girintilerin Diferansiyel Geometrik Tanımı

Daha önce de tanımlandığı gibi, L ölçek-uzay gösterimini, L_{cc} ve L_{rr} da eşitlik (2.22)'de tanımlanan Laplace işlecinin Hessian matrisinin (2.30) öz değerlerini temsil etmek üzere, Hessian matrisi şu şekilde tanımlanır:

$$H = \begin{bmatrix} L_{cc} & L_{cr} \\ L_{cr} & L_{rr} \end{bmatrix} \quad (2.30)$$

Bölgesel yönlü türev işleçlerine uygulanmasıyla oluşan koordinat döndürmesi ile p ve q 'nin yeni koordinatları temsil ettiği sistem(2.31) oluşturulur.

$$\partial_p = \partial_c \sin \beta - \partial_r \cos \beta, \partial_q = \partial_c \cos \beta + \partial_r \sin \beta \quad (2.31)$$

$\cos \beta$ ve $\sin \beta$ 'yi sırasıyla eşitlik (2.32) ve (2.33)'deki gibi seçersek, L 'nin karışık kısmi türevi L_{pq} döndürülmüş koordinatta sıfır olur.

$$\cos \beta = \sqrt{\frac{1}{2} \left(1 + \frac{L_{cc} - L_{rr}}{\sqrt{(L_{cc} - L_{rr})^2 + 4L_{cr}^2}} \right)} \quad (2.32)$$

$$\sin \beta = \text{sgn}(L_{cr}) \sqrt{\frac{1}{2} \left(1 - \frac{L_{cc} - L_{rr}}{\sqrt{(L_{cc} - L_{rr})^2 + 4L_{cr}^2}} \right)} \quad (2.33)$$

Böylece, çıkıntılarının formal diferansiyel geometrik tanımı sabit bir t ölçeğinde şöyle yapılabilir:

$$L_p = 0, L_{pp} \leq 0, |L_{pp}| \geq |L_{qq}|$$

Aynı şekilde girintilerin tanımı da şöyle gösterilebilir:

$$L_q = 0, L_{qq} \geq 0, |L_{pp}| \leq |L_{qq}|$$

(u, v) koordinat sistemine göre, v görüntü gradyanına paralel ve $\tan\alpha = \frac{L_r}{L_c}$ olmak üzere, sistem şöyle düzenlenebilir:

$$\partial_u = \partial_c \sin \alpha - \partial_r \cos \alpha, \partial_v = \partial_c \cos \alpha + \partial_r \sin \alpha \quad (2.34)$$

$$\begin{aligned} L_v^2 L_{uu} &= L_c^2 L_{rr} - 2L_c L_r L_{cr} + L_r^2 L_{cc} \\ L_v^2 L_{uv} &= L_c L_r (L_{cc} - L_{rr}) - (L_c^2 - L_r^2) L_{cr} \\ L_v^2 L_{vv} &= L_c^2 L_{cc} + 2L_c L_r L_{cr} + L_r^2 L_{rr} \end{aligned} \quad (2.35)$$

ve bu sistemde, (2.35) eşitlikleriyle çıkıntı ve girinti tanımı şöyle yapılabilir:

$$L_{uv} = 0, L_{uu}^2 - L_{vv}^2 \geq 0$$

L_{uu} değerinin işareti kutup değerini verecek şekilde, $L_{uu} \leq 0$ için çıkıntı, $L_{uu} \geq 0$ için girinti tanımı yapılır [9].

2.1.4.2 Değişken Ölçekli Çıkıntıların Hesaplanması

Daha önce bahsedildiği gibi, tek ölçekte çalışmak ölçek seçme seviyesinde çok hassas olabilir. Yapılan deneylerde, çıkıntı tespitleyicinin görüntüden bağlantılı bir eğri üretebilmesi için kullanılacak Gauss çekirdeğinin ölçek parametresinin çıkıntı tespitleyici yapısının genişliğine iyi ayarlanması gerektiği görülmüştür [40]. Ön bilgiler olmadan bu sorunu çözebilmek için, ölçek parametresini çıkıntı tanımının esas bir özelliği olarak kabul eden ölçek-uzay çıkıntıları kavramı ortaya çıkmıştır. Ölçek-uzay çıkıntısı kavramı ölçek parametresinin görüntüdeki çıkıntı yapısının genişliğine kendiliğinden ayarlamasını sağlamıştır. Literatürde bunun için değişik yaklaşımlar sunulmuştur [37] [38].

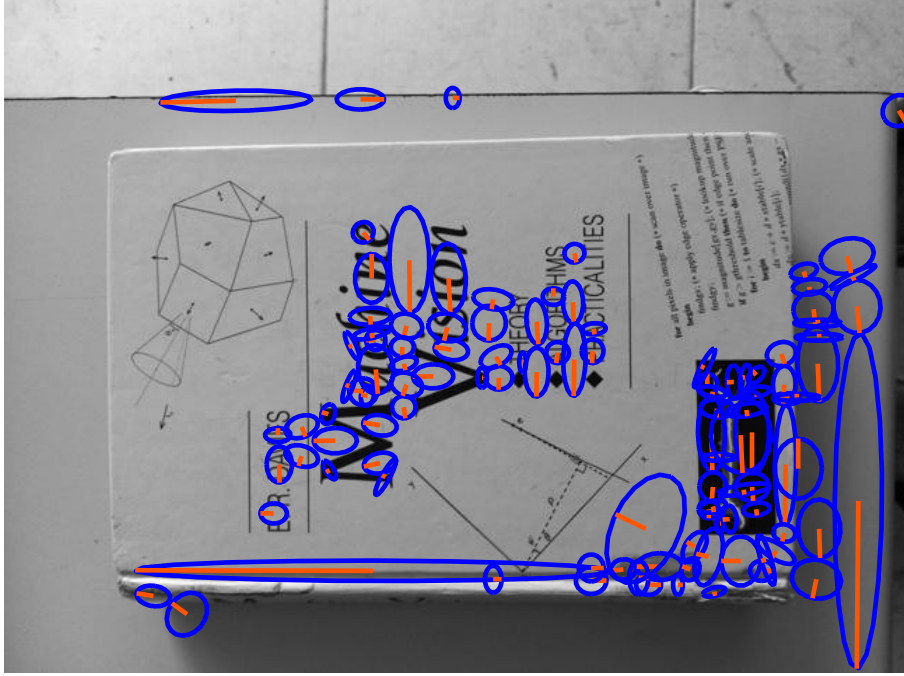
$R(c,r,t)$ çıkıntı kuvvetinin bir ölçütü olmak üzere, iki boyutlu bir görüntü için, t ölçek parametresinde ölçek-uzay çıkıntısı (2.36) eşitliğini sağlayan noktalar kümesidir. Benzer şekilde ölçek-uzay girintileri de (2.37) eşitliğini sağlayan noktalar kümesidir.

$$L_p = 0, L_{pp} \leq 0, \partial_t(R) = 0, \partial_{tt}(R) \leq 0 \quad (2.36)$$

$$L_q = 0, L_{qq} \geq 0, \partial_t(R) = 0, \partial_{tt}(R) \leq 0 \quad (2.37)$$

Bu tanımların bir sonucu olarak, iki boyutlu bir görüntü için ölçek-uzay çıkıntı kavramı tek boyutlu eğriler kümesini, ölçek parametresi ölçek-uzay çıkıntısı boyunca

değişime izin verildiği miktarda, üç boyutlu ölçek-uzayına taşır. Görüntüdeki çıkıntı tespitleyici sonrasında üç-boyutlu eğrinin iki boyutlu görüntü düzlemine yansımaları oluşturur.



Şekil 2.12. Çıkıntı Tespit Edilmiş Görüntü

Literatürde değişik çıkıntı kuvveti ölçütleri çıkarılmıştır (2.38)(2.39)(2.40). γ -normal türev terimleriyle ifade edilmiş esas eğrilik denklemi:

$$L_{pp,\gamma-norm} = \frac{t^\gamma}{2}(L_{cc} + L_{rr} - \sqrt{(L_{cc} - L_{rr})^2 + 4L_{cr}^2}) \quad (2.38)$$

$$\partial_\eta = t^{\frac{\gamma}{2}} \partial_c, \partial_\zeta = t^{frac{\gamma}{2}} \partial_r$$

γ -normal özdeğer karelerinin farklarının karesi:

$$N_{\gamma-norm} = (L_{pp,\gamma-norm}^2 - L_{qq,\gamma-norm}^2) = t^{4\gamma}(L_{cc} + L_{rr})^2((L_{cc} + L_{rr})^2 + 4L_{cr}^2) \quad (2.39)$$

γ -normal özdeğer farklarının karesi:

$$A_{\gamma-norm} = (L_{pp,\gamma-norm} - L_{qq,\gamma-norm})^2 = t^{2\gamma}((L_{cc} + L_{rr})^2 + 4L_{cr}^2) \quad (2.40)$$

Bu denklemlerde, γ -normal türevler, çıkıntı ve girinti tespitleyici algoritmaların düzgün ayarlanması açısından esastır. İki boyuta gömülmüş tek boyutlu bir Gauss çıkıntısı için tespitleme ölçeğinin, çıkıntı yapısının genişliğine eşit olması gerekmektedir, bunun için de $\gamma = \frac{3}{4}$ seçilmesi uygundur.

Yukarıdaki üç çıkıntı kuvveti ölçütünden, birincisi(2.38) damar veya haritadan yol çıkarma uygulamalarında kullanılan genel amaçlı çıkıntı kuvveti ölçütüdür. Üçüncüsü (2.40) ise parmak izi büyütmede, gerçek zamanlı el takibinde, ifade tanımada kullanılmakla birlikte, bölgesel görüntü istatistiklerini modelleme ve videolarda insan takibi için kullanılmaktadır.

2.1.5 Hough Dönüşümü ile Şekil Tespiti

Hough dönüşümü görüntü işleme ve analizinde bir özellik çıkarım tekniğidir. Bu tekniğin amacı görüntü içerisindeki nesnelere eksikli kopyalarını oylama yöntemiyle belirli bir şekil sınıfı içerisinde bulmaktır. Oylama prosedürü, nesne adaylıklarının algoritma tarafından oluşturulan, akümülatör uzayında bölgesel en büyük olarak adlandırılmış nesne adaylarının bulunduğu bir değiştirge uzayında gerçekleştirilir.

Klasik Hough dönüşümü bir görüntüdeki düz çizgilerin tanımlanmasıyla ilgilenirken [41], Hough dönüşümü kullanımı önce dairelerin konum tespitini [42], daha sonra rastgele şekillerin konumlarını tanımlamak [43] için genişletilmiştir.

2.1.5.1 Teori ve Uygulama

Bir çok durumda nesnenin sınır noktalarını ve birleşimlerini elde etmek için ön işlem olarak bir kenar tespitleyici kullanılabilir. Görüntüdeki veya kenar tespitleyicideki hatalardan, istenilen sınır çizgilerinde piksel eksiklikleri ve gürültüler oluşabilir. Bu nedenlerden, çıkarılmış kenar özelliklerini belirli bir çizgi veya daire kümesine toparlamak kolay olmayabilir. Hough dönüşümünün amacı parametrize edilmiş nesne kümeleri üzerinde oylama prosedürü uygulayarak, kenar piksellerinin kümelenmelerini nesne adaylığına dönüştürmeye imkan vermektir.

2.1.5.2 Çizgi Tespiti

Düz çizgileri tespit etmedeki ana kavram nokta-çizgi ikiliğidir. Bir nokta koordinatlarıyla tanımlanabildiği gibi üzerinden geçen çizgiler kümesiyle de tanımlanabilir. Bir nokta çizgiler kümesiyle tanımlanabildiği gibi, bir çizgi de noktalar kümesi olarak tanımlanabilir. Hough dönüşümünün en basit hali düz çizgi tespit eden lineer dönüşümdür [41] [42].

Yöntemin asıl uygulanma şekli çizgilerin eğim-kesme eşitliğinin (2.41) parametrize

edilmesini içerir.

$$y = mx + b \quad (2.41)$$

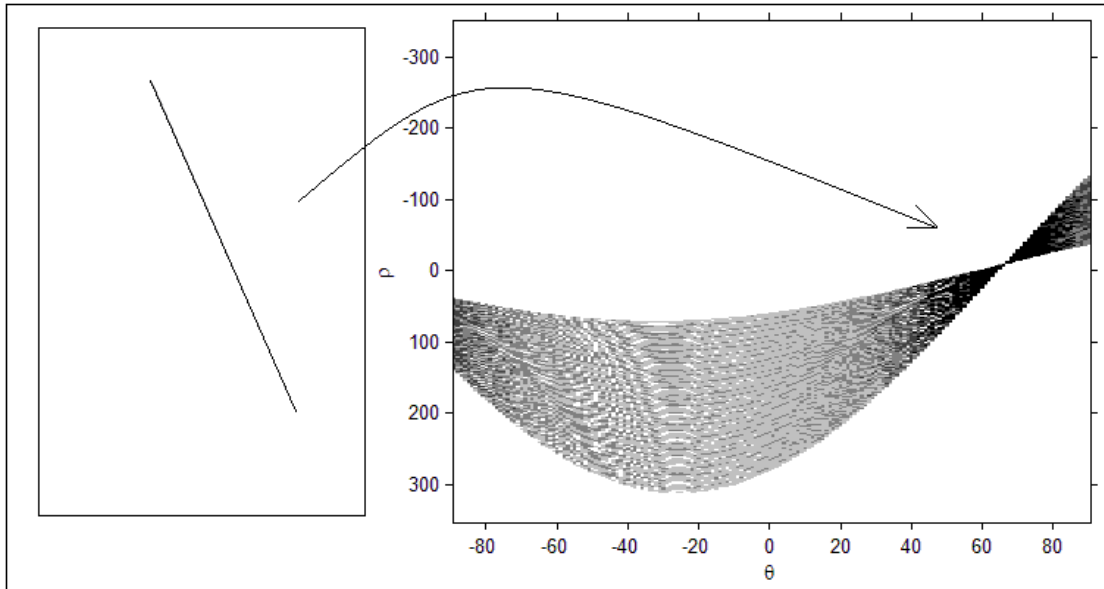
Böylece düz çizgiyi oluşturan noktalar (m, b) uzayında bir çizgiyi temsil edip, bu çizgilerin kesişimi bu uzayda tespit edilmiş çizgiyi gösterir. (m, b) değerlerinin sınırsız aralıkta olması sıkıntısı biri 1'den küçük eğimleri, diğeri 1 ve 1'den büyük değerleri kapsayan iki küme kullanılarak çözülmüştür ve eşitlik (2.42)'e dönüştürülmüştür.

$$x = \tilde{m}y + \tilde{b}, \quad \tilde{m} = \frac{1}{m} \quad (2.42)$$

Sınırsız aralıkta değerlerin olması başka bir yaklaşımla geliştirilmiştir; düz çizgi eğim-kesme ifadesi normal (θ, ρ) (2.43) biçimine dönüştürülmüştür.

$$\rho = c \cos \theta + r \sin \theta \quad (2.43)$$

Bu yöntemi uygularken, çizginin her noktası (θ, ρ) uzayında bir sinüs eğrisi olarak ifade edilir. (θ, ρ) uzayında bir noktadaki en fazla kesişim noktası (c, r) görüntü uzayında merkeze göre θ açısında, ρ uzaklığında düz çizgi olduğunu gösterir (Şekil 2.13).



Şekil 2.13. Hough Dönüşümüyle Çizgi Tespiti

Hough dönüşüm algoritmasında akümülatör denilen bir dizi kullanılır. Akümülatörün boyutları problemdeki bilinmeyen parametrelerin sayısına eşit olacak şekilde düzen-

lenir. İki boyutlu bir görüntüde çizgi tespiti için her iki yöntem için de bilinmeyen parametrelerin miktarı (θ, ρ) veya (m, b) olmak üzere ikidir. Kullandığımız ikinci yöntemde, akümülatör dizisinin θ ve ρ boyutları nicelendirilmiş değerlere tekabül eder. Her bir piksel için, o pikselin ve komşusunun görüntüde bir kenar olup olmadığına bakılır ve eğer bir kenar pikseliyse, o piksel için (θ, ρ) parametreleri hesaplanıp, akümülatörde o koordinatlara denk gelen dizi değeri arttırılır. Daha sonra oluşturulan dizideki bölgesel en büyük değerler ρ uzaklığında ve θ açısında düz çizgiler olduğunu gösterir.

2.1.5.3 Daire Tespiti

Yuvarlak nesnelerin yer tespiti özellikle kendiliğinden denetim ve birleştirme gibi endüstriyel uygulamalarda çok önemlidir. Üretilen bir çok parçanın denetimi, birleştirilmesi, ölçülmesi yüksek hassasiyetle yapılması gerekmektedir. Hough dönüşümü tekniği bu yuvarlak nesnelerin yerlerini tespit etmek için uygun bir yöntemdir ve daire tespiti için Hough dönüşümünün uygulanması, bu tekniğin en basit halidir [42].

Daire tespiti için yapılan ilk Hough dönüşümünde, öncelikle görüntüdeki bütün piksellerdeki gradyan büyüklükleri (2.44) ve açıları (2.45) kestirilip, kenar tespiti yapmak için eşik değerinden geçirilir. Sonra, merkezin yerini bulabilmek için daha önce bulunan bütün kenar noktalarının öngörülen yarıçap R uzaklığındaki noktaları (2.46) boş bir parametre uzayının belirli bölgelerinde toplanır. Burada toplanan değerlerin bölgesel en büyüğü bize dairenin yaklaşık merkez noktasını verir. Son olarak, kenar tespitinde çıkan sonuçlarda kenar kalınlıkları sıfır olmadığından ve oluşan gürültü daire merkezinin yerini etkileyeceğinden en büyüklerin bulunduğu bölgedeki değerlerin uygun bir ortalaması alınarak daha kesin bir merkez noktası bulunur (2.46).

$$g = (g_c^2 + g_r^2)^{\frac{1}{2}} \quad (2.44)$$

$$\theta = \arctan\left(\frac{g_r}{g_c}\right) \quad (2.45)$$

$$\begin{aligned} c_{merkez} = c - R \sin \theta &\implies c_{merkez} = c - R \left(\frac{g_r}{g}\right) \\ r_{merkez} = c - R \cos \theta &\implies r_{merkez} = c - R \left(\frac{g_c}{g}\right) \end{aligned} \quad (2.46)$$

Kenar tespitleyici algoritmasında kullanılan Sobel maskeleri (2.5) kullanılarak g_c ve g_r gradyan değerleri bulunabilir.

2.1.5.4 Genel Hough Dönüşümü (GHT)

Genel Hough dönüşümü yönteminde rastgele bir şeklin sınır çizgileri bir haritalandırılmayla başka bir uzaya aktarılmaktadır. Bu teknikte daha önceki Hough dönüşümlerinden farklı olarak görüntü içerisindeki nesnenin geometrik bir şekle sahip olması gerekmemektedir. Genel Hough dönüşümü için bir tablo oluşturulmalıdır. Bu tablodaki verilerden oylama yapılarak nesnenin şekli ve yeri tespit edilebilir [43].

Öncelikle görüntüdeki şeklin sınırları içerisinde kalacak bir M noktası seçilir. Bir A sınır noktasından M noktasına çizilecek $\vec{R} = \vec{A} - \vec{M}$ vektörünün uzunluğu (R) ve apsisle yaptığı açı (φ) sınır noktasının gradyan açısı değerinin (β) belirttiği tabloda $T[\vec{R}]$, R – Tablosu β 'nin fonksiyonları olarak $T[R(\beta), \varphi(\beta)]$ saklanır.

Bir şeklin yönelimi ve ölçeği daha önce tablo oluşturulan şekillerden farklıysa, tabloya iki yeni parametre daha eklenir (s), (θ). Böylece iki boyutlu tablo dört boyutlu bir tabloya dönüşür.

$$\begin{aligned} T_s[R(\beta)] &= sR(\beta) \\ T_\theta[R(\beta)] &= \text{rot}R((\beta - \theta) \bmod 2\pi), \theta \end{aligned} \quad (2.47)$$

Şeklin s ile ölçeklendirilmesi veya θ ile döndürülmesi sonucu (R – Tablosu) eşitlik (2.47)'deki halini alır.

Daha sonra boş bir parametre uzayı oluşturulup, oluşturulan R – tablosu kullanılarak şeklin her bir sınır noktasının gradyan açısına denk gelen ilgili $\vec{R}(\beta)$ değeri vektörel olarak eklenerek (2.48) bulunan noktadaki değer bir arttırılır.

$$\vec{M}' = \vec{A}' + \vec{R} \quad (2.48)$$

Sonrasında oluşturulan bu parametre uzayı başka parametre uzaylarıyla karşılaştırılarak nesnenin yeri ve şekli tespit edilmiş olur.

2.2 Özellik Tabanlı Tanımlayıcı Oluşturma Yöntemleri

Özellik tanımlayıcılar, bir görüntüdeki tespit edilen özelliklerin hemen hemen benzersiz tanımlayıcılarıdır. Genellikle özellik tanımlayıcılar bir görüntüde tespit edilen

ilgi noktalarında veya onların komşu bölgelerinde işlem yaparak o noktaya bir tanımlama yapmak için geliştirilmiştir.

2.2.1 Ölçek Değişimsiz Özellik Dönüşümü (SIFT)

SIFT, görüntü içerisinde tespit edilen anahtar noktalara özellik tanımlayıcı vektör oluşturulan bir yöntemdir. Bu özellikleri çıkarmak için görüntü bir dizi süzgeçten geçirilir, fark işlemleri yapılır, bölgesel enuç noktalar tespit edilir, bu noktalardan bazıları elenir ve kalan noktalar anahtar noktalar olarak tanımlanır. Bu noktaların özellikleri bu aşamadan sonra belli olur ve bölgesel işlemler için anahtar noktanın etrafında bir bölge seçilir, bu bölgedeki piksellerin gradyanı bulunur ve gradyanların ağırlıklı histogramı anahtar noktanın gradyan yönelimini verir. Bu yönelimle aynı bölge içerisinde her pikselin ayrı ayrı gradyan yönelimleri tanımlayıcıyı oluşturur [11] [22] [44] [45].

2.2.1.1 Anahtar Noktaların Bulunması

Gaussların Farkı: Görüntüye ölçek değişimsizliği kazandırmak için öncelikle Gaussların farkı süzgeci uygulanır (2.49).

$$D(c, r, \sigma) = (G(c, r, k\sigma) - G(c, r, \sigma)) * I(c, r) \quad (2.49)$$

Bölgesel Enuç Tespiti: Gaussların farkı uygulanmış görüntülerde her piksel komşu pikselleriyle karşılaştırılarak o piksellere göre en büyük veya en küçük olup olmadığına bakılır, eğer en büyük veya en küçükse işaretlenir ve aday anahtar nokta olarak kabul edilir.

Kesin Anahtar Noktaların Tespiti: Enuç piksellerden bir çoğu gerçekte tam sayıların belirlediği piksel noktalarında olmayıp, rasyonel değerlerdedir. Gerçek enuç olmayan değerler elenir. Ayrıca kalan piksellerden bazıları kenar tepkisi olduğundan veya kontrast eşliğinin altında olduğundan elenir. Kalan pikseller kesin anahtar noktaları olarak kabul edilir.

