

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**İLGİ BÖLGELERİ VE YEREL TANIMLAYICILAR İLE
GENELLEŞTİRİLMİŞ HOUGH DÖNÜŞÜMÜ VE EN AZ
EYLEMSİZLİK EKSENİ TABANLI HİZALAMA
YAKLAŞIMI İLE TÜRK İŞARET DİLİ TANIMA SİSTEMİ**

Bilg. Müh. Yük. Lis. Oğuz ALTUN

**FBE Bilgisayar Mühendisliği Anabilim Dalında
Hazırlanan**

DOKTORA TEZİ

Tez Savunma Tarihi	: 20 Aralık 2010
Tez Danışmanı	: Yrd.Doç.Dr. Songül ALBAYRAK (YTÜ)
Jüri Üyeleri	: Prof.Dr. A. Coşkun SÖNMEZ (YTÜ) Doç.Dr. Cem ÜNSALAN (YÜ) Doç.Dr. Yusuf Sinan AKGÜL (GYTE) Yrd.Doç.Dr. Elif KARSLIGİL (YTÜ)

İSTANBUL, 2010

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
2. KONU İLE İLGİLİ ÖNCEKİ ÇALIŞMALAR.....	5
2.1 El İşareti, İşaret Dili ve Parmak Hecellemesi Tanıma Çalışmaları	5
2.2 İlgili Noktaları ve İlgili Bölgeleri	6
2.3 Ölçek Uzayı	7
2.4 İlgili Bölgesi Çıkarma Yöntemlerinin Nesne Sınıfı Tanıma Açısından Karşılaştırılması.....	7
2.5 İlgili Bölgelerinden Çıkarılan Tanımlayıcılar.....	9
2.6 Tanıma Amaçlı İlgili Bölgeleri ve Tanımlayıcıları	10
2.7 Video Üzerinde İlgili Bölgeleri İle Çalışmalar	10
2.8 Standart Hough Transformu	11
2.9 Genelleştirilmiş Hough Transformu	12
2.10 Genelleştirilmiş Hough Transformu İle İmgelerde Şekil Tanıma Çalışmaları.....	12
2.11 İlgili bölgeleri ve Genelleştirilmiş Hough Transformu	14
2.11.1 Hough Transform Modeli Oluşturma - Doğrulama.....	14
2.11.2 Gürültülü Değerler ile Baş Etme	15
3. EN AZ EYLEMSİZLİK EKSENİ TABANLI HIZLI HİZALAMA İLE TÜRK İŞARET DİLİ TANIMA SİSTEMİ.....	16
3.1 Anahtar Kare Çıkarımı	16
3.2 Renk Kullanarak Ten Belirleme	17
3.3 Hızlı Hizalama	17
3.4 Ek Nesne Özellikleri.....	19
3.5 Video Veritabanı.....	20
3.6 Sınıflama Karşılaştırması.....	20
4. GENELLEŞTİRİLMİŞ HOUGH TRANSFORMU, İLGİLİ BÖLGELERİ VE YEREL TANIMLAYICILAR İLE TÜRK İŞARET DİLİ PARMAK HECELEMESİ TANIMA SİSTEMİ.....	22

4.1	Genelleştirilmiş Hough Transformu ve İlgi Bölgeleri İle Nesne Tanımaya Ayrıntılı Bir Bakış	22
4.2	İlgi Bölgeleri ve Yerel Tanımlayıcılar.....	22
4.3	Ten Rengi Tabanlı İlgi Bölgesi Eliminasyonu	23
4.4	Hough Transformu.....	23
4.5	Oylama tabanlı sınıflama	27
4.6	Veritabanı, Protokol ve Optimizasyon	28
4.7	Yapılan İyileştirmeler	28
4.7.1	Oy Kullanan Veritabanı Eşleşmelerinin Sayısı	28
4.7.2	Hough Birikeci Kova Boyutu	28
4.7.3	Doğrulama Adımları	29
4.7.4	Tüm Optimizasyon Sonuçlarının Birleştirilmesi	29
4.8	Yanlış Sınıflanan Parmak Hecelemeleri	30
5.	YEREL İLGİ BÖLGELERİNİN KATI-OLMAYAN NESNE SINIFI TANIMA AÇISINDAN BİR KARŞILAŞTIRMASI.....	32
5.1	Karşılaştırılan İlgi Bölgesi Bulucuları	32
5.2	Ayırt Edicilik	35
5.3	ExpRand: Eksponensiyal Rastgele Bölge Üreticisi.....	37
5.4	Veritabanı ve Protokol	38
5.5	Başarı Oranı	38
5.6	Değerlendirme ve Sonuçlar	38
6.	SONUÇLAR VE ÖNERİLER.....	41
	KAYNAKLAR.....	44
	EKLER	49
	Ek 1 Tez Çalışması Kapsamında Yapılan Bilimsel Yayınlar	50
	Ek 2 Tez Çalışması Başlangıcında Yapılması Düşünülen Ancak Yapılmayan Çalışmalar	51
	Ek 3 Tez Çalışması Sırasında Yapılan Ancak Yayına Dönüşmeyen Çalışmalar ..	56
	ÖZGEÇMİŞ.....	75