2.2.1.2 Anahtar Noktalara Yönelim Atama

Tespit edilen kesin anahtar noktalarının etrafında 16x16lık piksel alanı alınıp bu noktaların gradyanları bulunur. Bu gradyanların büyüklükleri 36adet 10°lik açı kümesi aralığında toplanır, ve bu toplamlardan bir histogram çıkarılır. Bu histogramda en

büyük küme ve en büyük küme değerinin 0.8 katından büyük kümelerden en büyük iki tanesi bu anahtar noktanın gradyan yönelimi olarak belirlenir.

2.2.1.3 Tanımlayıcı Oluşturma

Gradyan yönelimi bulunan noktanın etrafında yine 16x16lık alan 4x4lük alanlarla 16 parçaya ayrılarak, bu alanların gradyanları 8 adet 45°lik yönelim kümesine ayrılarak toplanır. Bu kümelerin değerleri $4 \times 4 \times 8 = 128$ elemanlı bir vektör oluşturur. Bu oluşan vektör anahtar noktasının tanımlayıcısı olarak belirlenir.

2.2.2 Hızlandırılmış Gürbüz Özellikler (SURF)

SIFT yönteminden etkilenerek anahtar noktası tespit etmeyle ve bu anahtar noktalarına özellik vektörü çıkarmakla gerçekleştirilen bir tekniktir [14].

Anahtar noktaları tespit etmek için damlacık tespitlemeye kullanılan Hessian determinantı yönteminin gelişmiş bir türü olan Hızlı-Hessian tespitleyici kullanılmıştır. Bu tespitleyici hesaplama süresini iyice düşüren tümlev görüntüleri kullanır. Tümlev görüntü, $I_{\Sigma}(\mathbf{x})$, her piksel koordinatında hesaplanır ve yeni bir görüntü elde edilir (2.50).

$$\mathbf{x} = (c, r)^T$$

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq c} \sum_{j=0}^{j \leq r} I(i, j) \quad (2.50)$$

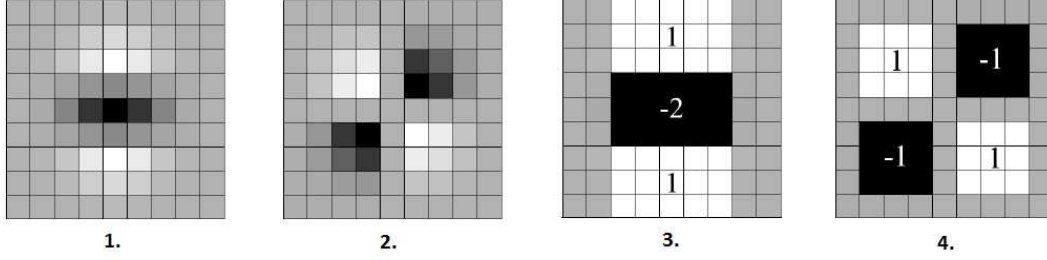
Tümlev görüntü elde edildikten sonra dikdörtgen bir alanda piksel büyüklüklerinin toplamlarının hesaplanması sadece üç toplama işlemi ile hesaplanabilir.

Hessian matrisinde determinantın en büyük olduğu noktalar damlacık tespitine benzer sonuçlar vermektedir, ayrıca ölçek seçimi de Hessian determinantı hesabına dayanmaktadır. $L_{cc}(\mathbf{x}, \sigma)$ verilen bir $\mathbf{x} = (c, r)^T$ noktasında, σ ölçeğinde Gauss'un c 'ye göre ikinci türevi $\frac{\partial^2}{\partial c^2} g(\sigma)$ ile evrişimi olmak üzere, Hessian matrisi eşitlik (2.51)'de tanımlanmıştır. $L_{rr}(\mathbf{x}, \sigma)$ ve $L_{cr}(\mathbf{x}, \sigma)$ de aynı şekilde sırasıyla r 'ye göre ikinci ve c ve r 'ye göre ikinci türevleriyle görüntünün evrişimini tanımlamaktadır.

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{cc}(\mathbf{x}, \sigma) & L_{cr}(\mathbf{x}, \sigma) \\ L_{cr}(\mathbf{x}, \sigma) & L_{rr}(\mathbf{x}, \sigma) \end{bmatrix} \quad (2.51)$$

Gauss süzgeçleri ölçek-uzay analizi için en uygundur, fakat bu yöntemin uygulan-

ması için bu süzgeçlerin pratikte ayrıklaştırılması ve kırılması gerekmektedir. Gerçek süzgeçlerin hiç bir durumda ideal olamaması ve Gaussların farkı yaklaşımı başarılı sonuç vermesinden dolayı, Hessian matrisi için bu yaklaşım kutu süzgeçlerle daha da ileri götürülebilir (Şekil 2.14). Tümlev görüntülerde bu yaklaşım hesap yükünü düşük tutmaktadır.



Şekil 2.14. 1. Gauss süzgecinin r yönünde ikinci kısmi türevi, 2. Gauss süzgecinin cr yönünde ikinci kısmi türevi, 3. Gauss süzgecinin r yönünde ikinci kısmi türevine SURF yaklaşımı, 4. Gauss süzgecinin cr yönünde ikinci kısmi türevine SURF yaklaşımı. Gri bölgeler sıfır olarak belirlenmiştir.

(Şekil 2.14)'daki 9x9 luk süzgeçler $\sigma = 1.2$ 'lik Gauss yaklaşımı olup, damlacık tepkilerini çıkarmak için en alt seviyeyi temsil eder. $|\mathbf{x}|_F$ Frobenius norm olmak üzere dikdörtgen alana uygulanan ağırlıklandırılmalar hesaplamalar için basite indirgenmektedir (2.53).

$$\det(H_{approx}) = D_{cc}D_{rr} - (\omega D_{cr})^2 \quad (2.52)$$

$$\omega = \frac{|L_{cr}(1.2)|_F |D_{rr}(1.2)|_F}{|L_{rr}(1.2)|_F |D_{cr}(1.2)|_F} = 0.912.. \approx 0.9 \quad (2.53)$$

D_{cc} , D_{rr} , D_{cr} , ayrıklaştırılmış Gauss süzgeci yaklaşımıyla görüntünün evriştirilmiş halleri olmak üzere, süzgecin göreceli ağırlık katsayısı ω yaklaşık Hessian determinanında (2.52) dengeyi bulmak için kullanılır.

Bütün ölçeklerde ilgi noktası tespit etmek için değişik boyutlarda ve ölçeklerde kutu süzgeçleri aynı anda tümlev görüntülere uygulayarak ölçek-uzay yapısı çıkarılabilir. Bu ölçek-uzay yapısında Hessian determinantlarının enuç tespiti yapılarak ilgi noktaları çıkartılır.

2.2.2.1 İlgi Noktası Tanımlayıcıları

SURF yönteminde de SIFT yönteminde kullanılan gradyan yerine gradyan bilgisine yakın c ve r yönlerinde Haar dalgacıklarının tepkileri kullanılır. Özellik hesaplamada yükü azaltmak için bu kullanılmaktadır. Ayrıca Laplace işlecinin işareti eşleme hızını da arttırmaktadır. Tanımlayıcıyı hesaplamak için iki temel aşama vardır. Bunlar sırasıyla yönelim atama ve SURF tanımlayıcı çıkarımıdır.

Yönelim Atama: Görüntünün döndürülmesine değişimsiz olmak için, ilgi noktalarına bir yönelim atanmaktadır. Bu yönelimi atamak için öncelikle ilgi noktasının etrafında, s bulunan ilgi noktasının ölçeği olmak üzere, yarıçapı $6s$ 'lik dairesel bir bölgede c ve r yönlerinde Haar dalgacıklarının tepkileri hesaplanır. Sonrasında hesaplanmış ve $\sigma = 2$ Gauss ağırlıklandırılmış dalgacık tepkilerinin, $\frac{\pi}{3}$ 'lük yönelim pencereleri içerisinde en baskın olanı kestirilir ve bu pencere içerisinde boyu en uzun tepkinin yönelim açısı ilgi noktasının yönelimini belirler.

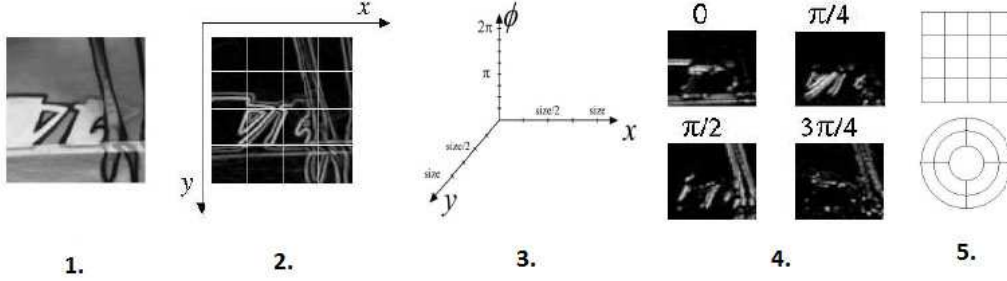
Haar Dalgacık Tepkilerinin Toplamıyla Tanımlayıcı Oluşturma: İlk olarak ilgi noktası etrafında yönelime göre bir kenarı $20s$ uzunluğunda kare bir bölge oluşturulur. Bu bölge $4 \times 4 = 16$ alt bölgeye bölünür. Daha sonra her bir alt bölge için 5×5 lik düzenli aralıklandırılmış örnek noktalarının c ve r yönlerinde Haar dalgacık tepkileri hesaplanır. Bölgelerin tam hesaplanamaması ve geometrik bozulmalara karşı koruma amaçlı büyük bölge ($\sigma = 3.3s$) Gauss penceresiyle ağırlıklandırılır. Sonra dalgacık tepkileri (d_c ve d_r) her bir alt bölgede toplanır ve tanımlayıcı vektörünün ilk girdilerini oluşturur. Kutuplar hakkında bilgi verebilmek için dalgacık tepkilerinin mutlak değerleri de vektörün iki elemanı olarak kaydedilir. Bir alt bölge 4 boyutlu $\mathbf{v} = (\sum d_c, \sum d_r, \sum |d_c|, \sum |d_r|)$ vektörü tanımlanmak üzere, $4 \times 4 = 16$ bölgenin vektörlerinin biraraya getirilmesiyle 64 boyutlu tanımlayıcı vektör oluşturulur.

2.2.3 Gradyan Konumu ve Yönelimi Histogramı (GLOH)

Anahtar noktaların tespit edilmesi aşamasında SIFT tekniğindeki tespitleme yönteminin yanında bu yönleme yakın sonuçlar veren Hessian-Laplace işleci kullanılır. Hessian-Laplace işleci SIFT'de kullanılan DoG yönteminden farklı olarak kenar tepkilerini fazla yansıtmaz ancak yakın sonuç vermektedir [12].

İlgi Noktalarında Tanımlayıcı Oluşturma: Esas olarak bu yöntem SIFT yöntemi-

nin geliştirilmiş bir biçimidir ve bir çok açıdan benzerlikler taşımaktadır. Tespitleyici aşamasında olduğu gibi, tanımlayıcı oluşturmada da bu yöntem SIFT yönteminde kullanılan gradyan yönelimi tabanını kullanmaktadır.



Şekil 2.15. SIFT Tanımlayıcı 1. Özelliği tespiti yapılmış bölge, 2. Gradyan görüntüsünde ızgaralar, 3. Histogram boyutları, 4. 8 yönelim düzleminde 4'ü, 5. Kartezyan ve log-polar konumlandırma ızgaraları. Görüldüğü gibi log-polar ızgaraların gösterdiği gibi bölge 4 açısal olmak üzere toplamda 9 bölgeye bölünmüştür.

SIFT tanımlayıcıda hesaplanan değerleri bölgesel ızgara yapısını (Şekil 2.15) yarıçapları 6, 11 ve 15, ve merkez hariç diğer bölgeleri 8 açısal alt bölgeye bölecek şekilde log-polar ızgara yapısına çevirmeye bir tanımlayıcı vektörü oluşturulur. Vektör 17 boyutlu olup, gradyan yönelimleri 16 kümeye bölündüğünden toplamda 272 boyutlu bir histogram elde edilir. Bu da eşleme performansı için Esas Bileşen Analizi yöntemiyle 128 boyuta indirilir.

2.2.4 Yönlü Gradyanların Histogramı (HOG)

Bu teknik temel olarak bir görüntünün içerisinde bir nesnenin veya şeklin varlığını kenar yönlerinin veya yoğunluk gradyanlarının dağılımından tanımlayıcılar oluşturur [13]. Bu tanımlayıcılar, görüntüyü bağlantılı küçük hücrelere bölerek bu hücrelerin gradyan yönlerinin histogramı veya hücrelerin içindeki kenar yönlerinin bulunmasıyla oluşturulur. Bu histogramların bir araya gelmesiyle tanımlayıcı vektörleri elde edilir [13].

Gradyan Hesaplama: Bir çok diğer yöntemlere göre bu yöntemde görüntünün ön işlenmesi, başarımda çok büyük katkıda bulunmadığı için yapılmaz. İlk olarak görüntünün tamamının gradyanı çıkarılır. Bunun için temel maskelerden faydalanılır (2.1) (2.3) (2.5).

Yönelim Kümeleme: İkinci olarak hücre histogramları çıkarılır. Hücrenin içindeki her bir piksel daha önce bulunmuş gradyan değerlerinden yönelim histogramı çıkarmak için bir ağırlık ekler. Hücrelerin şekli dikdörtgensel veya dairesel olabilir, histogram kanalları çift olarak işaretli veya işaretsiz olmasına göre 0° ile 180° arasında veya 0° ile 360° arasında dağılmıştır. Oylama ağırlığı için, pikselin katkısı gradyan büyüklüğünün kendisi veya bir fonksiyonu olabilir. Başka seçenekler olarak, gradyan büyüklüğünün karesi veya karekökü de kullanılabilir.

Tanımlayıcı Blokları: Kontrast ve ışıklandırma değişimlerine karşı, hücreleri düzenli olarak bir araya getirip birbirine bağlanmış bloklar oluşturmak için gradyan büyüklükleri bölgesel olarak normalize edilmelidir. İki ana hücre bloğu yapısı vardır, R-HOG dikdörtgensel bloklar, C-HOG dairesel bloklar olmak üzere,

R-HOG, genel olarak kare ızgaralarla bölümlendirilir ve üç parametreyle tanımlanır; blok başına hücre sayısı, hücre başına piksel sayısı ve hücre histogramı başına kanal sayısı. R-HOG blokları SIFT tanımlayıcılara çok benzemektedir, ancak R-HOG blokları tek bir ölçekte yoğun ızgaralarla ve yönelim atanmadan hesaplanır.

C-HOG blokları iki değişken şeklinde bulunabilir; tek bir merkez hücresi olanlar ve açısal bölümlendirilmiş merkez hücresi olanlar. Ayrıca C-HOG blokları dört parametreyle tanımlanabilir: açısal kümeler, yarıçapsal kümeler, merkez kümenin yarıçapı, ve eklenen yarıçapsal kümelerin yarıçapının büyüme çarpanı.

Bloklar normalize edilmelidir ve bu normalizasyon için dört farklı yöntem geliştirilmiştir. ν bir bloktaki bütün histogramları kapsayan normalize edilmemiş bir vektör ve $\|\nu\|_k$ k'inci norm ve ε küçük bir sabit olmak üzere olmak üzere, normalizasyon çarpanları şu şekilde tanımlanmıştır:

$$L1 - norm : f = \frac{\nu}{(\|\nu\|_1 + \varepsilon)}$$

$$L1 - sqrt : f = \sqrt{\frac{\nu}{(\|\nu\|_1 + \varepsilon)}}$$

$$L2 - norm : f = \frac{\nu}{\sqrt{\|\nu\|_2^2 + \varepsilon^2}}$$

L2-hys: L2-norm'un en büyük değerinin 0.2'de kesilerek yeniden normalize edilmesiyle oluşturulur.

İlgili çalışmada, dört normalizasyonda da normalize edilmemiş verilere göre tanımlayıcıların eşleştirmesinde iyileştirmeler sağlandığı belirtilmiştir [13]. Tanımlayıcılar bütün bölgelerden toplanan normalize edilmiş hücre histogramlarının parçalarının oluşturduğu vektörlerin birleşiminden oluşturulur.

3. NESNE TANIMA İÇİN ÖZELLİK ÇIKARIMI VE TANIMLAYICI OLUŞTURMA

3.1 Ölçek Değişimsiz Özellik Dönüşümü (SIFT) Yöntemi ve Uygulamaları

SIFT yöntemi robotik haritalama ve yönlendirme, görüntü birleştirme, 3B modelleme, video takibi, yüz ve mimik tanıma, nesne ve şekil tanıma gibi bir çok uygulamada kullanılmaktadır. Bunun için SIFT yönteminde görüntünün Gauss süzgeci ile yumuşatılmış halleri kullanılarak belirli ilgi noktaları çıkartılır ve bu noktalar için tanımlayıcı vektörler oluşturulur [37] [39].

Tüm tanıma yöntemlerinde olduğu gibi bu yöntem için de ilk aşama, anahtar noktalarını çıkarmaktır. Daha sonra çıkarılan ilgi noktalarının komşu çevresindeki noktalardan yönelim çıkarılıp ilgi noktasına atanmaktadır. En son olarak her bir ilgi noktasının atanmış her yönelimi için bir tanımlayıcı oluşturulmaktadır [11] [22] [44] [45].

3.1.1 Kullanılan Donanım ve Yazılım Araçları

Bütün deneysel sonuçları almak için *Intel*[®] *Core2 Q9550* modeli 2.8GHz hızında bir işlemci, 4GB 800MHz DDR2RAM, 64-bit Windows7 Professional işletim sistemi ile kullanılmıştır.

Bir çok bilimsel, endüstriyel araştırma için en çok tercih edilen benzetim ve uygulama platformlarından biri olan MATLAB kullanılmıştır. MATLAB içerisinde bir çok araç bulunmakta, bu araçlar yapılacak işler için özelleştirilmiş fonksiyon ve örnek program barındırmaktadır. Görüntü işleme ve gerekli benzetimleri yapmak MATLAB'da "Image Processing Toolbox" ve "Parallel Processing Toolbox" ve bunlara yardımcı bir çok araç ve fonksiyon kullanılmıştır. Ayrıca gereken diğer bir çok fonksiyon ve program da çalışma içerisinde oluşturulmuştur.

3.1.2 Ön İşlemler

Herhangi bir özellik tespitleme veya özellik çıkarımı yöntemini bir görüntüye uygulamak için genellikle görüntü gri ölçekte olmalıdır. En yaygın kullanılan sayısal görüntü formatlarından biri "jpg" formatıdır, bu formatta da genel olarak sayısal "RGB" renk formatı kullanılmaktadır. Eğer görüntü "RGB" renk formatındaysa ve işlemler gri ölçekte uygulanması gerekiyorsa, bu görüntü gri ölçeğe çevrilir (3.1) ve 8 bit piksel

tanımlamada kullanılan "0-255" kontrast değeri aralığı tüm görüntü işleme algoritmalarında standard olarak kullanılan "0-1" aralığına taşınır.

$$GR = 0.299R + 0.587G + 0.114B \quad (3.1)$$

3.2 Anahtar Noktaları Bulma

İlk olarak anahtar noktası adayları tespit edilir, sonra bu aday anahtar noktalarının bir kısmı ileride oluşturabileceği çeşitli olumsuzluklardan dolayı elenir ve elde kalanlar kesin anahtar noktalarıdır.

Aday anahtar noktaları bulmak için görüntü bir takım işlemlerden geçirilmektedir. Bu işlemler sırasıyla Gauss süzgeçleme, süzgeçlenen görüntülerin farklarının alınması, bu fark görüntülerinde enuç nokta tespitidir.

Anahtar noktaları kesinleştirmek için, tespit edilen aday anahtar noktalarının kesin koordinatları bulunur, bu işlem adayların bir kısmını elemektedir, sonra kalan adaylar bir eşik değerinden geçirilerek kenar tepkileri ve kontrastı düşük olan noktalar elenmektedir. Elenmeyen noktalar kesin anahtar noktaları olarak işaretlenmektedir.

3.2.1 Gaussların Farkları Piramidinin Oluşturulması

Aday anahtar noktalarını çıkarmak ve ölçek değişimsizliğini sağlayabilmek için ardarda Gauss süzgeçleme yöntemi kullanılmıştır. Aday anahtar noktaları çıkarmak için aynı görüntünün değişik açılardan ve boyutlarda tanımlanabilmesi gerekmektedir. Ölçeklendirmeye değişimsiz olan görüntü bölgelerini çıkarabilmek, bütün ölçek seviyelerinde kararlı özellik bölgelerinin taranmasıyla gerçekleştirilebilir. Çalışmalarda, bir çok çeşitli yaklaşım altında geçerli ölçek-uzay çekirdeği Gauss fonksiyonuyla oluşturulmaktadır. Bundan dolayı bir görüntünün ölçek-uzay fonksiyonu (3.2), değişken ölçekli Gauss fonksiyonuyla (3.3) evrişimiyle oluşturulur [38].

$$L(c, r, \sigma) = G(c, r, \sigma) * I(c, r) \quad (3.2)$$

$$G(c, r, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(c^2+r^2)}{2\sigma^2}} \quad (3.3)$$

Ölçek uzayında kararlı anahtar noktaların çıkarımı Gaussların farkı fonksiyonunun görüntüyle evrişimi sonucu (3.4) çıkarılabilir. Bu fonksiyonun seçilmesinin bir kaç

nedeni vardır. Elimizde daha önceden elde ettiğimiz ve daha sonra özellik tanımlamada kullanacağımız Gauss ile yumuşatılmış görüntülerimizin elimizde olması, sadece basit bir çıkarma işlemiyle bu fonksiyonu hesaplamamızı kolaylaştırmaktadır. Ayrıca yapılan çalışmalarda Gaussların farkının, ölçek-normal Gauss'un Laplace işlecine, $\sigma^2 \nabla^2 G$, çok iyi bir yaklaşımı olduğu çıkarılmıştır [11] [22].

$$D(c, r, \sigma) = (G(c, r, k\sigma) - G(c, r, \sigma)) * I(c, r) = L(c, r, k\sigma) - L(c, r, \sigma) \quad (3.4)$$

Yapılan başka bir çalışmada $\sigma^2 \nabla^2 G$ 'nin enuç tespiti, gradyan, Hessian veya Harris köşe tespiti yöntemlerinden daha kararlı görüntü özellikleri çıkardığı saptanmıştır. D ile $\sigma^2 \nabla^2 G$ arasındaki ilişki ısı yayılım denklemiyle (3.5) açıklanabilir. Buradan $\nabla^2 G$ 'nin $\frac{\partial G}{\partial \sigma}$ 'ya sonlu fark yaklaşımıyla hesaplanabileceği çıkarılabilir.

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla G \quad (3.5)$$

$$\sigma \nabla G = \frac{\partial G}{\partial \sigma} = \frac{G(c, r, k\sigma) - G(c, r, \sigma)}{k\sigma - \sigma} \quad (3.6)$$

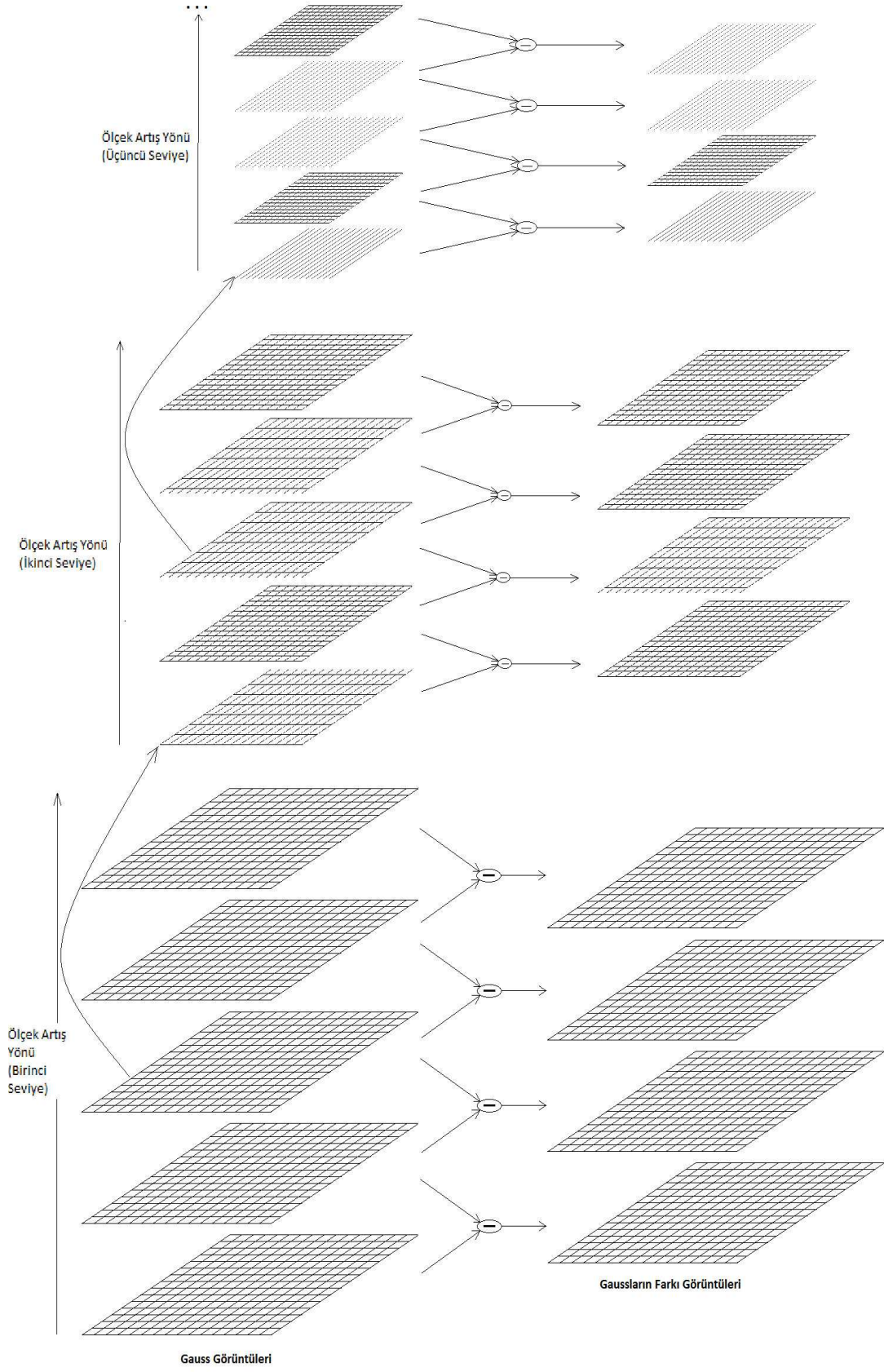
$$G(c, r, k\sigma) - G(c, r, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G$$

Gaussların farkı (DoG) fonksiyonu Gauss'un Laplace işleci fonksiyonuna göre hali hazırda σ^2 ölçeğiyle normalize edilmiştir. Enuç tespitinde, pratik anlamda Gaussların farkı yaklaşımının Gauss'un Laplace işlecine göre neredeyse hiç farkının olmadığı görülmüştür [22].

Ölçek uzayında değişimsizlik elde etmek için görüntü Gauss süzgecinden ardarda geçirilir, bununla k çarpanıyla ayrılmış görüntüler elde edilir. Her görüntü büyüklüğü seviyesi için ölçek katsayısı $k = 2^{1/s}$ olmak üzere süzgeçten geçirilmiş $s + 3$ tane görüntü oluşturulur.

Görüntüden daha fazla kararlı nokta elde edebilmek için öncelikle görüntü büyüklüğü iki katına çıkarılır ve $\frac{\sigma}{2}$ ölçeğinde örtüşme önler Gauss süzgecinden geçirilir. $s = 2$ seçilmesi yeterli görüldüğünden [22], 5 adet ardarda süzgeçlenmiş görüntü elde edilir. Sonra en üst ölçeğin iki altındaki görüntünün büyüklüğü yarıya indirilip işlemler tekrarlanır, bir seviye daha oluşturulur ve toplamda her seviyede 5'er adet görüntü olmak üzere 4 seviyeli bir görüntü piramidi oluşturulur (Şekil 3.1).

Oluşturulan görüntülerin farkları alınarak her seviyede 4'er adet Gauss farkı(DoG) görüntüsü elde edilir.



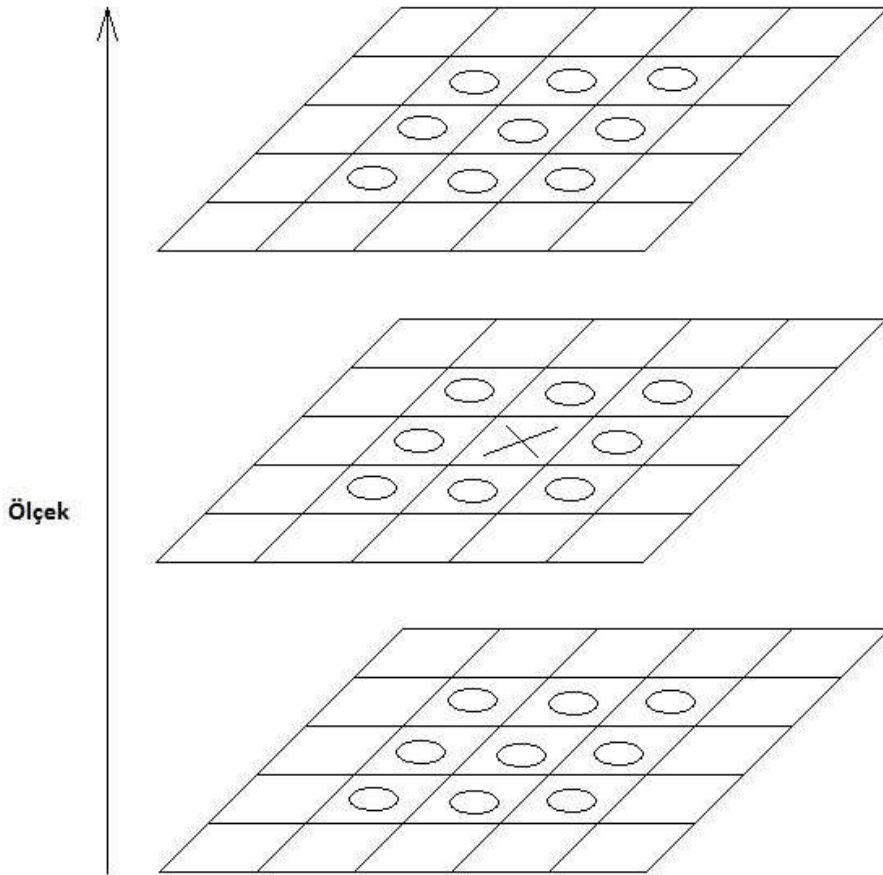
Şekil 3.1. Gauss ve Gaussların Farkı Piramitleri



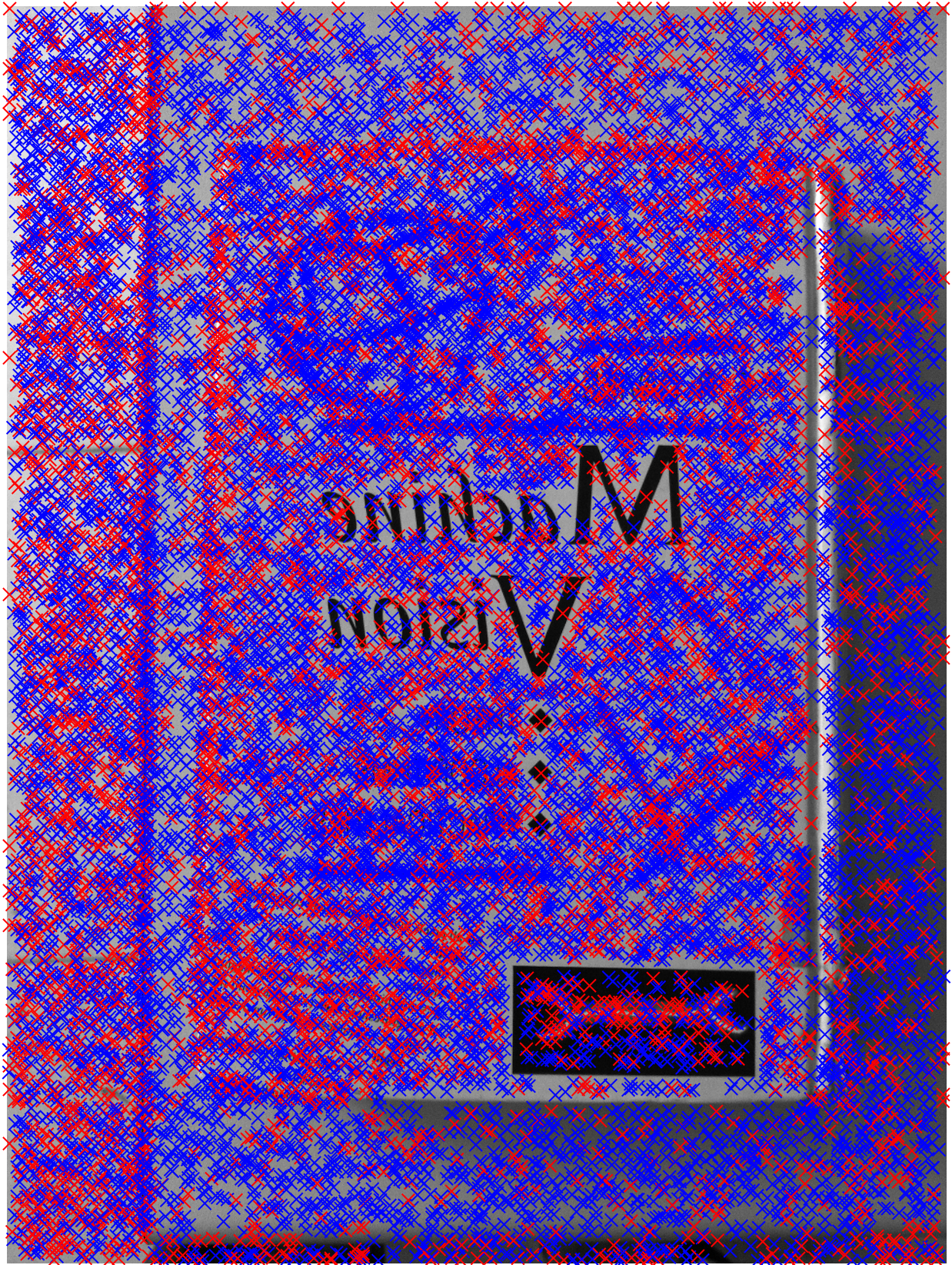
Şekil 3.2. Gauss Piramiti Görüntüleri

3.2.2 Gauss Farkı Piramitlerinde Bölgesel Enuç Tespiti

Gaussların farklarının (DoG) her seviyesi için 2. ve 3. görüntülerde bölgesel enu tespiti yapılır. Bölgesel enu tespiti için her piksel, komşusu olan 8 piksel ile ve bir üst ve bir alt ölçekteki iki görüntüdeki komşu 9'ar pikselle, toplamda 26 piksel ile karşılaştırılır (Şekil 3.3). Karşılaştırmaya tabi tutulan piksel eğer komşu 26 pikselin hepsinden büyükse enbüyük, eğer hepsinden küçükse enküçük aday anahtar noktası olarak işaretlenir (Şekil 3.4).



Şekil 3.3. Enu Değer Tespiti



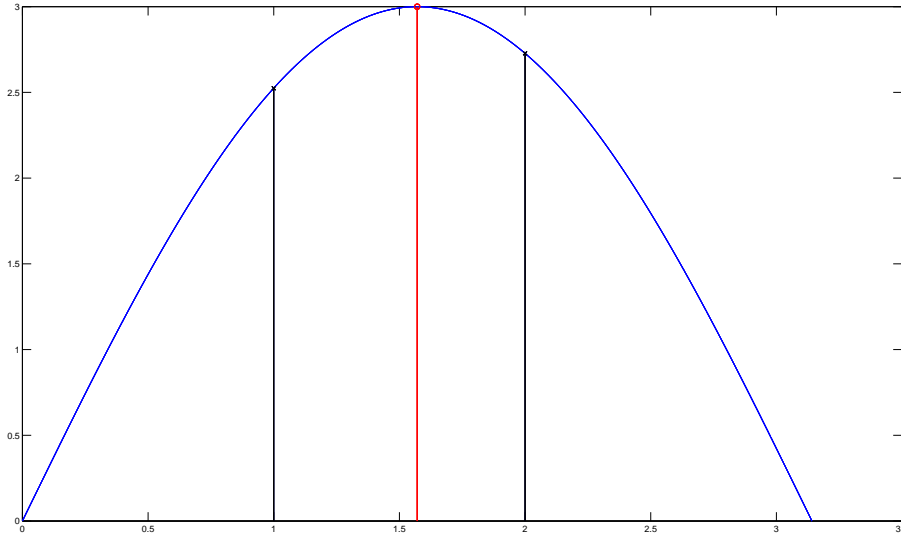
Şekil 3.4. Görüntüdeki Aday Anahtar Noktaları

3.2.3 Hassas Anahtar Noktaları Tespiti

Bulunan aday anahtar noktalarının enu deęerleri esas olarak tamsayı piksel noktalarında olmayıp, genellikle ara piksel deęerlerinde bulunmaktadır [44]. Bunun için ilgili Gauss farkı D görüntüsünde aday anahtar noktalarının bulunduğu piksel koordinatları için Taylor açılımı uygulanarak esas enu nokta deęeri bulunabilir (Şekil 3.5). $\mathbf{x} = (c, r, \sigma)^T$ koordinatındaki piksel deęerine ikinci dereceye kadar Taylor açılımı (3.7) uygulanıp, bu fonksiyonun birinci türevi alınıp sifıra eşitlendiğinde (3.8), $\hat{\mathbf{x}}$ kayması hesaplanır. $\hat{\mathbf{x}}$ kayması \mathbf{x} koordinatına vektörel olarak eklenerek esas enu noktasının yeni koordinatı bulunmuş olur.

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \quad (3.7)$$

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}} \quad (3.8)$$



Şekil 3.5. Taylor Açılımı Uygulandığında Bulunan Enbüyük Deęer

Eđer $\hat{\mathbf{x}}$ kaymasının boyu 0.5'ten büyük ise enu noktanın başka bir piksele daha yakın olduğuna karar verilip bu nokta elenir. Elenmeyen noktalar, eşitlik (3.8) ve eşitlik (3.7) ile birlikte yeniden hesaplanarak piksel deęeri daha önce bulunan koordinatlarda hesaplanır (3.9).

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}} \quad (3.9)$$

En son bulunan bütün enu noktaları dşk kontrast elemesinden geirilerek, boyu 0.03'ten kk olan pikseller elenir. Kesin anahtar noktalar (Şekil 3.6)'da grldğ gibi iřaretlenir.



Şekil 3.6. Görüntüdeki Hassas Tespit Sonrası Kesin Anahtar Noktaları

3.2.4 Kenar Tepkilerinin ve Düşük Kontrastların Elenmesi

Gaussların farkı (DoG) fonksiyonu görüntüdeki kenarlar boyunca yüksek tepkiler vermektedir. Anahtar noktalarının kararlılığı için kenarlarda bulunan bu noktaların elenmesi gerekmektedir [11].

Gauss farkı görüntüsünde zayıf tanımlanmış bir uç noktasının genel olarak kenar boyunca büyük bir asli eğriliği bulunurken, kenara dik yönde küçük bir asli eğriliği bulunmaktadır. Asli eğrilikler 2x2'lik Hessian matrisinden(3.10) hesaplanabilir.

$$\mathbf{H} = \begin{bmatrix} D_{cc} & D_{cr} \\ D_{cr} & D_{rr} \end{bmatrix} \quad (3.10)$$

\mathbf{H} 'nin özdeğerleri D 'nin asli eğrilikleriyle orantılıdır. Özdeğerlerin sadece oranları önemli olduğundan, daha önce kullanılmış bir yaklaşımla [30] özdeğerlerin hesaplanması yükünden kurtulunabilmektedir. α en büyük özdeğer, β en küçük özdeğer olmak üzere, matrisin izinden $Tr(\mathbf{H})$ özdeğerlerin toplamı, matrisin determinantından $Det(\mathbf{H})$ da çarpımları bulunabilir (3.11).

$$\begin{aligned} Tr(\mathbf{H}) &= D_{cc} + D_{rr} = \alpha + \beta \\ Det(\mathbf{H}) &= D_{cc}D_{rr} - (D_{cr})^2 = \alpha\beta \end{aligned} \quad (3.11)$$

Nadir olarak da gerçekleşse, eğer matrisin determinantı negatif bir değerde çıkarsa, eğriler farklı işaretlere sahip olacağından, bu nokta enuç olamayacağından elenir.

ρ en büyük özdeğer α ile en küçük özdeğer β arasındaki oran olmak üzere, matrisin izi ile matrisin determinantı arasındaki oran(3.12) özdeğerlere bağlı olmaktan çıkıp, sadece özdeğerlerin oranlarına bağlı hale getirilmiş olur.

$$\frac{[Tr(\mathbf{H})]^2}{Det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(\rho\beta + \beta)^2}{\rho\beta^2} = \frac{(\rho + 1)^2}{\rho} \quad (3.12)$$

$(\rho + 1)^2/\rho$ değeri kolayca görüleceği gibi ρ değeri ile birlikte artmakta ve özdeğerler birbirine eşit olduğunda en küçük olmaktadır. Asli eğriliklerin birbirine oranları belli bir ρ değerini geçmemesi gerektiğinden,

$$\frac{[Tr(\mathbf{H})]^2}{Det(\mathbf{H})} < \frac{(\rho + 1)^2}{\rho} \quad (3.13)$$

eşitsizliđinin sađlanması yeterli olmaktadır. Yapılan alıřmalarda, $\rho = 10$ seilerek, asli eđrilerin oranları 10'dan byk olan noktaların ıkarıldıđını, kenar tepkilerinin bir ok grntde %75'ten fazlasının elendiđi grlmřtr.

Bu iřlemler sonucunda, aday anahtar noktalarının ara koordinatlardaki deđerleri bulunur, bunlardan bir kısmı gerek enu olmadığı iin, bazıları 0.03 kontrast deđerinden daha dřk olduđundan [11] ve bazıları da kenar tepkisi olduđundan elenir. Kalan noktalar, (řekil 3.7)'de gsterildiđi gibi kararlı kesin anahtar noktaları olarak saklanır.



Şekil 3.7. Görüntüdeki Kenar Tepkileri Elendikten Sonraki Kesin Anahtar Noktaları

3.3 Yönelim Atama

Her bir anahtar noktaya, bölgesel özelliklerine dayalı uygun bir yönelim belirleyerek, anahtar noktası yönelimliyle birlikte tanımlanmış ve görüntünün dönmesine karşı değişimsizlik kazanılmış olur. Bu yaklaşım daha önceki çalışmalarda [46] da yapılan yönelim değişimsizlikle örtüşmektedir. Bu yaklaşımın tek dezavantajı tanımlayıcıları oluştururken görüntüdeki bazı bilgilerin kaybedilmesi ve bunun tanımlayıcının yapısına yansımalarıdır.