SİMGE LİSTESİ

R	RGB renk uzayında piksel renginin R (kırmızı) bileşeni
G	RGB renk uzayında piksel renginin G (yeşil) bileşeni
B	RGB renk uzayında piksel renginin B (mavi) bileşeni
f	Video karesi indeksi
Y	YCrCb renk uzayında piksel renginin Y bileşeni
C_r	YCrCb renk uzayında piksel renginin Cr bileşeni
C_b	YCrCb renk uzayında piksel renginin Cb bileşeni
D^f	f ve $f + 1$ kareleri arasındaki uzaklık
d	İlgi bölgesi belirleyicisi başarısı
r	Rastgele bölge üreticisi başarısı
q	Ayırt edicilik

KISALTMA LİSTESİ

SHT	Standart Hough Transformu
SIFT	Scale Invariant Feature Transform
FGHT	Fast Generalized Hough Transform
PHT	Probabilistik Hough Transformu
RHT	Randomized Hough Transform
kNN	k Nearest Neighnor
SVM	Support Vector Machine
RBF	Radial Basis Function
MNB	Multinomial Naive Bayes
HarLap	Harris-Laplace
HarAff	Harris-Affine
HesLap	Hessian-Laplace
HesAff	Hessian-Affine
EdgeLap	Edge Laplace
IBR	Intensity Based Regions
MSER	Maximally Stable Extremal Regions
PCBR	Principal Curvature Based Regions
DoG	Difference of Gaussians
kAS	k Adjacent Segments
SURF	Speeded Up Robust Feature
FAST	Features from Accelerated Segment Test
ExpRand	Eksponensiyal Rastgele Bölge Üreticisi
NRNP	Nesne Referans Noktası Pozisyonu