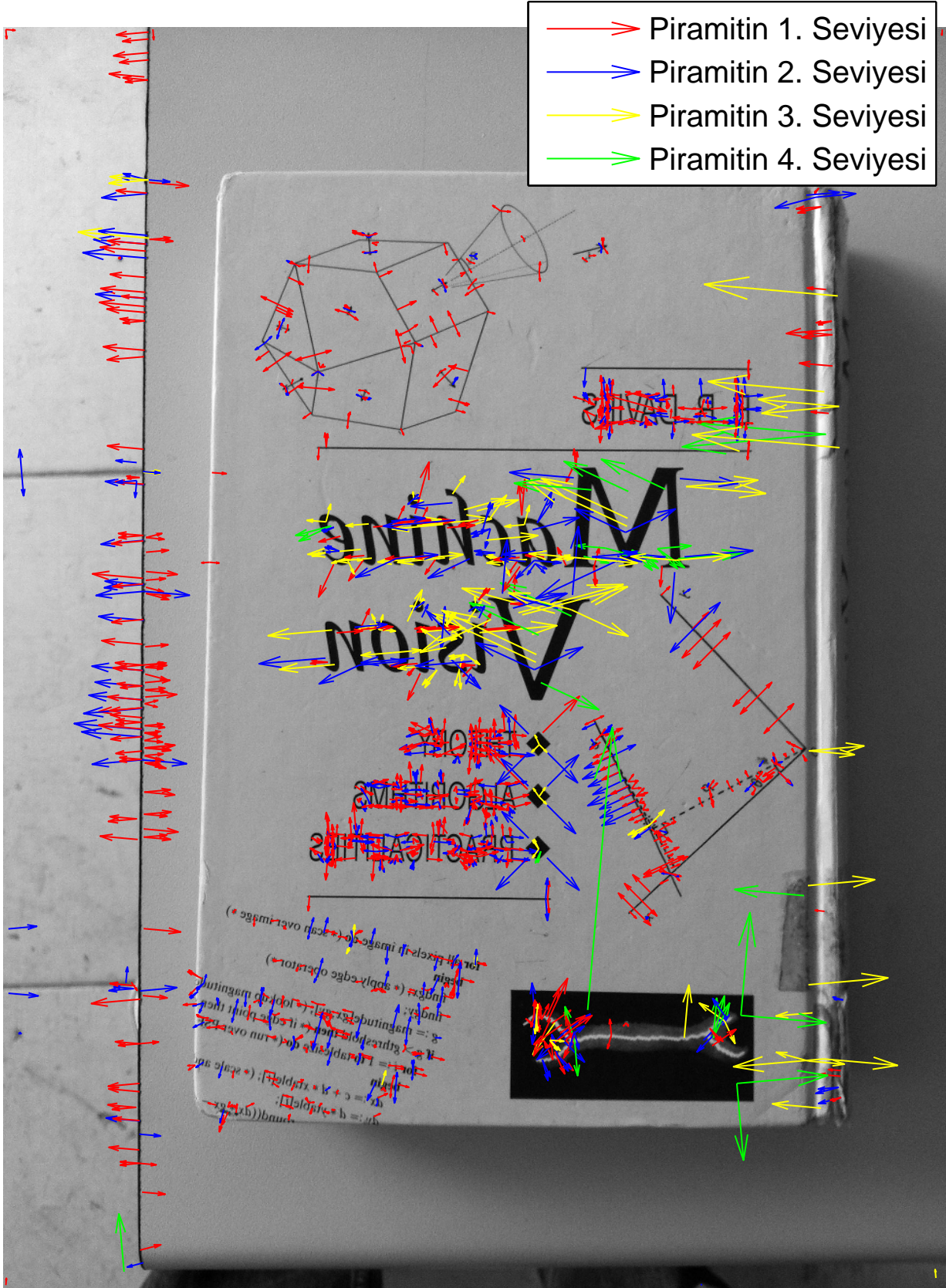
Anahtar noktanın yönelimini atamak için bir dizi yöntem içerisinde en kararlısı, bir bölge seçilip bu bölgenin ağırlıklandırılarak yöneliminin tanımlanması olarak bulunmuştur. Bunun için anahtar noktasına en yakın ölçekteki görüntü, L , seçilip bütün işlemler ölçek bağımsız hale getirilmektedir. Bu ölçekteki her bir anahtar noktasının etrafında 16×16 'lık bir bölge seçilerek buradaki piksellerin gradyan büyüklükleri ve yönelimleri hesaplanmaktadır (3.14).

$$\begin{aligned} m(c, r) &= \sqrt{(L(c+1, r) - L(c-1, r))^2 + (L(c, r+1) - L(c, r-1))^2} \\ \theta(c, r) &= \arctan((L(c, r+1) - L(c, r-1)) / (L(c+1, r) - L(c-1, r))) \end{aligned} \quad (3.14)$$

Anahtar noktasının etrafındaki kümede hesaplanan gradyan değerlerinden bir histogram çıkarmak için, 360° 'lik yönelim aralığı 10° 'lik 36 kümede toplanır. Her bir pikselin gradyan büyüklüğü, standart sapması, σ , bulunduğu ölçeğin 1.5 katı olan bir Gauss penceresinden geçirilerek ilgili yönelim kümesine eklenir. Bu işlem anahtar noktaya yakın olan piksellerin etkisinin fazla, uzak olanların etkisinin az olması için yapılmaktadır. Oluşturulan histogram kümelerinden değeri en büyük olan, anahtar noktasının ilk yönelimi olarak belirlenir.

Daha sonra, oluşturulacak tanımlayıcının kararlılığını ve tanımlama kuvvetini arttırmak için, eğer varsa, en büyük histogram kümesinin 0.8 katından daha büyük kümeler de anahtar noktasının yönelimi olarak atanır. Bazı durumlarda histogram kümelerinin değerleri birbirine çok yakın olduğundan 4 veya daha fazla yönelim çıkabilmekte, bu da kararlılığı arttırmak yerine azaltmakta, ayrıca işlem yükünü de arttırmaktadır; bu durumda işlem yükünü arttırmamak adına çalışmada yönelim sayısı en büyük 3 kümeyi yönelim olarak atayacak şekilde sınırlandırılmıştır.

Sonuç olarak her anahtar noktasına en az 1, en çok 3 yönelim atanmış olur. Bu yönelimler bir sonraki aşamada tanımlayıcı vektör oluşturulurken kullanılır [11].



Şekil 3.8. Görüntüdeki Anahtar Noktaların Gradyan Yönelimleri

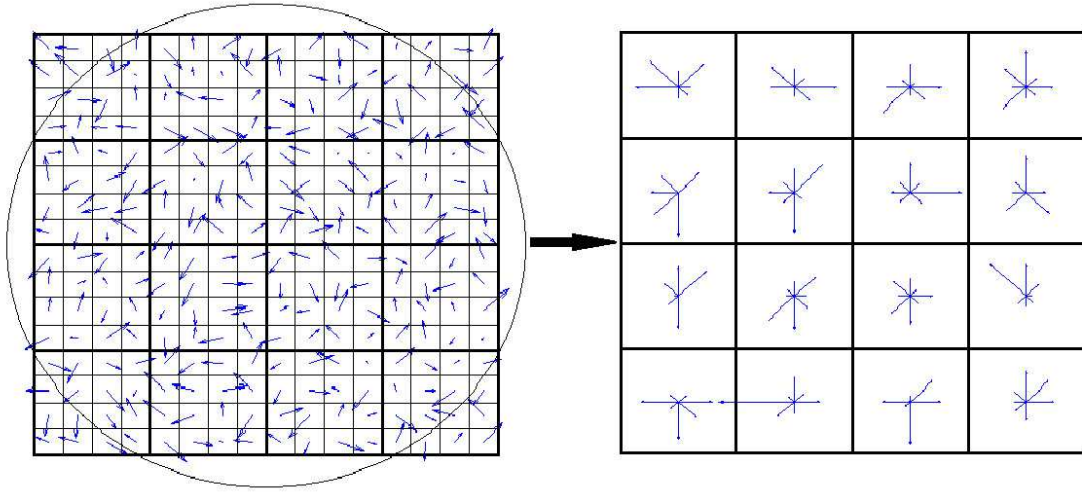
3.4 Bölgesel Görüntü Tanımlayıcı

Daha önceki aşamalarda tanımlayıcıya konuma, ölçeğe ve dönmeye karşı değişimsizlik kazandırılmıştı. Bu aşamada ışıklandırma veya $3B$ görüş açısı gibi kalan diğer değişkenlere karşı değişimsizlik elde edilmekte ve tanımlayıcı vektörler oluşturulmaktadır.

Önceki başka çalışmalarda en çok kullanılan yaklaşımlardan birisi, uygun ölçekte anahtar noktalarının etrafındaki bölgenin piksel şiddetlerini normalize edip, eşleştirme için basitçe ilintilendirmeye dayanmaktadır. Ancak basitçe ilintilendirme birçok değişime karşı çok hassas olduğundan farklı yöntemler geliştirilmiştir. Başka bir yaklaşımla beyin zarındaki görmeyi sağlayan kompleks nöronlar modellenmiş, bu nöronlar gibi belirli bir yönelimde, uzaysal sıklıkta gradyanlara tepki veren bir yöntem geliştirilmiştir. Çalışmada bu yöntem üzerinden geliştirilen bir teknik kullanılmıştır [11] [22].

3.4.1 Tanımlayıcı Vektörü Oluşturma

Tanımlayıcı vektör (Şekil 3.9) görüldüğü gibi oluşturulur. Tanımlayıcı vektör anahtar noktaları eşleme için kullanılan, $4 \times 4 \times 8 = 128$ uzunluğunda bir vektördür.



Şekil 3.9. Tanımlayıcı Vektörün Oluşturulması

Bunun için öncelikle daha önceden yapılmış olan piksel gradyanları kullanılır. Anahtar noktası etrafında aynı ölçekteki L görüntüsündeki gradyanlar 16×16 bir kare pencere içerisinde tanımlanır. Dönmeye karşı değişimsizlik elde etmek için anahtar

noktasının etrafındaki alan daha önce bulunmuş yönelime göre döndürülür, ayrıca bu alanda bulunan bütün piksel gradyanları da anahtar noktasının yönelimine göre döndürülür.

Döndürülen bölge yönelim atama aşamasındaki gibi, $\sigma = 0.5 * \text{KenarUzunluğu} = 8$ standart sapması olan Gauss penceresiyle ağırlıklandırılır. Ağırlıklandırmanın amacı, anahtar noktasına uzak piksellerin etkisinin az olmasının sağlayarak, anahtar noktasındaki penceresinin genellikle yanlış kaydetmelerden oluşan küçük konum değişikliklerinin tanımlayıcıya olan etkisini azaltmaktır.

(Şekil 3.9)'da gösterildiği gibi, anahtar noktası etrafındaki 16x16 kare pencere 4x4'lük 16 ayrı alt pencereye ayrılır. Her bir alt pencerede, ağırlıklandırılmış gradyanların yönelim olarak 45°'lik 8 kümelik histogramı çıkarılır. Daha sonra buradan çıkarılan histogramlar 16 alt pencere birleştirilerek tek boyuta indirilip, uzunluğu 128 olan bir vektöre dönüştürülür. Yöntemin ilk tanıtıldığı çalışmada [22] daha çok kümeleme kullanmanın tanımlayıcı vektörün bozukluklara olan hassasiyetini arttırdığı tespit edilmiştir; ayrıca pencere büyüklüğünü arttırmak da (5x5) aynı şekilde gürültü ve bozukluklara karşı hassasiyeti arttırmıştır.

Vektör, oluşturulduktan sonra son bir işlemde daha geçirilir. Bu işlemde vektör, ışıklandırma değişimlerine karşı güçlendirilir. Bunun için oluşturulan vektör birim uzunluğa normalize edilir. Bunun nedeni görüntü kontrastında değişimin bütün pikselleri belli bir sabitle çarpıp, gradyanları da aynı değerle çarpmasıdır ve bu değişim vektörün normalize edilmesiyle büyük oranda geçersiz kılınabilir. Ayrıca parlaklık-taki değişim bütün piksel değerlerine bir sabit ekleyeceği için gradyan değerlerinde değişim çok düşük olacağından tanımlayıcı vektör ışıklandırmadaki yönsel ve kuvvetsel değişikliklerden etkilenmemiş olur. Yine de, kamera doyumu noktasından veya yönlendirmeyi farklı miktarlarda etkileyen 3B yüzeylerin aydınlatılmasından kaynaklı doğrusal olmayan ışıklandırma değişimleri meydana gelebilir. Bu değişimler de bazı piksel gradyanlarının uzunluklarının büyük ölçüde etkilemekte, yönelimlerini fazla değiştirmemektedir; bu sebepten büyük gradyan uzunluklarının etkilerini azaltmak amacıyla normalize edilmiş vektör 0.2 eşik değerinden geçirilip, birim uzunluğa yeniden normalize edilir [11].

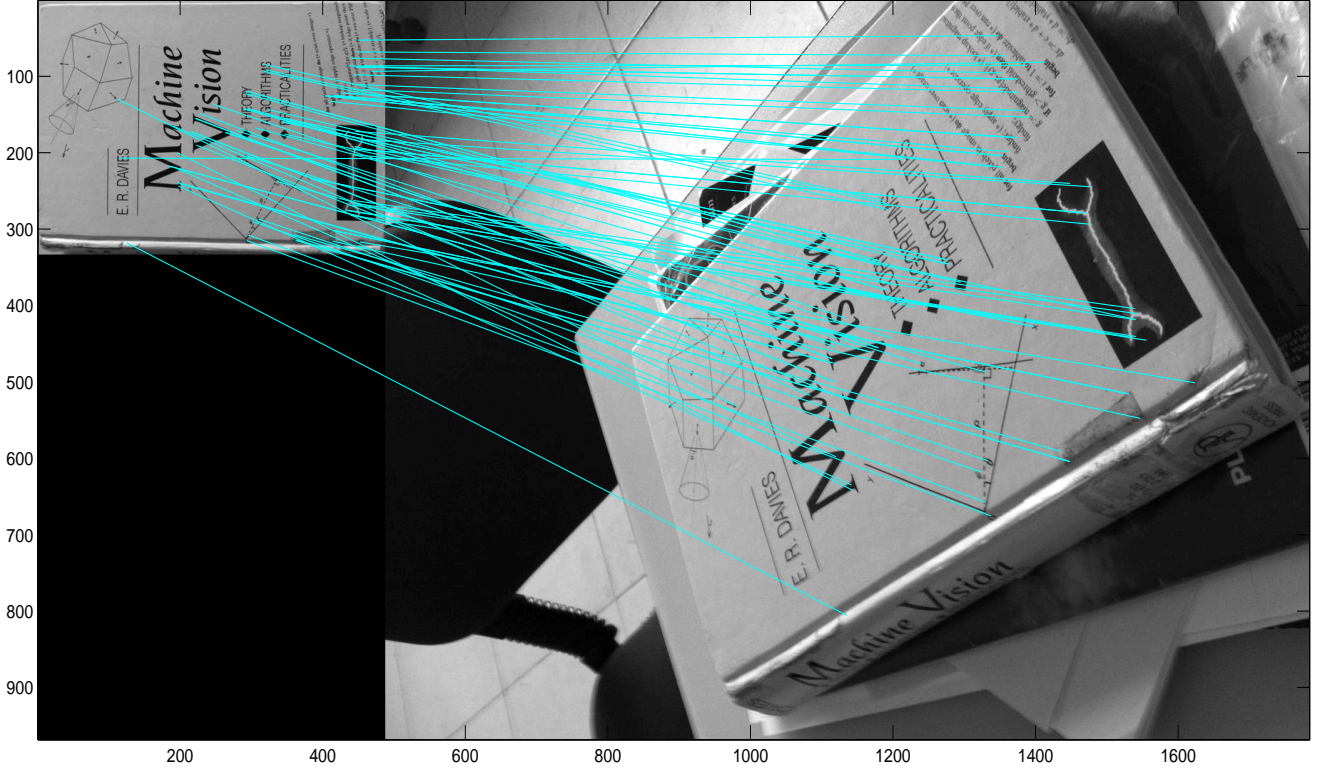
4. ÖZELLİK TANIMLAYICILARI EŞLEME VE HIZLANDIRMA

Özellik tanımlayıcılarını doğru oluşturmak kadar bu tanımlayıcıları doğru eşlemek de çok önemlidir. Daha önce oluşturulmuş bir tanımlayıcı vektör veri tabanındaki verilerle eşleme yapmak istediğimiz resimden çıkardığımız özellik tanımlayıcı vektörler tek tek eşlenmelidir. Ölçek Değişimsiz Özellik Dönüşümü ve tanımlayıcı vektör oluşturulan diğer yöntemlerle nesne eşlemek için yapılan çalışmalarda [11] [20], en doğru sonucun tanımlayıcı vektörleri arasında Öklit uzaklığının (4.1) bulunmasıyla alınabileceğini göstermiştir. En küçük Öklit uzaklığına sahip eşleşme doğru eşleşme olasılığı en yüksek eşleşmedir. Bu test, herbir tanımlayıcı vektör için yapılmaktadır; işlem yükünü azaltmak için Öklit uzaklıklarının kareleri karşılaştırılabilir.

$$d = \sqrt{(\bar{x}' - \bar{x})^2} \quad (4.1)$$

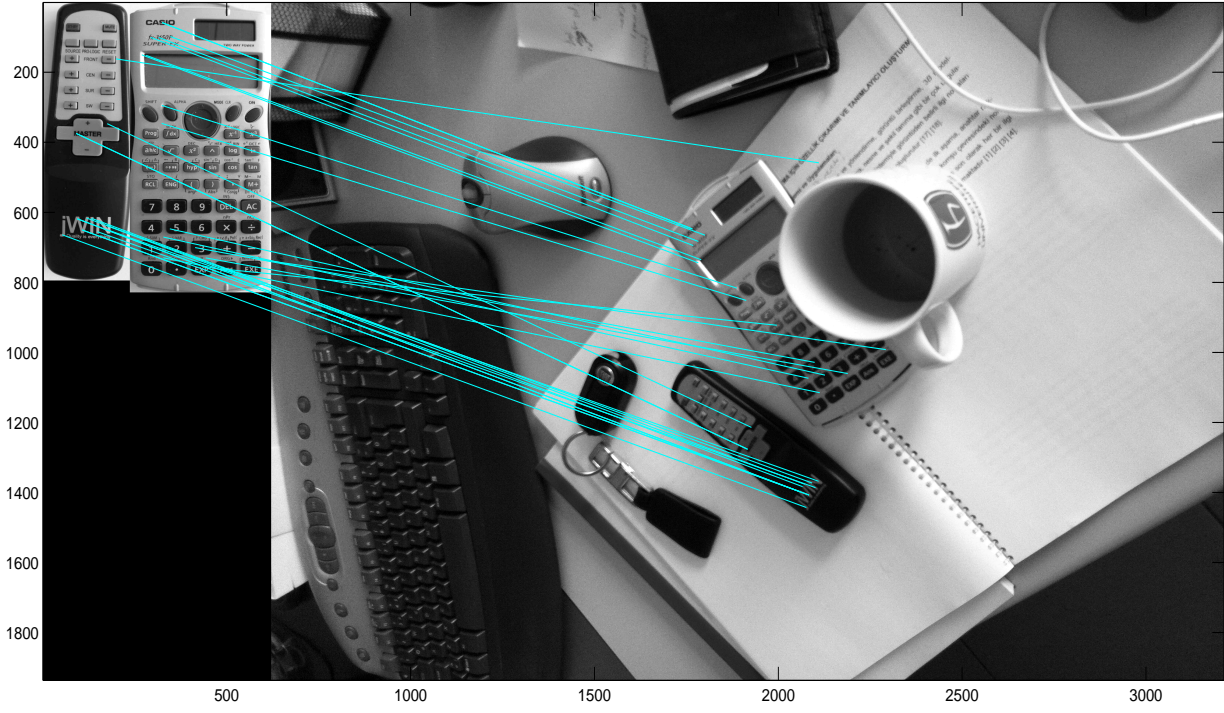
İki tanımlayıcı vektörün doğru eşleşme olasılığı görüntü veri tabanının büyüklüğüne bağlıdır. Veri tabanı 2000 civarında tanımlayıcı vektörden oluştuğu durumda doğru eşleşme olasılığı 0.8 iken; veri tabanı büyüdükçe bu olasılık azalmakta, 100000 vektörlü veri tabanında bu olasılık 0.75'lere kadar düşmektedir.

Bu başarıyı arttırmak için önerilen yöntem en yakın test edilen vektöre en yakın vektörle en yakın ikinci vektör arasındaki orandır. Birinci en yakın vektör ile ikinci en yakın vektör arasındaki oran 0.8'den büyükse yanlış bir eşleşme olduğu varsayılır. Bu uygulamayla birlikte doğru eşleşmelerin sadece %5'ini elerken, yanlış eşleşmelerin %90'ını elemektedir. Daha önceki doğru eşleşme olasılığı, veri tabanının büyüklüğüyle çok değişmeyip yaklaşık 0.94 olmaktadır [11]. (Şekil 4.1)'de görüldüğü gibi sisteme tanıtılan nesne ana resimden rahatlıkla ayırt edilebilmektedir, ayrıca yanlış eşleşme de görülmemektedir.

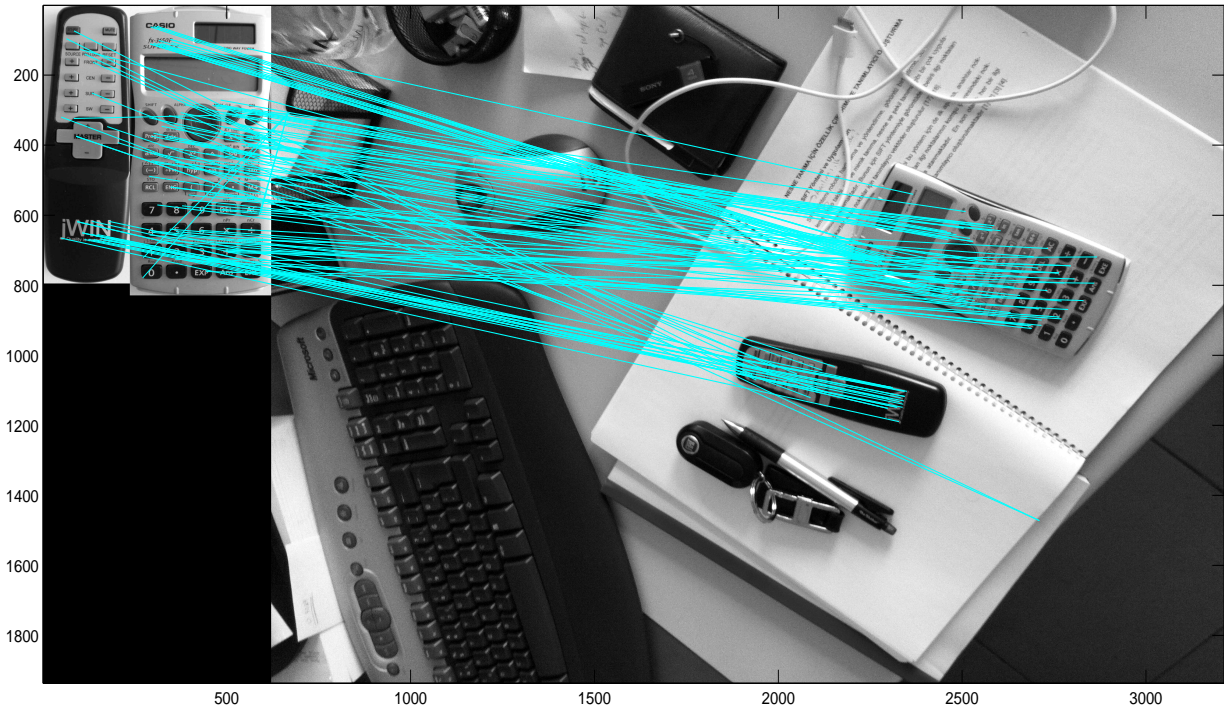


Şekil 4.1. Döndürülmüş, Ölçeklendirilmiş, Işıklandırması ve 3B Görüş Açısı Değiştirilmiş Nesnenin Eşleştirilmesi

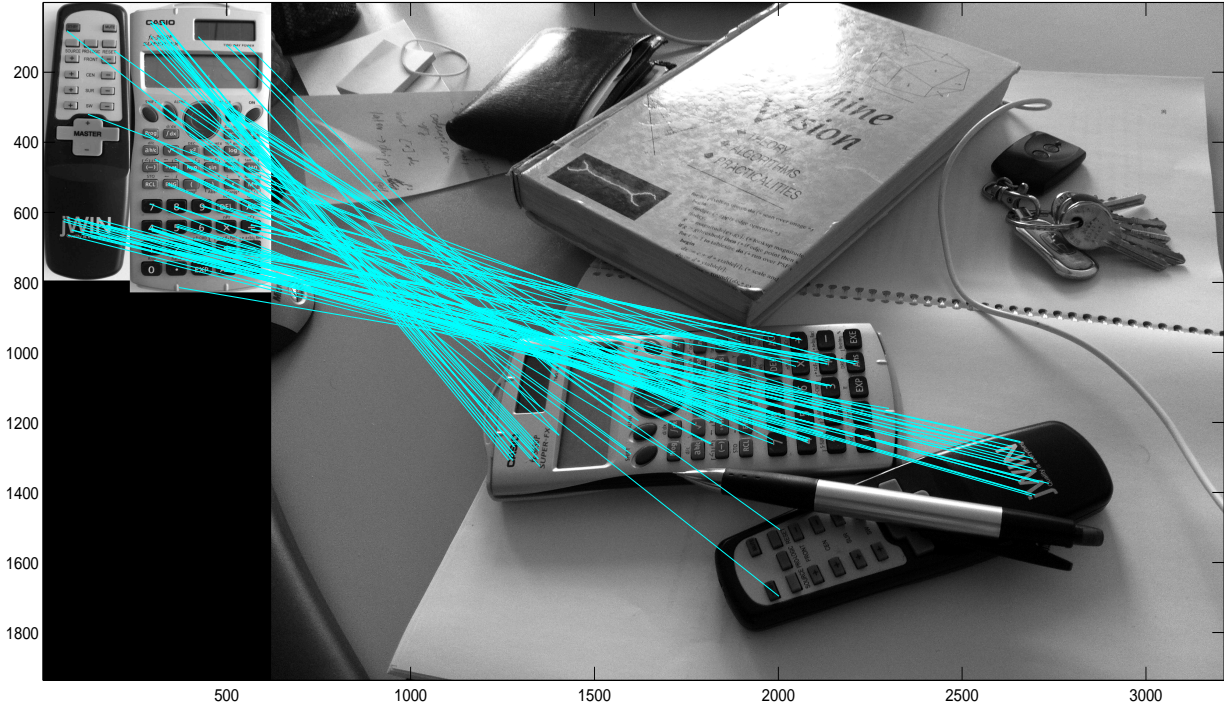
Şekil 4.2, Şekil 4.3 ve Şekil 4.4'te sisteme iki nesne tanıtılmış, bunlardan Şekil 4.2 ve Şekil 4.3'de görüntüleme aynı açılarla yapılmış, Şekil 4.4'te ise görüntüleme açısı ve buna bağlı olarak ışıklandırma da değişmiştir. Yine bu görüntülerden Şekil 4.2 ve Şekil 4.4'te bir nesnenin %10 ila %25'lik kısımlarının görüntüsü engellenmiştir. Şekil 4.2 ve Şekil 4.3'de (Çizelge 4.1)'da belirtildiği gibi az sayıda da olsa yanlış eşleşmeler olmuş ancak nesnelerin tanınma kriterleri sağlanmıştır. Şekil 4.4'te ise hiç yanlış eşleşmeye rastlanmamış, başarımlar kriterleri yine bütünüyle sağlanmıştır.