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1	Türk alfabesinde bulunan 29 harfin parmak hecelemelerinden çıkarılan anahtar kareler.	2
Şekil 2.1	Bir ilgi noktası ve onu destekleyen ilgi bölgesi.	7
Şekil 2.2	(a) Doğru denklemi (2.1) içinde kullanılan d ve θ parametreleri. (b) Hough birikeci. Kaynak: (Shapiro ve Stockman, 2001).	12
Şekil 3.1	En az ikinci moment eksen ve yönelim açısı θ	18
Şekil 3.2	Hızlı hizalama işleminin aşamaları. (a) Orijinal kare. (b) Belirlenen ten bölgeleri. (c) Nesnenin içinde bulunduğu dikdörtgen. (d) Yönelimi yönelim açısı yardımı ile düzenlenmiş nesne. (e) Nesneyi içeren yeniden boyutlandırılmış sınırlayan kare.	19
Şekil 3.3	Metodumuzun başarısız olabildiği iki örnek: (a) Ğ (b) G (c) Ş (d) S.	21
Şekil 4.1	Hough Transformu eğitim ve tanıma aşamalarının özeti. NRNP nesne referans noktası pozisyonu anlamına gelmektedir.	22
Şekil 4.2	Ten rengi ile el bölgesi çıkarma işlemi sonucu.	23
Şekil 4.3	Genelleştirilmiş Hough Transformu, ilgi bölgeleri ve yerel tanımlayıcılar ile şekil tanımda eğitim aşaması.	24
Şekil 4.4	Genelleştirilmiş Hough Transformu, ilgi bölgeleri ve yerel tanımlayıcılar ile şekil tanımda tanım aşaması.	25
Şekil 4.5	Doğru bir eşlemenin sorgu ve eğitim imgeleri üzerinde gösterilen Hough Transformu bileşenleri.	27
Şekil 4.6	Yanlış bir eşleme üzerinde Hough Transformu bileşenleri.	27
Şekil 4.7	Karıştırılan sınıflardan örnekler.	30
Şekil 5.1	ExpRand tarafından üretilen 100 adet rastgele ölçek. Ölçekler grafiğe aktarılmadan önce dağılımın anlaşılabilir hale gelmesi için sıralanmıştır. Yatay eksen ölçek numarasını, dikey eksen ise ölçeği vermektedir.	37
Şekil 2.	Uzay Zaman Genelleştirilmiş Hough Transformunda bileşik özellik, bileşik özellik afin koordinat merkezi ve hareket nesnesinin merkezinin yeri.	51
Şekil 3.	Sürekli Uzay-Zaman Genelleştirilmiş Hough Transformu için bileşik birikeç.	52
Şekil 4	Bir T modeli ve üzerinde tanımlı parametreler. Kaynak: (Suetake vd., 2006).	52
Şekil 5	Genişletilmiş C-Tablosu. Kaynak: (Suetake vd., 2006).	53
Şekil 6	Suetake vd. (Suetake vd., 2006) nın genelleştirilmiş bulanık Hough Transformu için C_k alanı. Kaynak: (Suetake vd., 2006).	53
Şekil 7	Gürültülü bir imgede bozulmuş bir yıldız şeklinin bulunması. (a) Model imgesi (128 x 128 piksel). (b) Gürültülü bir hedef imgesi. (c) Kimura ve Watanabe (Kimura ve Watanabe, 1998) yaklaşımı sonucu. (d) Suetake vd. (Suetake vd., 2006) yaklaşımı sonucu. Kaynak: (Suetake vd., 2006).	54
Şekil 8	(a) Doğru denklemi $d = -r\sin\theta + c\cos\theta$ da kullanılan d ve θ parametreleri. (b) 256 x 256 boyutlarındaki imgelerde doğru bulmak için kullanılan birikeç örneği. Kaynak: (Shapiro ve Stockman, 2001).	55
Şekil 9	Basit bir Gauss sinyali üzerinde $[-1 \ 1]$ ve $[-1 \ 0 \ 1]/2$ operatörlerinin çıktıları.	57
Şekil 10	“a” orijinal imgesi üzerinde $h_x = [-1 \ 0 \ 1]/2$ ve $h_{xx} = [1 \ -2 \ 1]$ operatörlerini karşılaştıran çıktı dizisi.	58
Şekil 11	“a” orijinal imgesi üzerinde $h_x = [-1 \ 0 \ 1]/2$ ile birinci türev ve h_x 'in arka arkaya iki kere uygulanması.	59
Şekil 12	“a” orijinal imgesi üzerinde $h_x = [-1 \ 1]$ ve $h_{xx} = [1 \ -2 \ 1]$	60

Şekil 13	“a” orijinal imgesi üzerinde $h_x = [-1 \ 1]$ uygulanması.	60
Şekil 14	Farklı suni imgelerin farklı türev operatörleri ile elde edilen çıktılar.....	63
Şekil 15	Ön dolgu ile konvolüsyon. w ön dolgunun genişliğidir.....	64
Şekil 16	conv2 ve ozconv2 fonksiyonlarının karşılaştırılması.	67
Şekil 18	Bir meteorun hareketi	69
Şekil 20	Zamana göre meteorun hareketi.....	70
Şekil 22	İki boyutlu meteor hareketine gösteren bir simülasyon çıktısı	70
Şekil 24	Yay kuvvetinin sisteme eklendiği ilk simülasyonlardan birinin çıktısı	71
Şekil 26	Sentetik verilerle yapılan iki boyutlu simülasyon çıktısı.....	71
Şekil 28	Bir el resmi üzerindeki yıldızlara konulan meteorların aynı başlangıç noktasından başlayarak gerçekleştirdikleri altı simülasyon çıktısı.....	72
Şekil 29	A harfini gösteren iki anahtar kare üzerinde birbiri ile eşleşen SIFT noktaları.....	73