Şekil 4.2. Kısmi Engelleme Olduğunda Nesne Tanıma

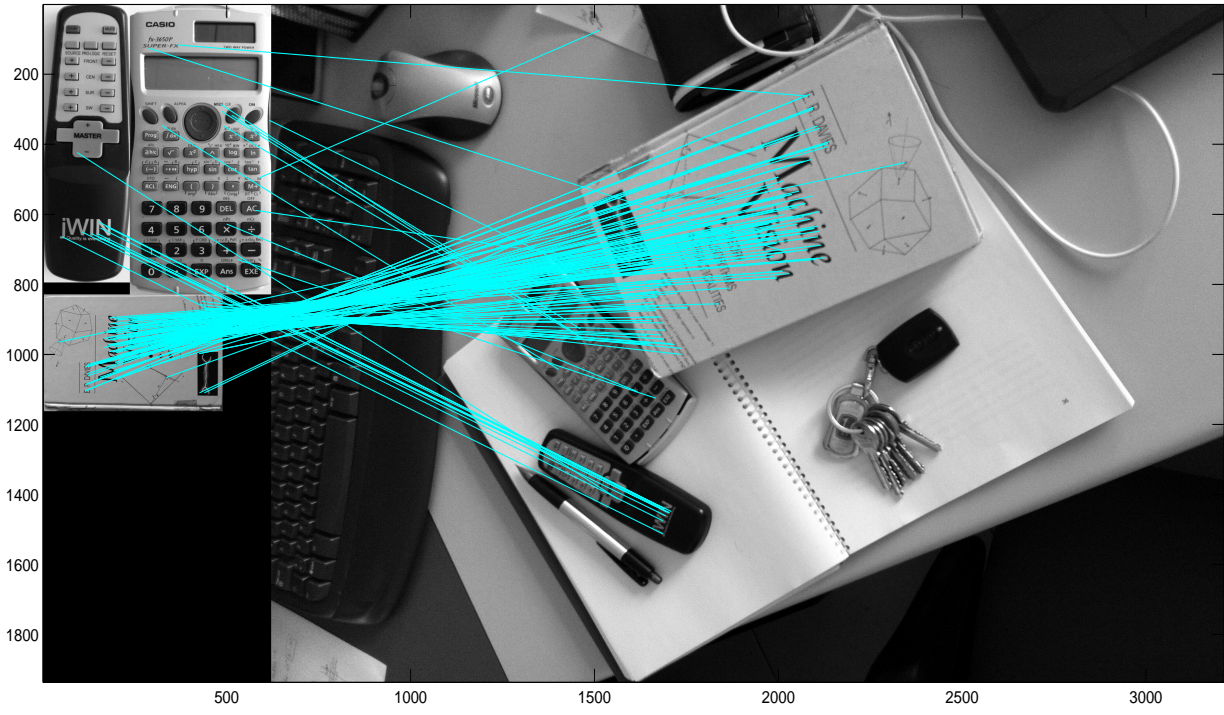


Şekil 4.3. Yönelimler Değiştiğinde Nesne Tanıma

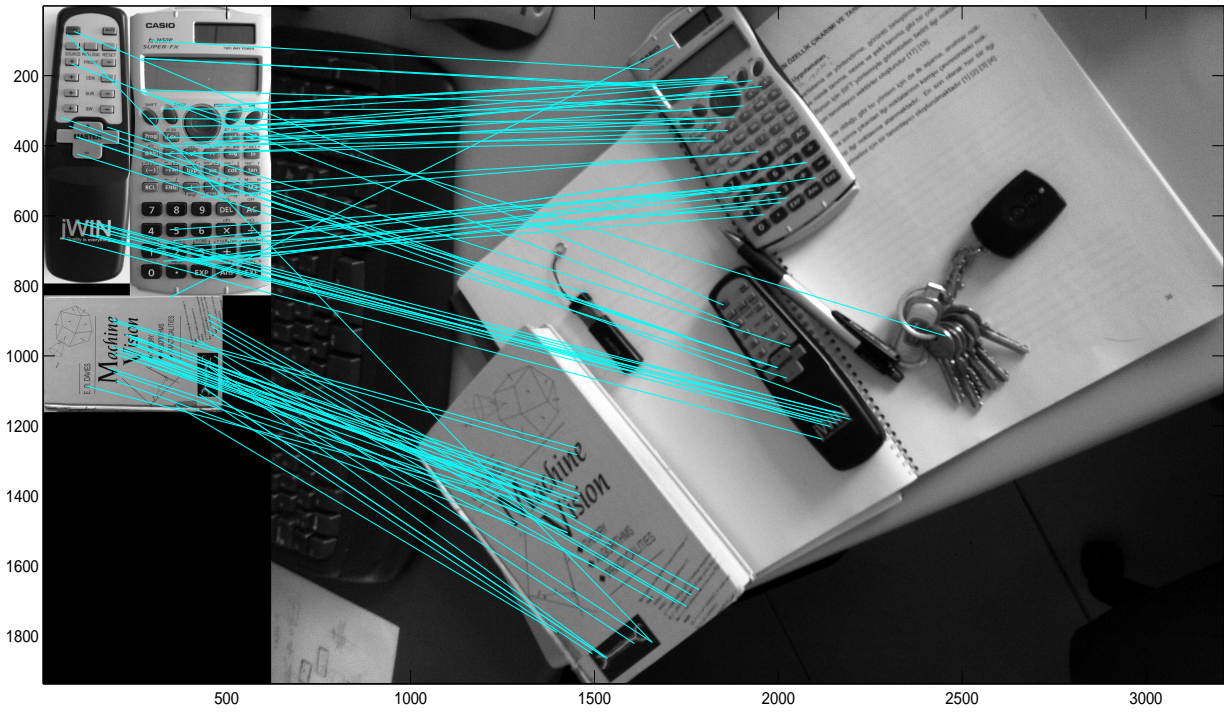


Şekil 4.4. Bakış Açısı ve Işıklandırma Değiştiğinde ve Kısmi Engelleme Olduğunda Nesne Tanıma

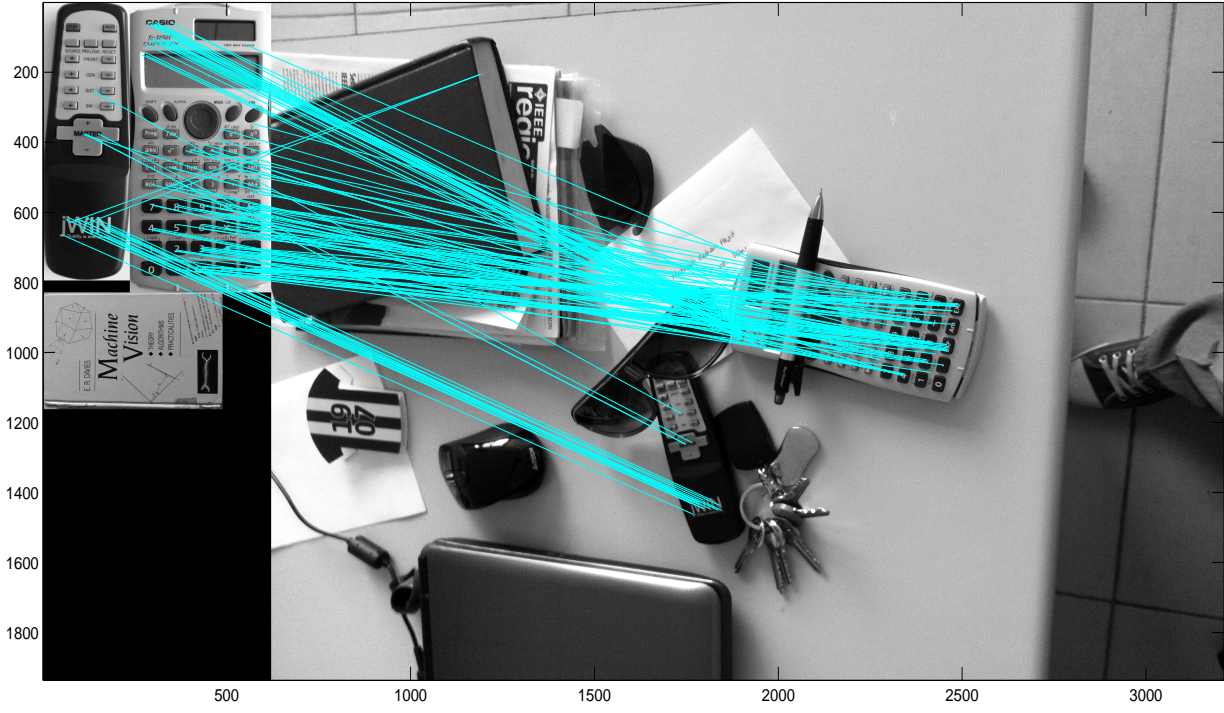
Şekil 4.5'ten Şekil 4.10'a kadar olan görüntülerde ise tanıma işlemlerinin zorlaşması adına üçüncü bir nesne tanıtılmış ve farklı görüntülerle eşleştirmesi yapılmıştır. Şekil 4.5 ve Şekil 4.6'da görüntüler tepeden, kısmen bulanık çekilmiş ve bu resimlerde nesnelerin yerleri değiştirilmiştir; (Çizelge 4.1)'de belirtildiği gibi yanlış eşleşmeler yok denecek kadar az, nesnelerin tanıma kriterleri sağlanmıştır. Şekil 4.7 ve Şekil 4.8'de önce ışıklandırma değiştirilmiş, sonra görüntünün görüş açısı değiştirilmiş, her ikisinde de nesnelere %10 ila %15 arasında engelleme uygulanmış, çok az bir yanlış eşleşme olmuş, nesne tanıma kriterleri sağlanmıştır. Şekil 4.9'da ışıklandırma koşulları değişmiş, ve nesnelerdeki bulanıklık azaltılmış, eşleştirme başarımları hiç yanlış eşleştirme olmayacak şekilde artmıştır. Şekil 4.10'da ise Şekil 4.5'e göre bulanıklık azaltılmış, buna bağlı olarak eşleştirme başarımları artmıştır.



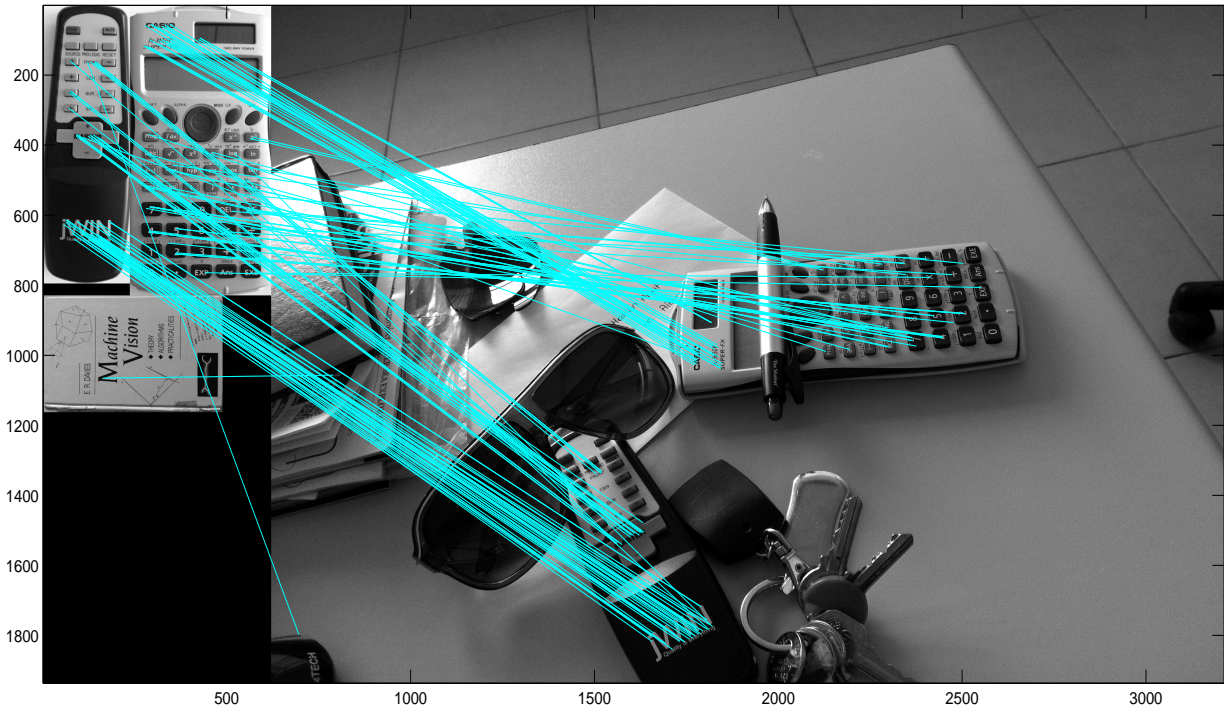
Şekil 4.5. Nesne Sayısı Arttırıldığında ve Yönelim ve Işıklandırma Değişirildiğinde Nesne Tanıma



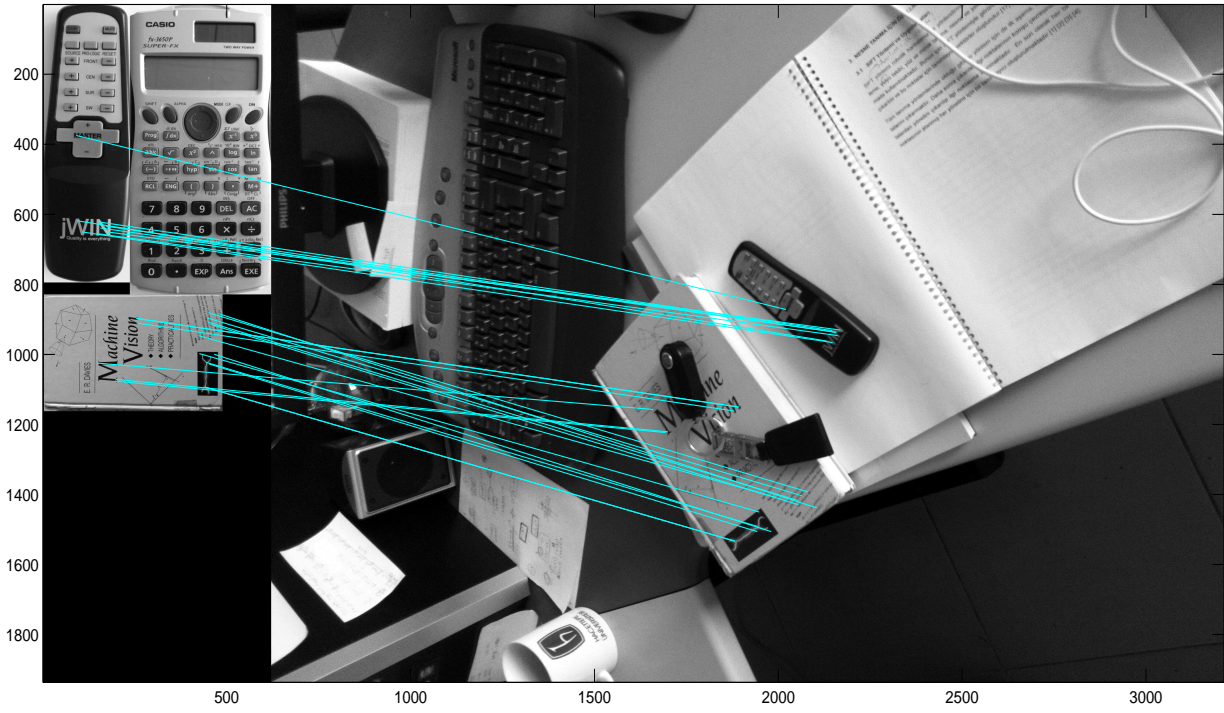
Şekil 4.6. Nesne Sayısı Arttırıldığında ve Yönelim Değiştiğinde Nesne Tanıma



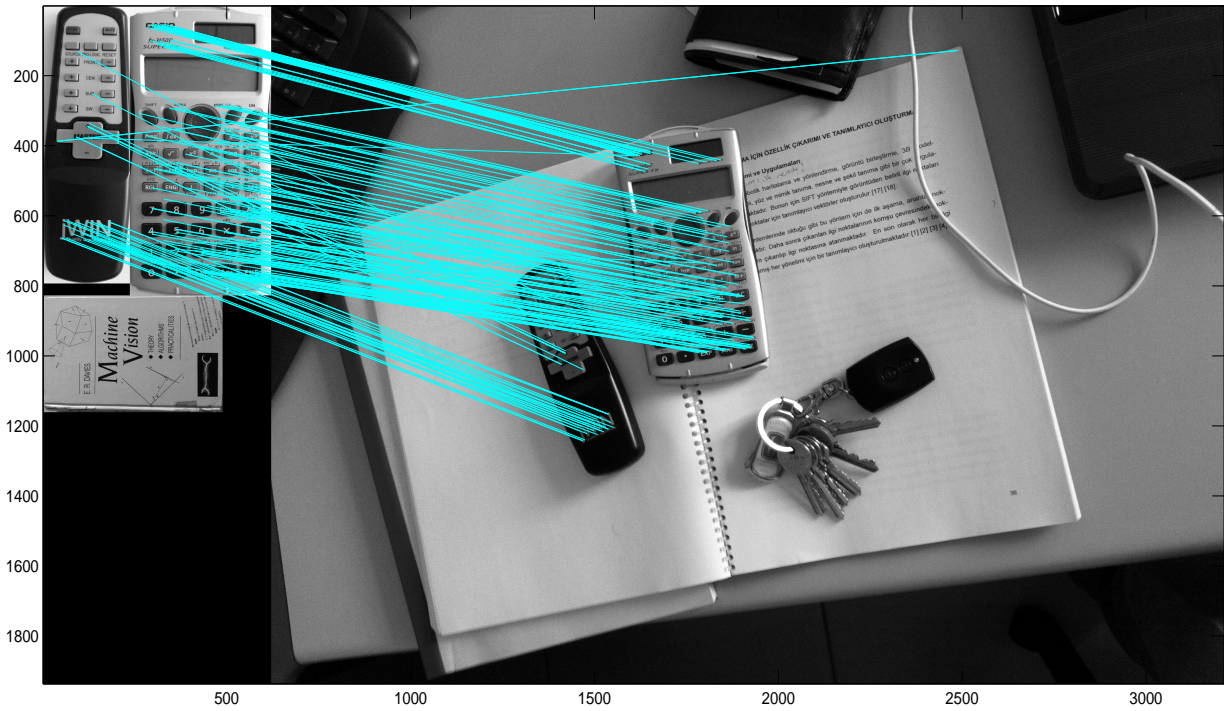
Şekil 4.7. Nesne Sayısı Arttırıldığında ve Bakış Açısı Değiştiğinde Nesne Tanıma



Şekil 4.8. Nesne Sayısı Arttırıldığında, Bakış Açısı ve Işıklandırma Değiştiğinde Nesne Tanıma

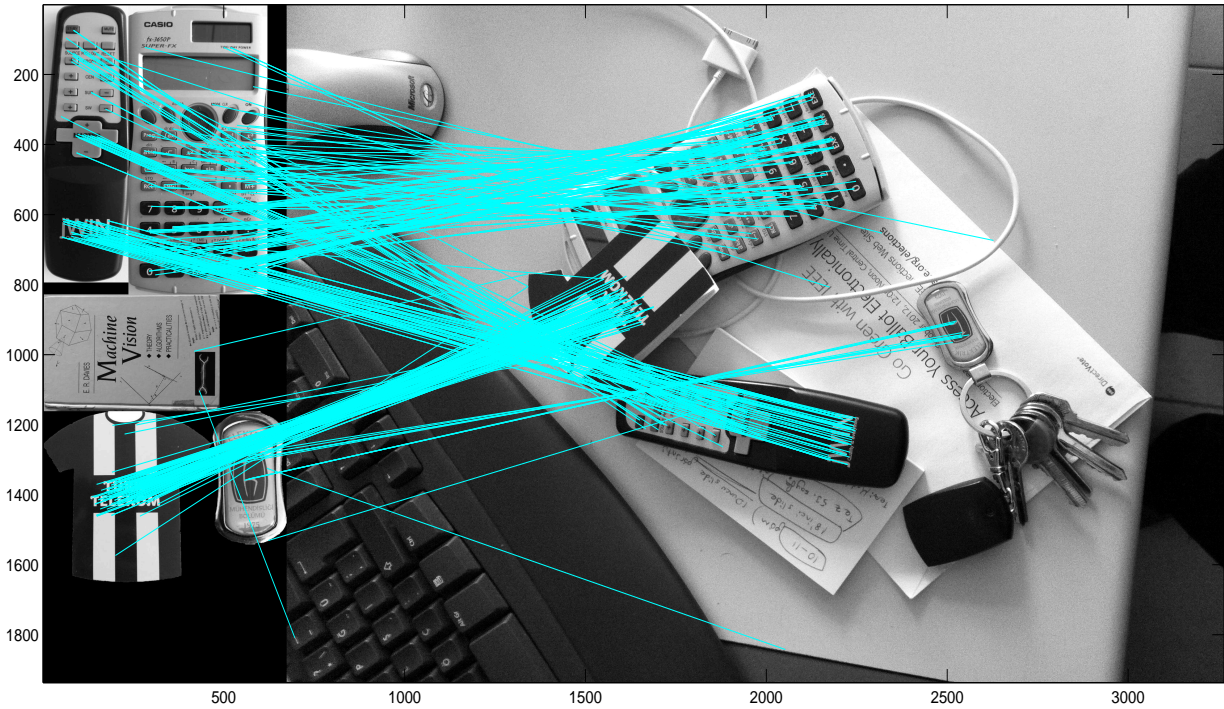


Şekil 4.9. Nesne Sayısı Arttırıldığında, Bakış Açısı ve Işıklandırma Değiştiğinde Nesne Tanıma

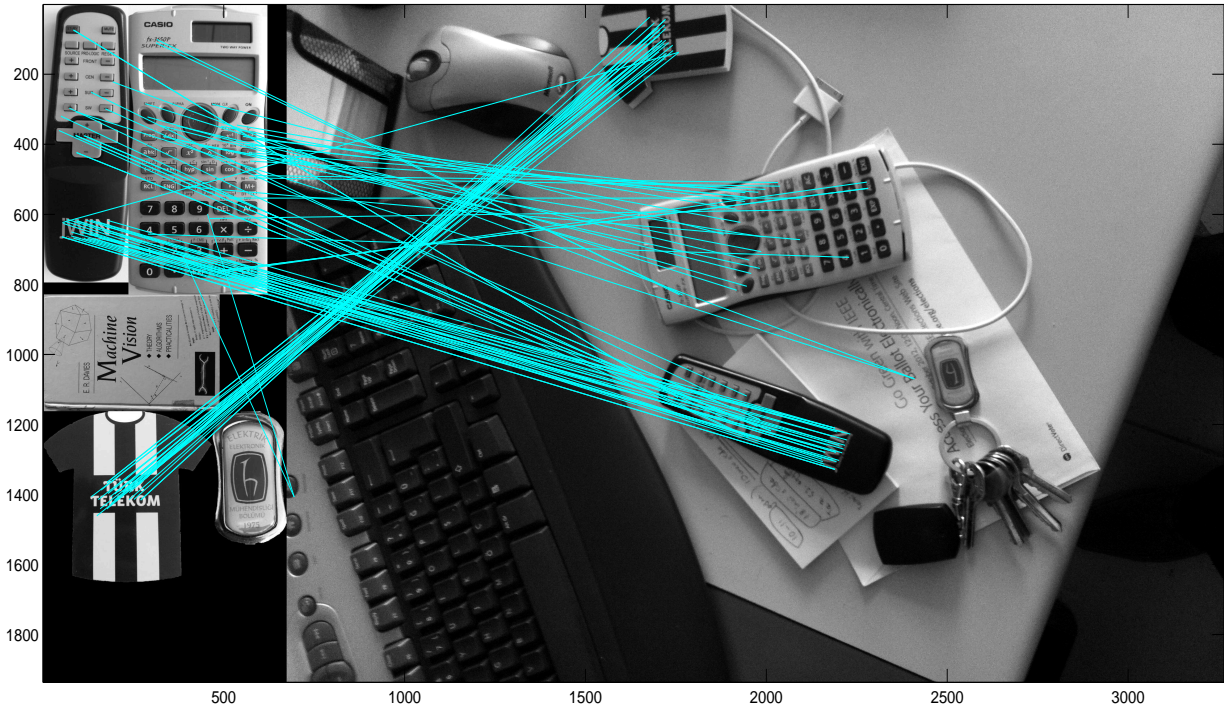


Şekil 4.10. Nesne Sayısı Arttırıldığında, Bakış Açısı ve Işıklandırma Değiştiğinde Nesne Tanıma

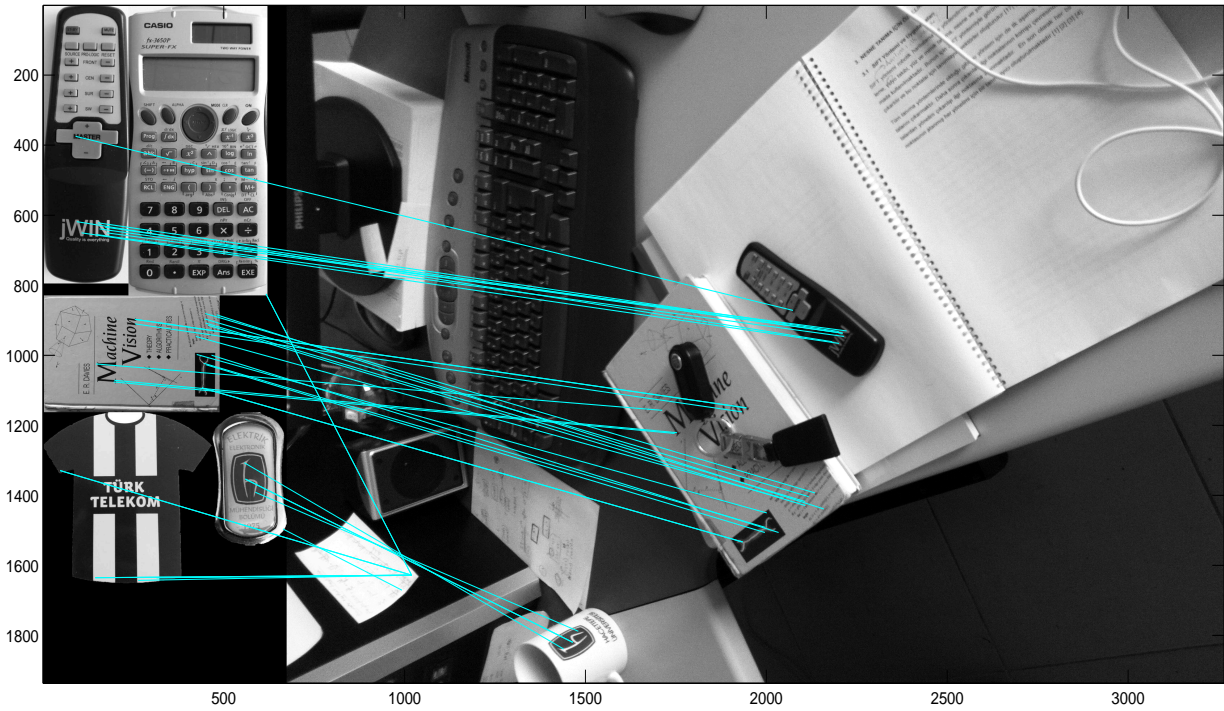
Şekil 4.11'den Şekil 4.15'e kadar olan görüntülerde yine işlemleri zorlaştırmak adına iki yeni nesne sisteme tanıtılmış ve eşleştirme işlemleri tekrarlanmıştır. Şekil 4.11 ve Şekil 4.12'de görüntülerde nesnelere yerleri ve ölçek değiştirilmiştir, ölçek çok küçüldüğünden Şekil 4.12'de anahtarlık görüntüsü tanınamamıştır, ancak yine de eşleştirme kriterleri sağlanmış diğer nesnelere başarımlar korunmuş ve diğer nesnelere doğru tanınmıştır. Şekil 4.13, Şekil 4.14 ve Şekil 4.15'te görüntülere %10 ila %30 arasında engelleme uygulanmış, farklı açılardan ve farklı ışıklandırma koşullarında nesnelere başarıyla tanınmış, (Çizelge 4.1)'de gösterildiği gibi eşleşme başarımları



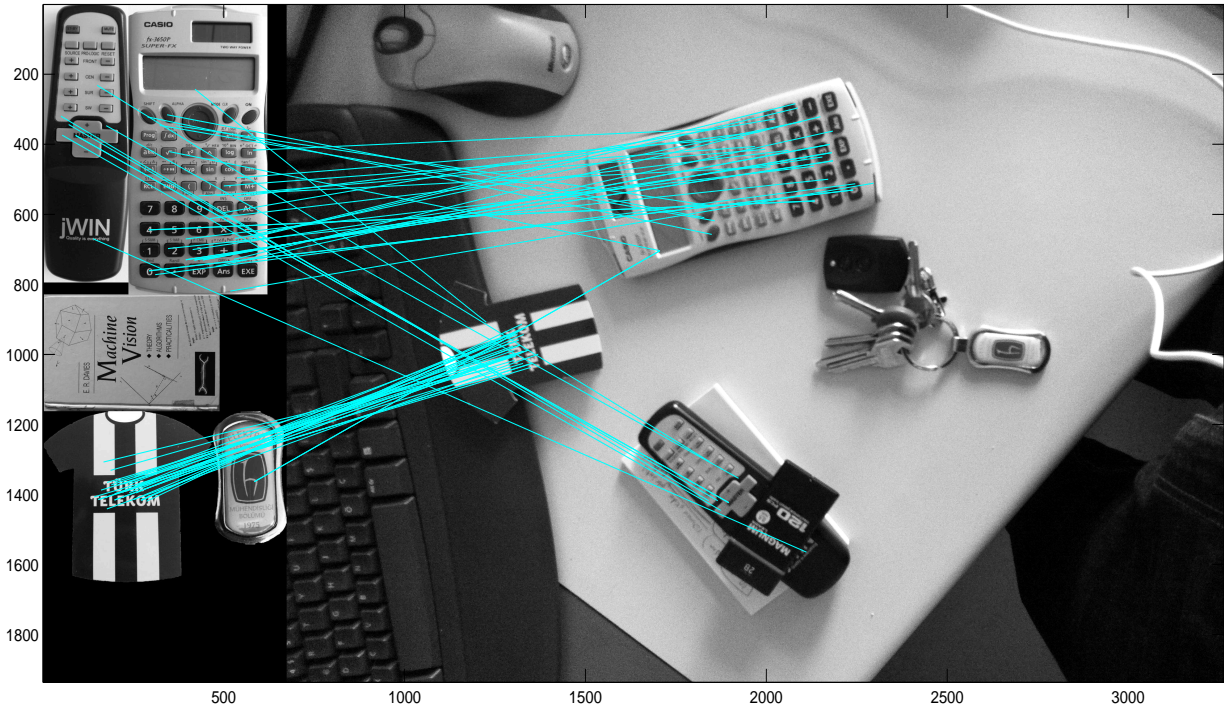
Şekil 4.11. Nesne Sayısı Arttırıldığında ve Işıkları Arttırıldığında Nesne Tanıma



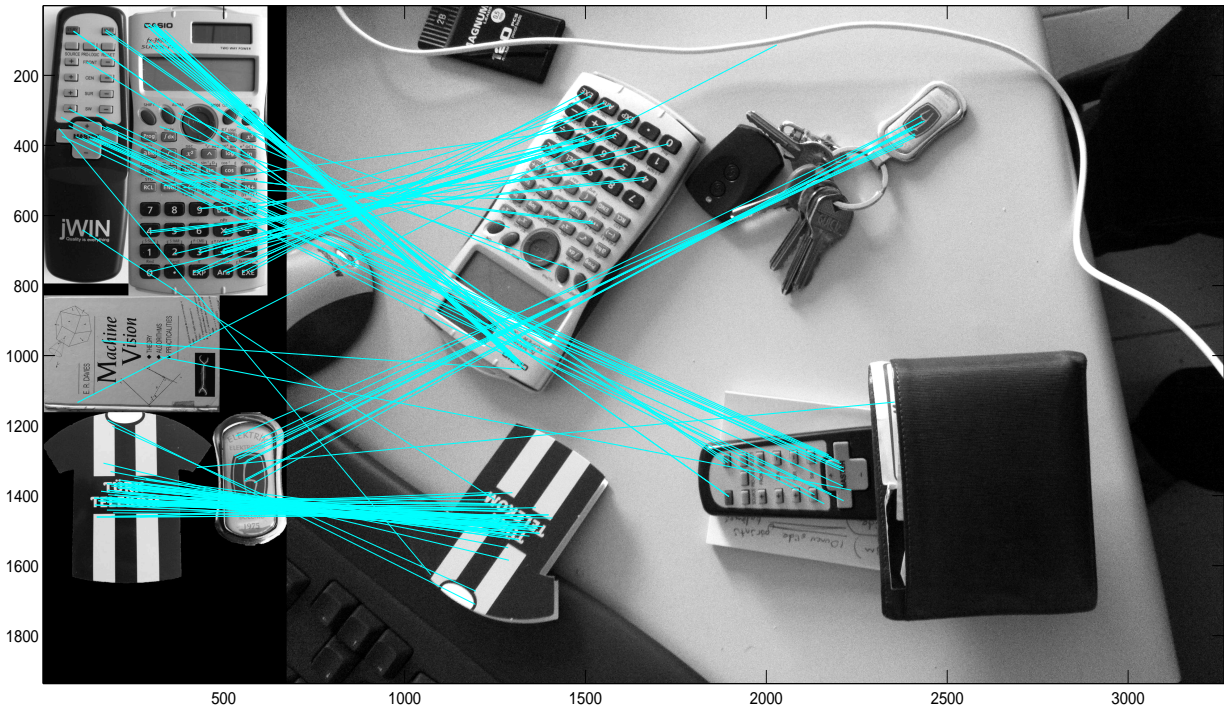
Şekil 4.12. Nesne Sayısı Arttırıldığında ve Koşullar Değiştiğinde Nesne Tanıma



Şekil 4.13. Nesne Sayısı Arttırıldığında ve Işıklandırma ve Görüş Açısı Değiştiğinde Nesne Tanıma



Şekil 4.14. Nesne Sayısı Arttırıldığında ve Işıklandırma Değiştiğinde ve Kısmi Engelleme Olduğunda Nesne Tanıma



Şekil 4.15. Nesne Sayısı Arttırıldığında ve Kısmi Engelleme Olduğunda Nesne Tanıma

Yukarıdaki örnek görüntülerde görüldüğü gibi nesnelere sisteme tanıtılmış ve bu test kullanılarak yüksek eşleşme başarımla nesnelere ana resimle eşlenmiştir, bu başarımlar (Çizelge 4.1)'de gösterilmiştir. Orijinal çalışmalarla [11] [22] karşılaştırıldığında, doğru eşleşme başarımının %98 olduğu görülmekte, çalışmamızda ise sonuçların hemen hemen hiç değişmediği görülmektedir.

Eşleşme Görüntüsü	Eşleşen Nokta Sayısı	Yanlış Eşleşme Yüzdesi
Şekil4.2	1 yanlış, 29 doğru eşleşme	%3.3
Şekil4.3	7 yanlış, 115 doğru eşleşme	%5.7
Şekil4.4	0 yanlış, 126 doğru eşleşme	%0
Şekil4.5	3 yanlış, 162 doğru eşleşme	%1.8
Şekil4.6	0 yanlış, 228 doğru eşleşme	%0
Şekil4.7	2 yanlış, 135 doğru eşleşme	%1.5
Şekil4.8	3 yanlış, 125 doğru eşleşme	%2.3
Şekil4.9	0 yanlış, 26 doğru eşleşme	%0
Şekil4.10	1 yanlış, 183 doğru eşleşme	%0.5
Şekil4.11	10 yanlış, 336 doğru eşleşme	%2.9
Şekil4.12	6 yanlış, 90 doğru eşleşme	%6.3
Şekil4.13	3 yanlış, 36 doğru eşleşme	%7.7
Şekil4.14	1 yanlış, 72 doğru eşleşme	%1.3
Şekil4.15	3 yanlış, 161 doğru eşleşme	%1.8
Toplam	40 yanlış, 1824 doğru eşleşme	%2.1

Çizelge 4.1. Eşleşme Başarımı Tablosu

En yakın komşu araması vektör sayısı çok olduğu zaman verimli bir arama yöntemi olmaktan çıkmaktadır, bunun yerine KD ağacı algoritması kullanılabilir [47], ancak KD ağacı algoritması da vektör boyu 10'dan büyük olduğu zaman en yakın komşu aramasına göre bir kazanım sağlamamaktadır [48] [49]. Veri tabanı büyüdükçe verimi arttırabilmek için KD ağacı arama algoritmasının değiştirilmiş bir biçimi olan Eniyi İlk Küme (BBF) algoritması kullanılmaktadır [50]; bu yöntemde her zaman en yakın komşu bulunmamakta ancak arama süresini belirli bir aralıkta tutmak adına bu yöntem büyük veri tabanlarında kullanılmaktadır.

4.1 Enyakın Komşu Araması (NNS)

Basit Enyakın Komşu Araması (NNS), bütün veri tabanındaki elemanları tek tek eldeki vektörle karşılaştırmaya dayalıdır. Veri tabanında kayıtlı bütün vektörlerin eldeki vektörle arasındaki Öklit uzaklığına bakılır ve Öklit uzaklığı en küçük olan vektör veri tabanında işaretlenir. Bu teknikte, işlem yükü çok yüksek olmamasına rağmen, arama zamanı karmaşıklığı, n karşılaştırılacak vektör sayısı olmak üzere, $O(n)$ olduğundan, veri tabanı büyüdükçe, karşılaştırılacak eleman sayısı birlikte aynı oranda işlem süresi de artar, ancak her zaman tam doğru sonuca erişilir.

Sürenin kısıtlı olduğu ve veri tabanının büyüdüğü durumlarda, bu teknik verimsiz hale gelmekte; bunun yerine daha karmaşık yapıları olan, KD ağacı algoritması veya ondan türetilmiş yöntemler kullanılır.

4.2 KD Ağacı Algoritmasıyla Arama

KD ağacı (KDT) algoritması K boyutlu sonlu sayıda noktanın saklanması için oluşturulan bir veri yapısıdır; KD ağacı bir ikili ağaç yapısıdır. İkili ağaç oluşturmaya benzer bir şekilde K boyutlu ağaç da oluşturulmaktadır. KD ağacını oluştururken rastgele bir boyut seçilip ağaç oluşturulabilir, ancak bu çoğu zaman dengesiz dağılımlara neden olur [47].

Dengeli bir KD ağacı oluşturmak için öncelikle veri kümesinde K boyuttan her bir boyutun en küçük ve en büyük elemanlarına bakılıp, hangi boyutta en büyük farkın olduğuna bakılır ve en fazla farka sahip olan boyutta ayırma yapılır. Yine aynı boyuttaki sayıların ortadeğeri bulunur ve (Şekil 4.16)'deki gibi orta değerden küçük olanlar sol alt dalda, büyük olanlar sağ alt dalda kalacak şekilde ağaç oluşturulur. KD ağacını tamamlamak için işlem tekrarlanarak, hiç eleman kalmayana kadar devam edilir [49].

Örneğimizi daha da açıklayacak olursak, $K = 3$ boyutlu 15 vektörlü bir veri kümesi var. Bu kümeyi KD ağacına dengeli bir şekilde dağıtmak için, tüm elemanları tüm boyutlarında en küçük ve en büyük değerlere bakıyoruz. Burada 1.Boyut'ta en küçük değer 1, en büyük değer 9; 2.Boyut'ta en küçük değer 2, en büyük değer 8; 3.Boyut'ta en küçük değer 2, en büyük değer 9, en büyük değerle en küçük değer arasındaki farkın en fazla olduğu boyut 1.Boyut olduğundan birinci boyutta ayırma yapıyor, birinci

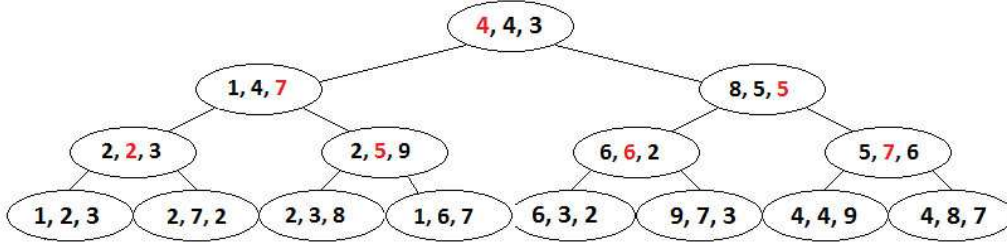
boyuta göre sıralama yaptığımızda orta değerimizden küçük olanlar sol dala, büyük olanlar sağ dala yazılıyor. Sonra soldaki dala baktığımızda 1.*Boyut*'ta enküçük değer 1, enbüyük değer 2; 2.*Boyut*'ta enküçük değer 2, enbüyük değer 7; 3.*Boyut*'ta enküçük değer 2, enbüyük değer 9, enbüyük değerle enküçük değer arasındaki farkın en fazla olduğu boyut 3.*Boyut* olduğundan üçüncü boyutta sıralama yapıp, orta değerden küçükler solda, büyükler sağda kalacak şekilde işlem tekrarlanır. 15 elemanın tamamı bir dalda kalacak şekilde işlemler tekrar edilir ve bütün elemanlar bittiğinde işlem sonlandırılır.

1. Boyut	2. Boyut	3. Boyut
1	2	3
1	4	7
1	6	7
2	2	3
2	3	8
2	5	9
2	7	2
4	4	3
4	4	9
4	8	7
5	7	6
6	3	2
6	6	2
8	5	5
9	7	3

Çizelge 4.2. KD Ağacı Örnek Veri Kümesi

Veri tabanı için KD ağacı oluşturulduktan sonra KD ağacı yapısında arama işlemine geçilir. Bu yöntem kullanılacağı zaman bütün veri tabanımızı oluşturacak bütün tanımlayıcı vektörler oluşturulur ve KD ağacı yapısı şekliyle veri tabanında saklanır; bu yüzden KD ağacı oluşturma süresi işlem başarımı açısından önemli bir etken oluşturmamaktadır.

KD ağacı içerisinde arama işleminde, öncelikle en yakın komşusu aramak istenilen



Şekil 4.16. KD Ağacı Yapısı, $K = 3$

tanımlayıcı vektör KD ağacında en tepedeki elemandan başlayarak (Şekil 4.16) alt dallara ayrılan boyutlarda tanımlayıcı vektörü için gerekli dala doğru taşınır; baktığımız boyutta tanımlayıcı vektörün değeri orta değerden küçükse sol alt dala, büyükse sağ alt dala yönlendirilir. En alttaki dala gidene kadar tanımlayıcı vektörle daldaki vektörün Öklit uzaklığına bakılır, bu vektör ve uzaklığı geçici olarak saklanır, en alt dala ulaşına kadar bu işlem tekrarlanır. Daha sonra en yakın vektörün bulunduğu dala gidilir, bu dalın ters tarafındaki dalda daha yakın olabilecek vektör olup olmadığına bakılır; eğer varsa bu dal ve alt dallarında arama devam ettirilir ve en yakın vektör için işlem tekrarlanır, eğer yoksa en yakın vektör kaydedilir ve işlem sonlandırılır.

KD ağacında en yakın araması yapılırken, eğer KD ağacı orta noktaları düzgün seçilmişse, genellikle ilk aşamada bulunan en yakın dalın bir üst dalında veya ona yakın birkaç dalda arama yapılır ve en yakın araması başarıyla sonlanır. Eğer dengesiz bir KD ağacı seçildiyse, bütün dallar aranacak ve KD ağacından bir fayda sağlanmayacaktır.