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1	Türk İşaret Dili Alfabesi parmak hecelemeleri üzerinde yedi sınıflama algoritmasının başarı yüzdeleri	20
Çizelge 4.1	Tüm optimizasyonlar bir arada kullanıldığında doğru, yanlış ve boş sınıflamaların sayısı	30
Çizelge 4.2	Deneyden çıkarılan kısmi sınıf karıştırma matrisi. Test kümesinde her sınıftan 5 örnek vardır.	31
Çizelge 5.1	Ortalama bölge sayısına kıyasla rastgele bölgelerin (ExpRand bölgelerinin) başarısı. Üçüncü sütun rastgele bölgelerin standart sapmasını vermektedir. Ortalama ve standart sapmalar 6 çalıştırma sonucunda elde edilmiştir.	36
Çizelge 5.2	Farklı ilgi bölgesi belirleyicileri kullanılarak yapılan nesne sınıfı tanıma denemelerinin sonuçları	39
Çizelge 1	Farklı dolgu teknikleri ile elde edilen ölçek uzayı düzlemlerinin (plane) orijinal imgeden piksel başına farkları. Küçük değerler daha iyidir.	65
Çizelge 2	Ölçek uzayı hazırlanması için piksel başına geçen zaman.	65
Çizelge 3	Farklı dolgu teknikleri ile elde edilen ölçek uzayı düzlemlerinin (plane) bozulmamış (groundtruth) ölçek uzayı düzlemlerinden piksel başına farkları.	66
Çizelge 4	Farklı dolgu teknikleri ile elde edilen ölçek uzayı düzlemlerinin (plane) bozulmamış (groundtruth) ölçek uzayı düzlemlerinden piksel başına farkları.	67

ÖNSÖZ

Tez çalışması uzun bir yolculuk. Bu yolculukta bana destek olan herkese tek tek teşekkürlerimi bildirmek mümkün olmayacağı için ilk sıralarda gelen bir kaç kişiyi saymak istiyorum.

Öncelikle yoğun iş temposuna rağmen düzenli bir şekilde bana zaman ayıran, araştırma ve iş deneyimlerini benimle paylaşan, yazılarımı dikkatle gözden geçiren Songül Albayrak hocama teşekkürü bir borç bilirim. Kendisi olmasa bu tez olmazdı.

Tez izleme komitesindeki jüri üyelerim Cem Ünsalan ve A. Coşkun Sönmez hocalarıma da teşekkür etmem gerekiyor. Her ikisi de tez izleme toplantılarında yapıcı – yönlendirici – ufuk açıcı yorum ve eleştirilerini esirgemediler.

Bölüm başkanımız Oya Kalıpsız hocama bölüm içerisindeki olumlu, dostane yönetim tarzı ve araştırmaların önünü açmadaki gayretleri nedeni ile teşekkür etmeliyim.

Yine bölüm içindeki iş yükünü benimle paylaşan, araştırma, iş ve felsefi konulardaki tartışma ve paylaşımlar ile bölümümüzü daha eğlenceli hale getiren tüm bölüm hocalarıma ve araştırma görevlisi arkadaşlarıma da burada teşekkür etmek isterim.

Hayatım boyunca bana her bakımdan destek olan anne-babam Leyla ve Ahmet Altun’a, kardeşlerim Lale, Hale, Ali ve Ökkeş’e ne kadar teşekkür etsem azdır. Bu tez çalışması sırasında evlendim ve bu yine bu tez çalışması sırasında kızım Selma dünyaya geldi. Bu süreçlerde ailemin kıymetini tekrar tekrar anladım.

Ve ailemin yeni üyeleri; yoğun çalışma dönemlerimde bana katlanan eşim Şeyma, hayatıma yeni bir renk katan Selma... Sizinle herşey bambaşka...

Ağustos, 2010

ÖZET

İLGİ BÖLGELERİ VE YEREL TANIMLAYICILAR İLE GENELLEŞTİRİLMİŞ HOUGH DÖNÜŞÜMÜ VE EN AZ EYLEMSİZLİK EKSENİ TABANLI HIZALAMA YAKLAŞIMI İLE TÜRK İŞARET DİLİ TANIMA SİSTEMİ

Türk İşaret Dili Alfabesi harf işaretleri görüntüler üzerinden tanınmıştır.