KD ağacı algoritmasının arama süresi karmaşıklığı vektör boyutuyla ilintilidir, $O(n^{1-1/k} + m)$; burada k vektörün boyu, m yapılan arama sayısıdır. Daha önce de anlatıldığı gibi KD ağacı algoritması vektör boyutu çok büyük olduğu zaman en yakın komşu eşleştirmede arama süresinin artmasına sebep olmaktadır [48] [49] [51], yapılan deneylerde de KD ağacı algoritmasıyla aramanın verimsiz olduğunu görülmüş, bu yüzden bu yöntem karşılaştırmaya alınmamıştır.

4.3 Eniyi İlk Küme (BBF) Algoritmasıyla Arama

Eniyi İlk Küme (BBF) KD ağacıyla arama algoritmasının bir türevidir [50] [52]. Büyük veri tabanlarında KD ağacı algoritması bir aşamaya kadar iyi sonuçlar vermekte, ancak belirli bir aşamadan sonra, vektör boyutu 10'u aştığı durumlarda, hız açısından kötü sonuçlar alınmaktadır.

Veri tabanı büyüdükçe Basit Enyakın Komşu Araması çok fazla zaman almaktadır [52]. Zamanın önemli olduğu durumlarda Eniyi İlk Küme algoritmasıyla arama yapılması verimli olmaktadır. Özellikle veri tabanı büyüklüğü 10^4 elemanı ve vektör boyutu 10'u aştığı durumlarda, belirli bir süre zarfında eşleştirme yapmak önem kazanmaktadır, ancak bu algoritmayla enyakın komşu araması ortalama bir hatayla yapılmaktadır. Kazanılan süreye bakıldığında zaman büyük veri tabanlarında bu hata önemsenmeyecek kadar küçük kalmaktadır [50].

Daha önce yapılan çalışmalarda, KD ağacı algoritması üzerinde farklı sayıda deneyler yapılmıştır. Bu çalışmalarda ağacın aranacak dal sayısı 200 ile sınırlandırılmıştır, ve bu miktarın en iyi sonucu verdiği belirlenmiştir. Bunun dışında yine ilk çalışmada bu sayı 480 aramayla karşılaştırılmış, doğru eşleşme kaybının çok azalmadığı görülmüş, arama zamanının ise 2 kattan daha fazla arttığı görülmüştür. Arama sayısının 200 ile sınırlandırıldığı yöntem bu yüzden Eniyi İlk Küme (BBF) araması adı verilmiştir [50].Yine daha önceki çalışmada [11], 10^5 tanımlayıcı vektöre sahip bir veri tabanında eşleme yapılmak istendiğinde Basit Enyakın Komşu aramasına göre hız olarak 2 kattan daha fazla bir iyileştirme için sadece %5 doğru eşleşme kaybı olduğu gösterilmektedir.

4.4 Yöntemlerin Karşılaştırılması ve Hızlandırma

SIFT yönteminde oluşturulan tanımlayıcı vektörlerin boyutu 128 olduğundan, ilk bakışta Eniyi İlk Küme algoritmasıyla eşleme yapılması en verimli yöntem olarak görülmektedir. Yapılan bir çok çalışmada da, diğer yöntemler göz ardı edilmiş, Eniyi İlk Küme algoritmasıyla, KD ağacıyla arama algoritmasına başka sınırlamalar getirerek karşılaştırmalar yapılmış, [50] [51] ancak veri tabanının küçük olabileceği durumlar incelenmemiş, bu durumlarda hız başarımının nasıl olacağı bilinmemektedir.

Yapılan denemelerde eşleme performansı veri tabanında tanımlanan görüntü sayı-

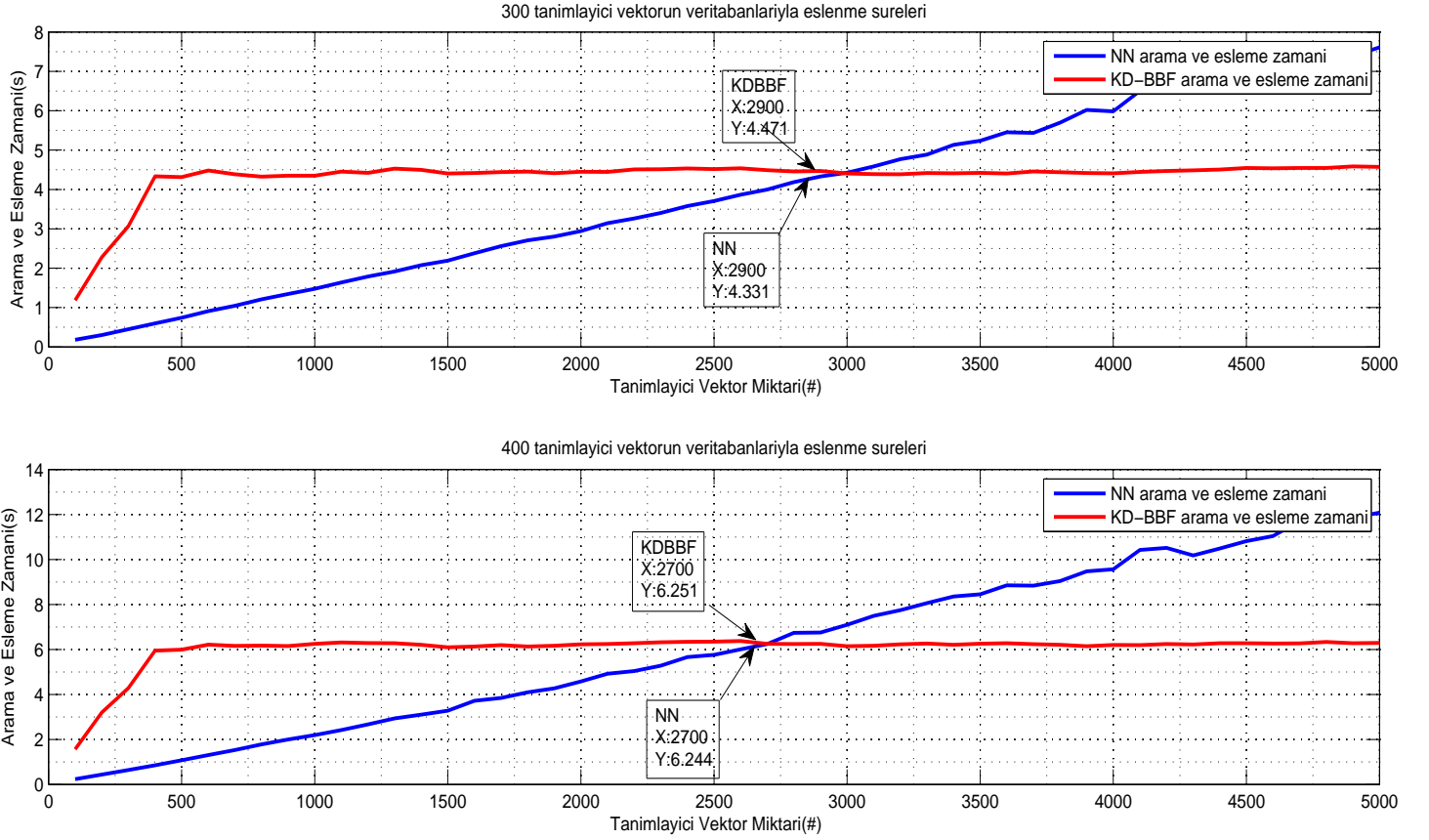
sayıyla, dolaylı olarak tanımlayıcı vektör sayısı ile belirli bir miktara kadar birebir ilintili olduğu görülmüştür. Eniyi İlk Küme aramasıyla eşleme performansı vektör sayısı belli bir miktarı aşana kadar verimsiz kalmakta, ancak belirli bir sayıdan sonra arama süresi sabit kalmakta ve üstünlük sağlamakta, belirli bir veri tabanı büyüklüğüne ulaşıncaya kadar Basit Enyakın Komşu aramasına göre yavaş kalmaktadır. Bunun nedeni olarak KD ağacı üzerinde yapılan işlemlerin sabit bir işlem süresi alması gösterilebilir.

İncelenen çalışmalarda, SIFT yönteminin kullanıldığı ve veri tabanının küçük olduğu durumlarla ilgili bir denemeye rastlanmamıştır. Yine incelenen çalışmalarda eniyi ilk küme algoritmasıyla KD ağacı algoritması arasında karşılaştırma yapılmış, ancak yapılan karşılaştırmalar zaman karşılaştırması olmayıp, eşleme başarımı karşılaştırması olmuştur. Hız başarımı açısından herhangi bir karşılaştırma yapılmamış, sadece ortalama işlem süresi verilmiştir. Basit enyakın komşu araması yöntemiyle zaman üstünlüğü elde edildiğine değinildiği görülmemiştir.

Yaklaşık 640x480 çözünürlükteki standard bir görüntüden yaklaşık 300 – 400 civarında tanımlayıcı vektör çıkarılmaktadır. Bir nesnenin görüntüsünden de ortalama 200 – 250 civarında tanımlayıcı vektör oluşturulmaktadır. Bir görüntüden çıkarılan 300 ve 400 tanımlayıcı vektörle yapılan benzetimlerde (Şekil 4.17), Basit Enyakın Komşu Aramasıyla (NNS) Eniyi İlk Küme (BBF) eşlemesinin zaman başarımları karşılaştırılmıştır. Veri tabanı büyüklüğü yaklaşık 2700 elemandan az olduğu durumlarda basit enyakın komşu aramasıyla eşlemede, eniyi ilk küme aramasıyla eşlemeden daha iyi zaman sonuçları elde edilmiştir. Görüntüden elde edilen tanımlayıcı vektör sayısı azaldıkça Basit Enyakın Komşu aramasının daha büyük veri tabanlarında zaman başarımı üstünlüğü sonucu çıkarılmıştır.

Analitik olarak incelendiğinde, kullandığımız vektör boyu için Basit Enyakın Komşu Araması yönteminin arama zamanı karmaşıklığı $O(n)$, Eniyi İlk Küme Araması yönteminin arama zamanı karmaşıklığı $O(n^{1-1/128} + 200)$ olarak görülmektedir. Buradaki 128 kullandığımız vektörlerin boyu, 200 ise arama sayısıdır. Bu iki yöntemin analitik hesaplamasında veri tabanındaki vektör sayısı 3265 olana kadar Basit Enyakın Komşu Aramasının avantajlı olması beklenmektedir, bu deneysel sonuçlarla benzerlik gösteren bir sonuçtur, aradaki fark da Eniyi İlk Küme araması algoritmasının her zaman 200 arama yapmayıp, bazı durumlarda 200'den daha az aramayla sonuca

ulaşabilmesi olarak görülmektedir.

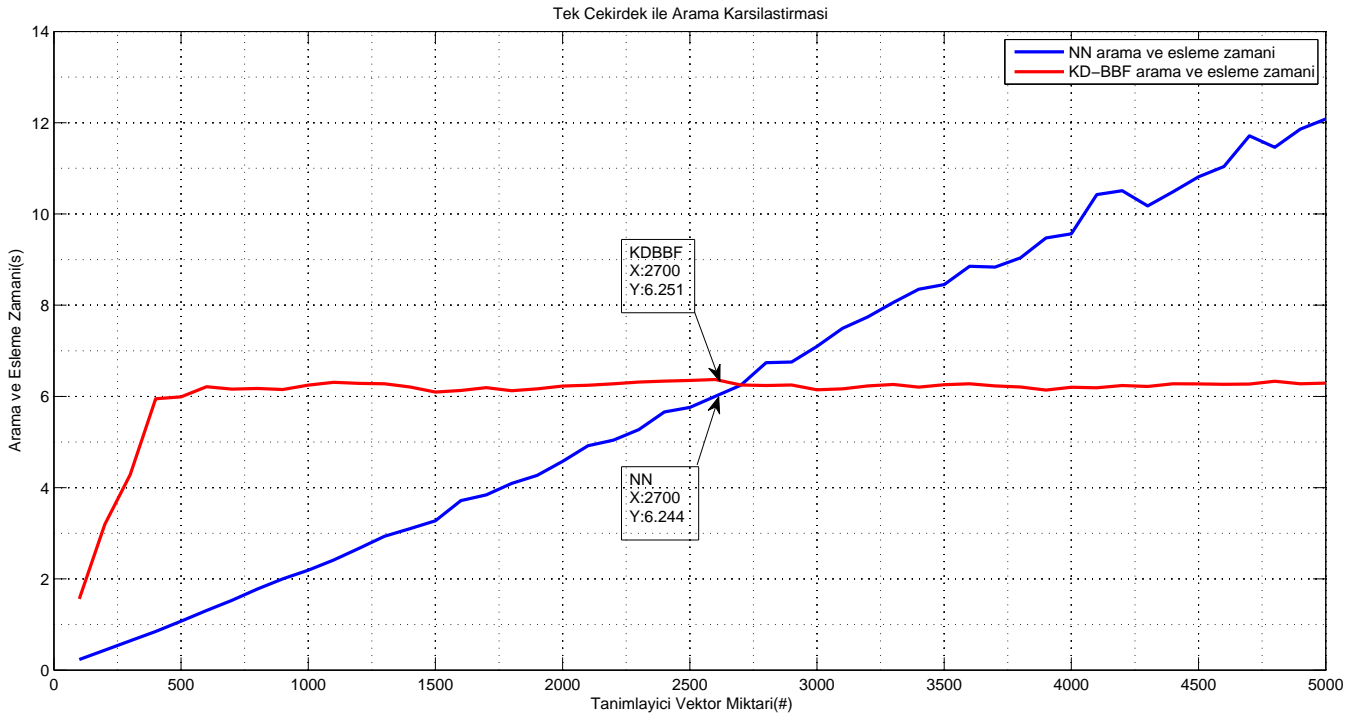


Şekil 4.17. 300 ve 400 Tanımlayıcı Vektörlü Arama ve Eşleme İşleminde Basit Enyakin Komşu ve Eniyi İlk Küme Araması Karşılaştırması

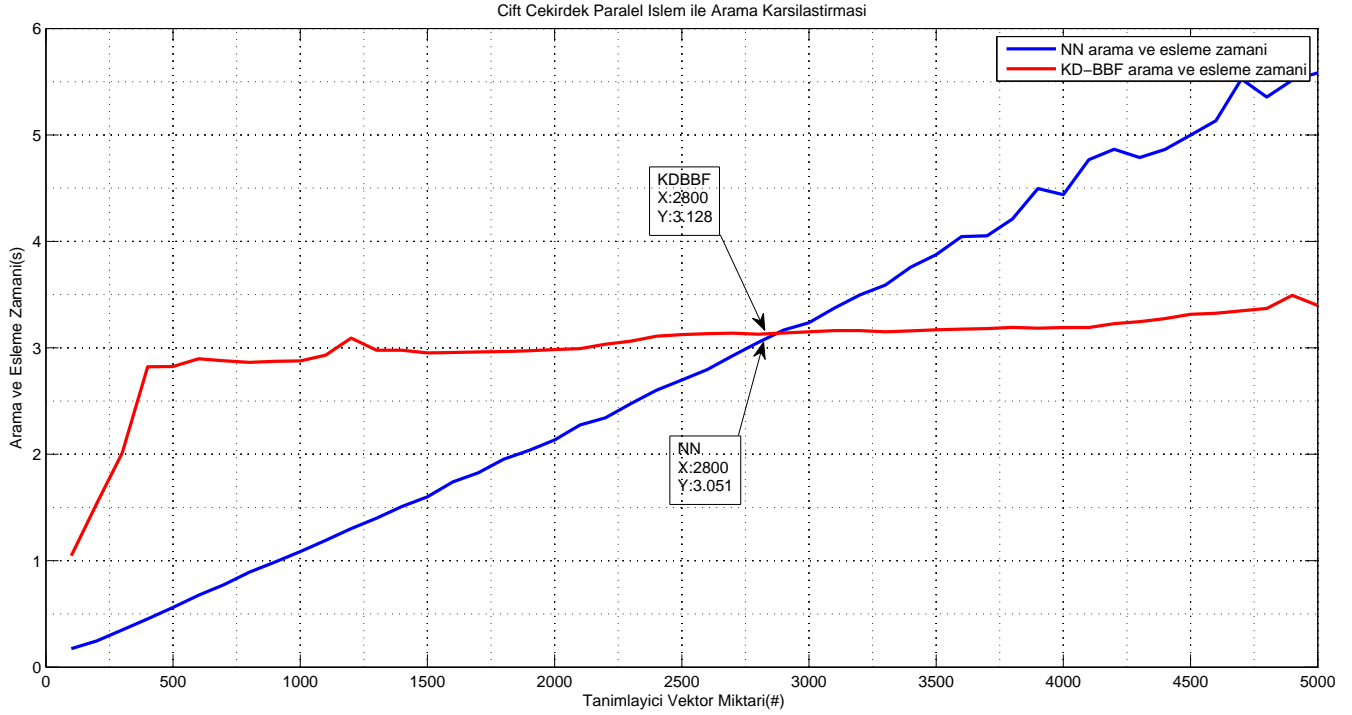
Yapılan benzetimlerde, Basit Enyakin Komşu Araması (NNS) ve Eniyi İlk Küme (BBF) aramalarında paralel işlemci kullanımında başarımler iyileşmeleri olduğu görülmüştür. Paralel işlem kullanılan işlemci sayısı ile aynı oranda arama ve eşleme sürelerini düşürmektedir. Ayrıca Basit Enyakin Komşu Aramasının (NNS), Eniyi İlk Küme (BBF) aramasına göre üstünlüğü de artmaktadır. Tek işlemcili, bir görüntüden çıkarılan 400 tanımlayıcı vektörlü benzetimde, Basit Enyakin Komşu aramasıyla yapılan eşlemede veri tabanı büyüklüğü 2700 elemana çıkarılana kadar Eniyi İlk Küme eşlemesine üstünlüğü sürmekte (Şekil 4.18), çekirdek sayısı 2'ye çıktığında veri tabanı büyüklüğü 2800 tanımlayıcı vektör oluncaya kadar Basit Enyakin Komşu araması üstünlük sağlamaya devam etmektedir (Şekil 4.19). Yine 3 ve 4 çekirdekli benzetimlerde de Basit Enyakin Komşu aramasının Eniyi İlk Küme aramasına üstünlüğü veri tabanı büyüklüğü 2900, 3000 tanımlayıcı vektöre kadar artarak devam

etmektedir (Şekil 4.20), (Şekil 4.21). Tüm paralel işlemlerde her iki yöntemin de zaman başarımı artmasının yanında, daha önce çalışmalarda rastlanmamış olan Enyakın Komşu Aramasının belirli veri tabanı büyüklüklerine kadar zaman başarımı üstünlüğü gösterilmiştir.

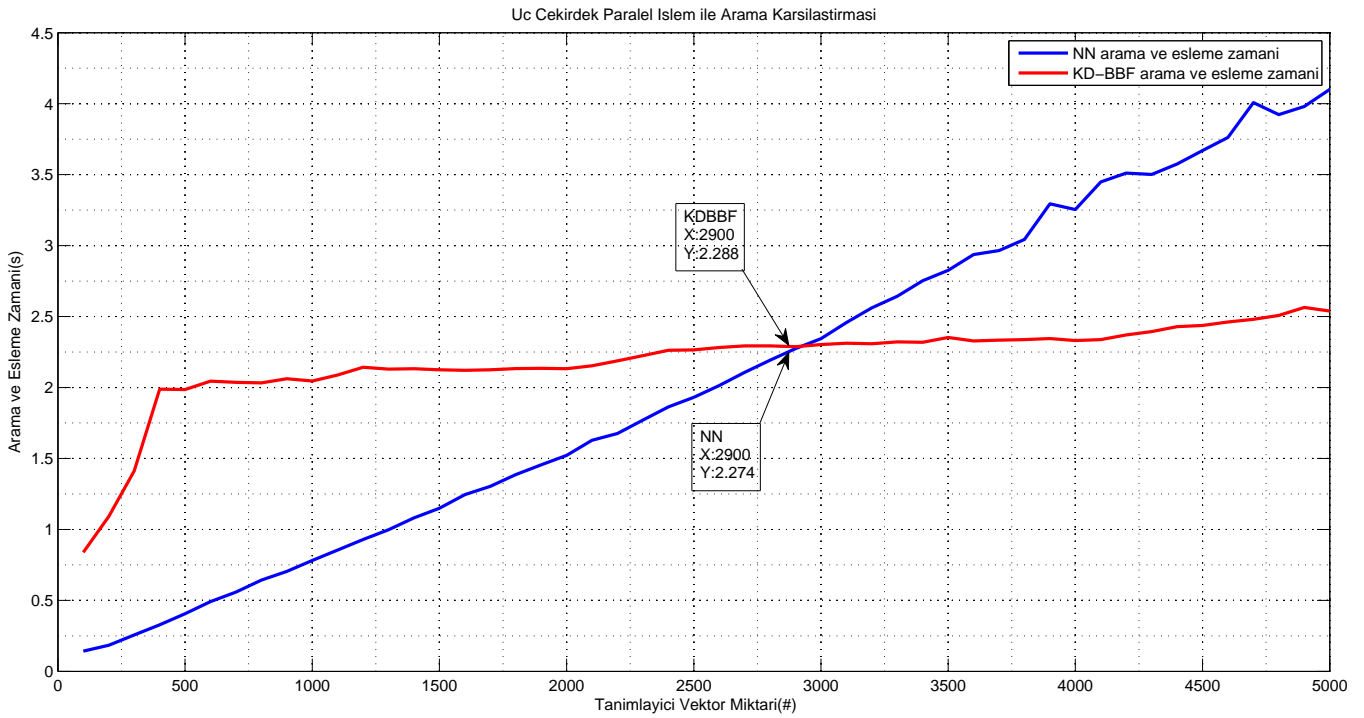
Yine aynı şekilde, (Şekil 4.23)'de görüldüğü üzere paralel işlemcilerin kullanımında işlemci sayısı belirli bir vektör büyüklüğüne ulaşıldığında daha da belirgin hale gelmekte, veri tabanı büyüdükçe Basit Enyakın Komşu Aramasında paralel işlemci kullanımının avantajı artmaktadır. Bununla birlikte (??)'te görüldüğü üzere paralel işlemci kullanıldığında veri tabanı büyüdükçe Eniyi İlk Küme Aramasının zaman iyileşmesi aynı oranda olmamaktadır.



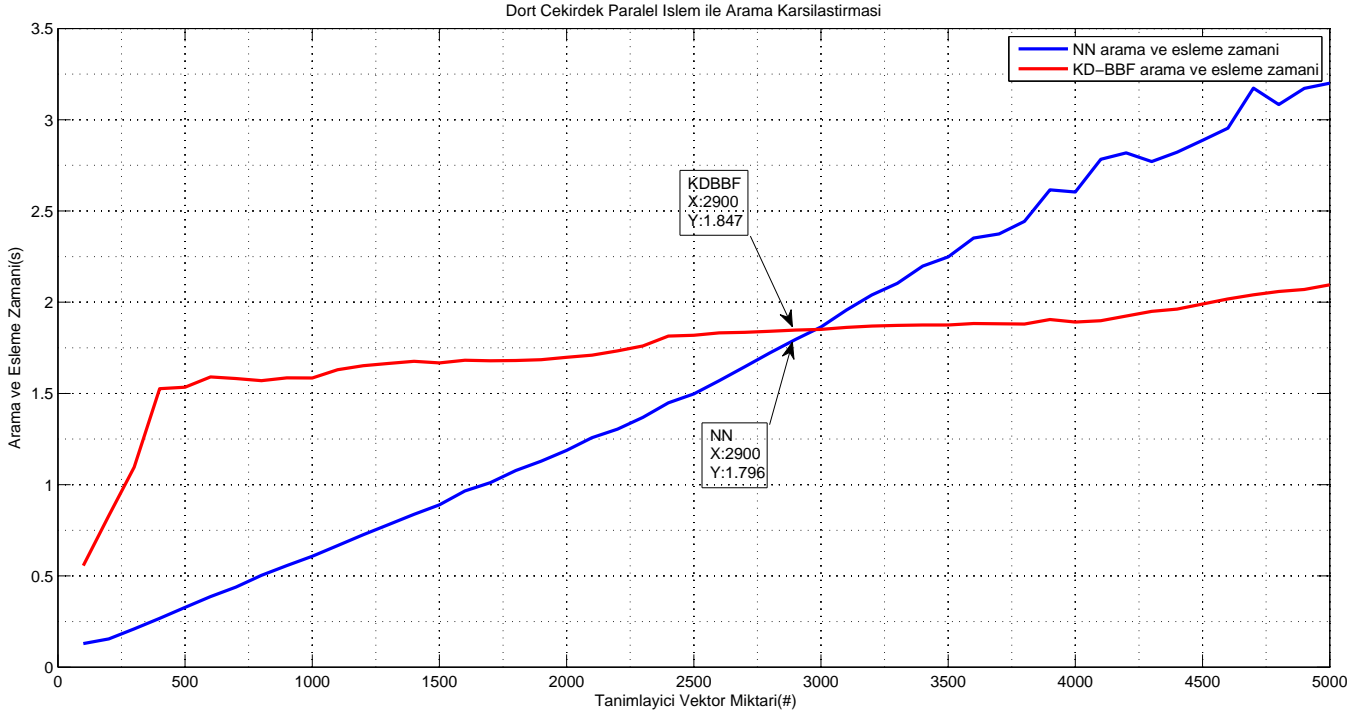
Şekil 4.18. Tek İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması



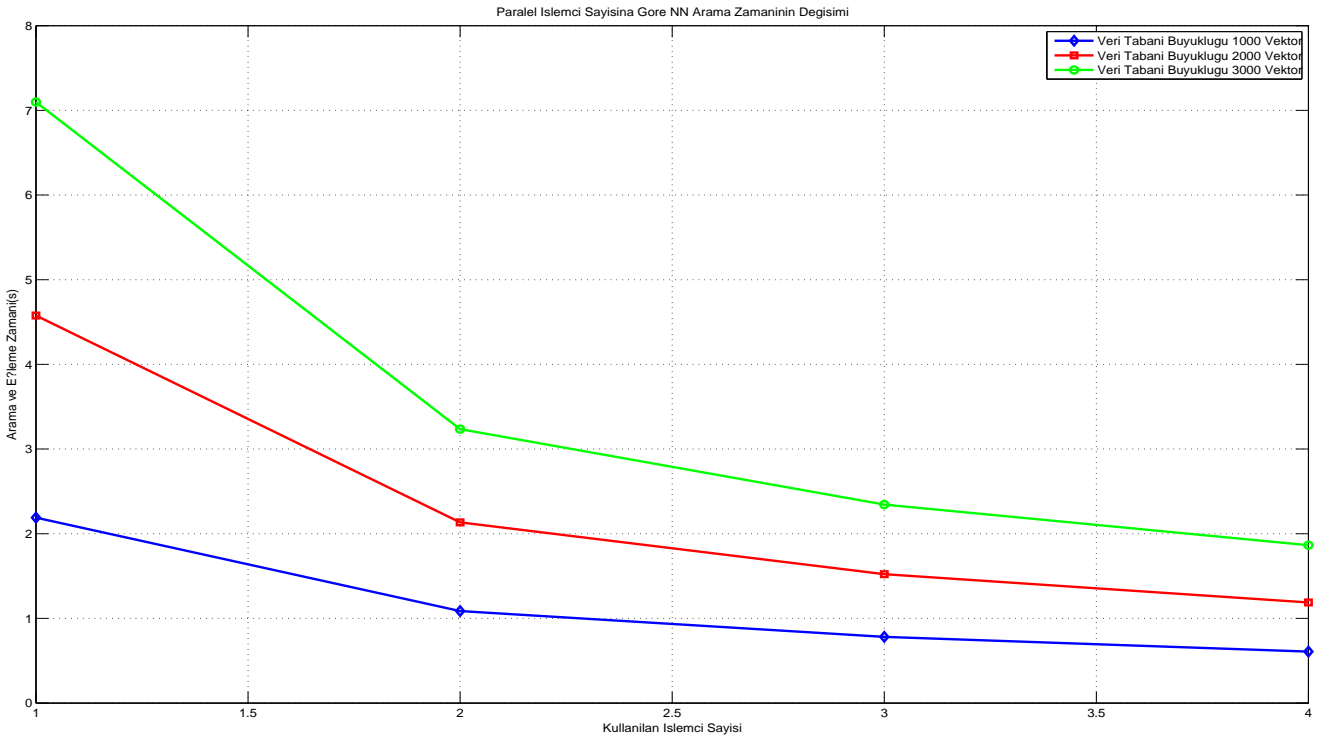
Şekil 4.19. Paralel Çift İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması



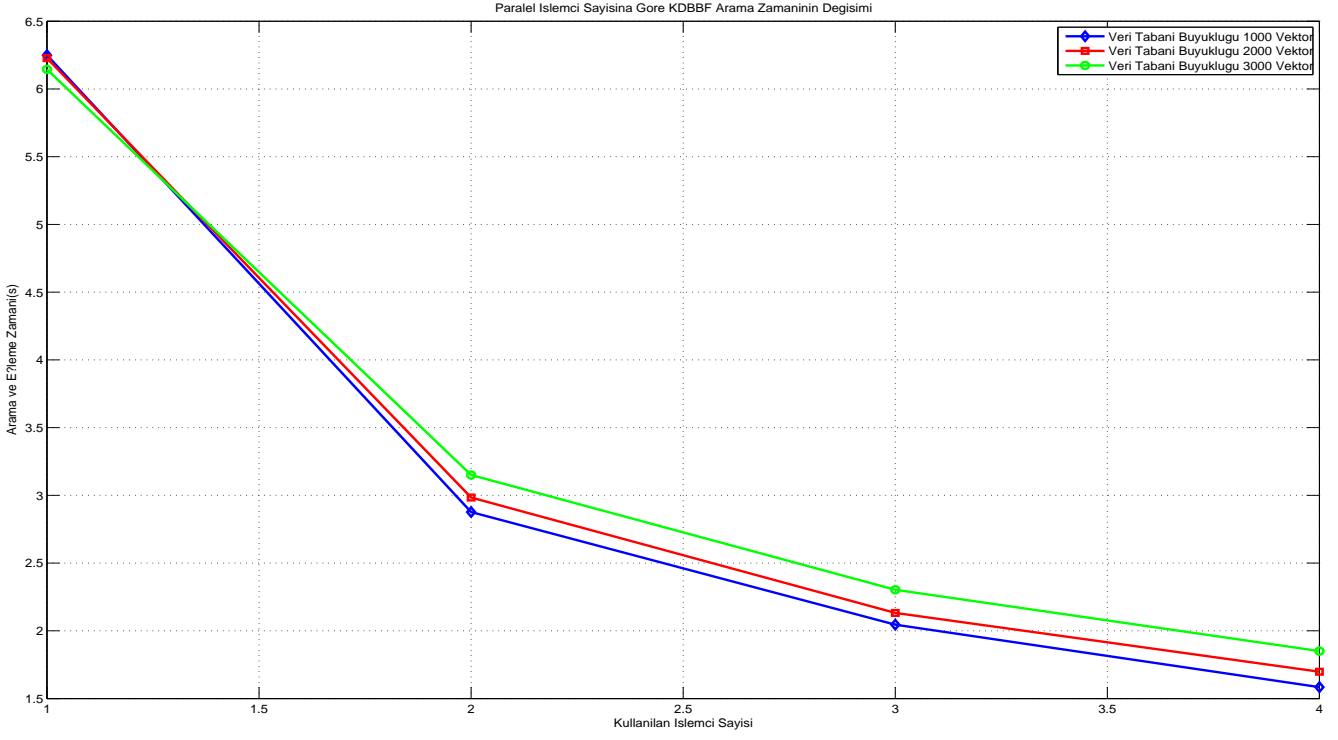
Şekil 4.20. Paralel Üç İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması



Şekil 4.21. Paralel Dört İşlemciyle 400 Tanımlayıcı Vektörlü Arama ve Eşleştirme İşleminde Basit Enyakın Komşu ve Eniyi İlk Küme Araması Zaman Karşılaştırması



Şekil 4.22. Farklı Sayıda Paralel İşlemcilerin Kullanımının Basit Enyakın Komşu Aramasında, Çeşitli Veri Tabanı Büyüklüklerinde Zaman Başarımına Etkisi



Şekil 4.23. Farklı Sayıda Paralel İşlemcilerin Kullanımının Eniyi İlk Küme Aramasında, Çeşitli Veri Tabanı Büyüklüklerinde Zaman Başarımına Etkisi

5. SONUÇ

Nesne tanımadaki eşleşme süresi önemli bir zaman almaktadır. Nesne tanıma çalışmalarında genel olarak eşleşme başarımı dikkate alınmış, hız başarımına çok önem verilmemiştir. Ancak özellikle hedef takibi gibi gerçek zamanlı askeri uygulamalarda, yüz tanıma ve tarama gibi güvenlik uygulamalarında hızın önemi artmaktadır.

Bu tezde Ölçek Değişimsiz Özellik Dönüşümü (SIFT) yöntemiyle görüntüler ve nesnelere eşleştirilmeye çalışılmış, bu yöntemde eşleştirme esnasında harcanan zamanı iyileştirmek için yöntemler karşılaştırılmıştır. Zaman iyileştirmeleri sadece tezde uygulanan yöntem için değil, vektörel tanımlayıcı oluşturulan ve tanımlayıcı vektörlerin eşleşmesi için Öklit uzaklığına bakılan SURF, GLOH gibi diğer yöntemler için de bir iyileştirme olmuştur.

Daha önce zaman karşılaştırması yapılmamış olan Basit Enyakın Komşu Araması (NNS) ve Eniyi İlk Küme (BBF) yöntemleri karşılaştırılmış, ve bu yöntemlerden Basit Enyakın Komşu Aramasında vektör büyüklüğüne çok bağımlı olmadan, veri tabanındaki vektör sayısı belli bir sayıya ulaşana kadar daha iyi sonuç verdiği görülmüştür. Bu sonuçlar, yüz veya retina tanıma gibi bir kaç kişinin tanıtıldığı güvenlik sistemlerinde, bir kaç nesneyi tanıyıp kendiliğinden ayrıştırma yapma amaçlı endüstriyel bantlardaki robot kol sistemlerinde veya belirli bir hedefin takip edilmek istendiği askeri sistemlerde kullanılabilir, gerçek zamanlı veya gerçek zamanlıya yakın başarımlar sağlanabilecektir.

Bu tezde yapılan karşılaştırmalarda MATLAB ortamında paralel işlemci kullanımıyla benzetimler yapılmış, bu kullanımla zaman açısından beklendiği gibi iyileştirmeler gerçekleştirilmiştir. Eniyi küme araması açısından iyileştirmeler olsa da, paralel işlemci kullanımı basit enyakın komşu aramasında oransal olarak daha etkili bir iyileştirme sonucu ortaya çıkarmıştır. Bunun sonucunda paralel işlemci kullanımında, basit işlemlerin karmaşık işlemlerden daha fazla gelişme gösterdiği anlaşılmaktadır. Bu da paralel işlemci kullanımının basit enyakın komşu kullanımını, yine veri tabanı büyüklüğüne bakılarak daha çok yerde kullanılabilir hale getirilebileceğini göstermiştir. Basit enyakın komşu araması fazla işlem gücü gerektirmediğinden, çok güçlü olmayan işlemci çekirdekleriyle de rahatlıkla arama yapılabilir; bu yöntem çok çekirdekli sistemlerde daha verimli uygulanabilecektir.

Bu tezden çıkarılan sonuçlar, bir çok yöntemde kullanılan nesne ve görüntüden çıkarılan tanımlayıcı vektörlerin eşleştirme işleminin belirli koşullarda hızlandırılması için bir katkı sağlayacaktır. Bu sonuçlarla, ilerideki çalışmalarımızda, gömülü sistem ve FPGA yapılarında birden fazla işlemci çekirdeği kullanılarak; sistem gereksinimleri, benzetimlerde bulunan ve gösterilen sınır değerlerdeki veri tabanlarından daha az sayıda tanımlayıcı vektöre sahip olduğunda, gerçek zamanlı veya gerçek zamanlıya yakın sonuçlar elde edilebilecektir. Ayrıca gömülü sistemler, sınırlı miktarda hafızaya sahip olduğundan, veri tabanı büyüklüğü belirli bir kısıtı aşamayacağından, Basit En-yakın Komşu Aramasıyla sağladığımız hız başarımı gömülü sistem çalışmalarında büyük bir artı olarak kazanılacaktır.

KAYNAKLAR DİZİNİ

- [1] K. Okada, M. Kojima, S. Tokutsu, T. Maki, Y. Mori, M. Inaba, "Multi-Cue 3D Object Recognition in Knowledge-Based Vision-Guided Humanoid Robot System", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3217–3222, November 2007.
- [2] E.G. Rajan, K. Jayasankar and S. Kirankumar, "Object Recognition and Tracking in a Multiple Target Environment", IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, pp. 2390–2395, October 1994.
- [3] Y. Meng, "Implementing Scale Invariant Feature Transform Method", School of Computer Sciences University of St. Andrews Poster Event Report, 2006.
- [4] S. Prabhakar, S. Pankanti, A.K. Jain, "Biometric Recognition: Security and Privacy Concerns", IEEE Security and Privacy, vol.1, issue 2, pp.33-42, March 2003.
- [5] A. de la Escalera, J.M. Armingol and M. Mata, "Traffic Sign Recognition and Analysis for Intelligent Vehicles", Image and Vision Computing, vol. 21, issue 3, pp.247–258, March 2003.
- [6] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, issue 5, pp.564–577, May 2003.
- [7] Y. Li, I. Atmosukarto, M. Kobashi, J. Yeun and L.G. Shapiro, "Object and Event Recognition for Aerial Surveillance, Research Paper by National Science Foundation by ARDA, 2005.
- [8] G.L. Foresti, "Object Recognition and Tracking for Remote Video Surveillance", IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, issue 7, pp. 1045–1062, October 1999.
- [9] J. Damon, "Properties of Ridges and Cores in Two-Dimensional Images", Journal of Mathematical Imaging and Vision 10 (2), pp. 163–174, March 1999.
- [10] J. Miller, "Relative Critical Sets in and Applications to Image Analysis", Ph.D. Dissertation, University of North Carolina, 1998.
- [11] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
- [12] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, 27, pp 1615–1630, 2005.
- [13] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893, June 2005.

- [14] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [15] J. Lee, Y. Kim, C. Park and J. Paik, "Robust Feature Detection Using 2D Wavelet Transform under Low Light Environment", *International Conference on Intelligent Computing, LNCIS*, vol. 345, pp. 1042–1050, 2006.
- [16] I. Skrypnik and D.G. Lowe, "Scene Modelling, Recognition and Tracking with Invariant Image Features", *Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp.110–119, 2003.
- [17] C.Y. Lin and E. Setiawan, "Object Orientation Recognition Based on SIFT and SVM by Using Stereo Camera", *IEEE International Conference on Robotics and Biomimetics*, pp. 371–376, February 2008.
- [18] A.Suga, K. Fukuda, T. Takiguchi and Y. Ariki, "Object Recognition and Segmentation Using SIFT and Graph Cuts", *International Conference on Pattern Recognition*, pp. 1–4, 2008.
- [19] F. Alhwarin, C. Wang, D. Risti-Durrant, A. Gräser, "Improved SIFT-Features Matching for Object Recognition", *BCS International Academic Conference on Visions of Computer Science*, pp.179–190, 2008.
- [20] A. Auclair, L.D. Cohen and N. Vincent, "How to Use SIFT Vectors to Analyze an Image with Database Templates", *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics*, Springer-Verlag, pp.224-236, 2008.
- [21] E.N. Mortensen, D. Hongli and L. Shapiro, "A SIFT Descriptor with Global Context", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 184-190, 2005.
- [22] D.G. Lowe, "Object Recognition from Local Scale-Invariant Features", *Proceedings of the International Conference on Computer Vision*, 2, pp. 1150–1157, 1999.
- [23] K. Mikolajczyk, A. Zisserman and C. Schmid, "Shape Recognition with Edge-Based Features", *British Machine Vision Conference*, Norwich, UK, pp. 779–788, 2003.
- [24] L.G.Roberts, "Machine Perception Of Three-Dimensional Solids", Ph.D. Thesis Report Massachusetts Institute of Technology Lincoln Laboratory, May 1965.
- [25] N.Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, issue 1, pp. 62–66, 1979.
- [26] J.M.S.Prewitt, "Object Enhancement and Extraction", *New York Academic*, pp. 75–149, 1970.
- [27] D.Kroon, "Numerical Optimization of Kernel Based Image Derivatives", *Short Paper University Twente*, 2009.

- [28] J.Canny, "A Computational Approach To Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):pp. 679–698, 1986.
- [29] H.Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover", Tech. Report, Robotics Institute, Carnegie Mellon University and Doctoral Dissertation, Stanford University, CMU-RI-TR-80-03, September 2001.
- [30] C.Harris and M.Stephens, "A Combined Corner and Edge Detector", Proceedings of the 4th Alvey Vision Conference, pp. 147–151, 1998.
- [31] J.Shi and C.Tomasi, "Good Features To Track", IEEE Conference on Computer Vision and Pattern Recognition CVPR 94, pp. 593–600, 1994.
- [32] W.Förstner and E.Gülch, "A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features", ISPRS Intercommission Workshop, Interlaken, June 1987.
- [33] H. Wang and M. Brady, "Real-time corner detection algorithm for motion estimation", Image and Vision Computing 13 (9), pp. 695–703, 1995.
- [34] S. M. Smith and J. M. Brady, "SUSAN - A New Approach to Low Level Image Processing", International Journal of Computer Vision 23 (1), pp. 45–78, May 1997.
- [35] M. Trajkovic and M. Hedley, "Fast corner detection", Image and Vision Computing 16 (2), pp. 75–87, 1998.
- [36] T. Lindeberg, "Detecting Salient Blob-Like Image Structures and Their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention", International Journal of Computer Vision 11 (3), pp 283–318, 1993.
- [37] T. Lindeberg, "Feature Detection with Automatic Scale Selection", International Journal of Computer Vision 30 (2), pp. 77–116, 1998.
- [38] T. Lindeberg, "Scale-space Theory: A Basic Tool for Analysing Structures at Different Scales", Journal of Applied Statistics 21 (2), pp. 224–270, 1994.
- [39] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors", International Journal of Computer Vision 60 (1), pp. 63–86, 2004.
- [40] T.Lindeberg, "Edge detection and ridge detection with automatic scale selection", International Journal of Computer Vision 30 (2), pp.117–154, 1998.
- [41] P.V.C. Hough, "Machine Analysis of Bubble Chamber Pictures", Proceedings of International Conference on High Energy Accelerators and Instrumentation, 1959.
- [42] R.O. Duda and P.E. Hart, "Use of The Hough Transformation To Detect Lines and Curves in Pictures", Community of ACM, vol.15, no.1, pp. 11–15, January 1972.
- [43] D.H. Ballard, "Generalizing the Houghtransform to Detect Arbitrary Shapes", Pattern Recognition, Elsevier, vol. 13, Issue 2, pp. 111–122, 1981.

- [44] M.Brown, D.G.Lowe, "Invariant Features from Interest Point Groups" British Machine Vision Conference, Cardiff, Wales, pp. 656-665, 2002.
- [45] D.Lowe, "Local Feature View Clustering for 3D Object Recognition", Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, vol. 1, pp.682-688, December 2001.
- [46] C.Schmid, and R.Mohr, "Local Grayvalue Invariants for Image Retrieval", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19(5), pp.530-534, 1997.
- [47] S.H. Yen, C.Y. Shih, T.K. Li and H.W. Cheng, "Applying Multiple KD-Trees in High Dimensional Nearest Neighbour Searching", International Journal of Circuits, Systems and Signal Processing, vol. 4, issue 4, pp.153–160, 2010.
- [48] J.R. Friedman, J.L. Bentley and R.A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time", ACM Transactions on Mathematical Software, vol.3, no.3, pp.209–226, September 1977.
- [49] A.W. Moore, "An Introductory Tutorial on KD Trees", PhD. Thesis, Technical Report No. 209 Computer Laboratory University of Cambridge, 1991.
- [50] J.S. Beis and D.G. Lowe, "Shape Indexing Using Approximate Nearest Neighbor Search in High Dimensional Spaces", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1000-1006, June 1997.
- [51] N. Kumar, L. Zhang and S. Naylor, "What is a Good Nearest Neighbor Algorithm for Finding Similar Patches Images?", National Defense, Science and Engineering Graduate Fellowship, Lecture Notes in Computer Science, vol. 5303, pp. 364–378, 2008.
- [52] J. Kybic and I. Vnucko, "Approximate Best Bin First K-D Tree All Nearest Neighbor Search with Incremental Updates", Research Reports of Center for Machine Perception Czech Technical University, Prague, no. 10, July 2010.
- [53] E.R. Davies, "Machine Vision: Theory, Algorithms, Practicalities", Academic Press, London, 2002.
- [54] D.D.A.P. Ariyawasa and T.A. Clarke, "High Speed Correspondence for Object Recognition and Tracking", International Society for Optics and Photonics, Vol. 3197, pp. 70–79, 1997.

ÖZGEÇMİŞ

Adı Soyadı : MEHMET HAKAN AKŞİT
Doğum Yeri : Ankara
Doğum Yılı : 26.02.1986
Medeni Hali : Bekar

Eğitim ve Akademik Durumu

Ortaokul-Lise 1997-2004 : Anadolu Lisesi, Ankara Atatürk
Lisans 2004-2009 : Hacettepe Üniversitesi
Elektrik ve Elektronik Mühendisliği Bölümü, ANKARA

Yabancı Dil : İngilizce

İş Tecrübesi

Ağustos 2009-... : Hacettepe Üniversitesi
Elektrik ve Elektronik Mühendisliği Bölümü, ANKARA
Araştırma Görevlisi