Bu amaçla birinci aşamada yönelim açısını kullanarak el nesnelerini hizalayan yeni bir hizalama yöntemi geliştirilmiştir. Hizalanan el işaretleri kullanılarak yapılan sınıflayıcı karşılaştırmasında en iyi sınıflamanın 1NN ve SVM sınıflayıcıları ile sağlandığı görülmüştür.

İkinci aşamada Genelleştirilmiş Hough Transformu, ilgi bölgeleri (interest regions) ve yerel tanımlayıcılar (local descriptors) kullanan bir tanıma sistemi gerçekleştirilmiştir.

Üçüncü aşamada ilgi bölgesi belirleyicilerinin (interest region detector) bir karşılaştırması yapılmış ve EdgeLap (Mikolajczyk vd., 2003) bölgelerinin en iyi tanıma başarısını sağladığı gözlemlenmiştir. İlgi bölgelerinin kalitesini sayısallaştırılmak amacı ile yeni bir ayırt edicilik istatistiği geliştirilmiş ve bu istatistik açısından bakıldığında SURF (Bay vd., 2008) ve DoG (Lowe, 2004) belirleyicilerinin öne çıktığı görülmüştür. Geliştirilen ayırt edicilik istatistiği ilgi bölgesi belirleyicisinin bölgelerinin başarısını rastgele üretilen bölgelerin başarısı ile kıyaslamaktadır. Bu amaçla yeni bir rastgele bölge üreticisi, ExpRand, geliştirilmiştir. Yapılan denemelerde ExpRand üreticisinin en iyi belirleyici ile eş tanıma başarısı verdiği ve ilgi bölgesi belirleyicilerinin birçoğundan daha ayırt edici olduğu gözlemlenmiştir.

Anahtar Kelimeler: Türk İşaret Dili, Parmak Heccelemesi, Genelleştirilmiş Hough Transformu, İlgi Bölgeleri, Yerel Tanımlayıcılar, EdgeLap, SURF, HarAff, HesAff, kAS, FAST, IBR, PCBR, HarLap, HesLap, SR, MSER, DoG, SIFT, En Az Eylemsizlik Eksen, 1NN, SVM, Rastgele Orman, RBF Ağı, MNB, J48, Ayırt Edicilik

ABSTRACT

TURKISH SIGN LANGUAGE RECOGNITION USING GENERALIZED HOUGH TRANSFORM WITH INTEREST REGIONS AND LOCAL DESCRIPTORS, AND USING AXIS OF LEAST INERTIA BASED ALIGNMENT

The recognition of Turkish Sign Language Alphabet finger spelling signs from images is inspected.

In the first stage of the thesis the hand regions are aligned with a novel alignment method based on the orientation angle between axis of least inertia and the y axis. 1NN and SVM come up as the best classifiers in the classification evaluation for aligned hand regions.

In the second stage a recognition system based on Generalized Hough Transform, interest regions and local descriptors are implemented.

In the third stage an evaluation of the interest regions is performed. EdgeLap regions get the best recognition results. In order to quantify the discriminating power of the interest region detectors a novel discriminancy measure is introduced. SURF (Bay vd., 2008) and DoG (Lowe, 2004) regions have the best discriminancy scores. The discriminancy measure introduced compares the recognition success of a region detector with the recognition success of a random region generator. A new random region generator, ExpRand is introduced for this purpose. Regions generated by ExpRand get success rate on par with the best detector and they have more discriminating power than regions of most detectors.

Keywords: Turkish Sign Language, Fingerspelling, Generalized Hough Transform, Interest Regions, Local Descriptors, EdgeLap, SURF, HarAff, HesAff, kAS, FAST, IBR, PCBR, HarLap, HesLap, SR, MSER, DoG, SIFT, Axis of Least Inertia, 1NN, SVM, Random Forest, RBF Network, MNB, J48, Discriminancy

1. GİRİŞ

İşaret dili jestler, yüz ifadeleri ve vücut dilini kullanan bir iletişim yoludur. Çoğunlukla işitme engelli ve işitme sıkıntısı çeken kimseler tarafından kullanılmaktadır. İşaret dilinde iki ana iletişim yaklaşımı vardır: Birinci tipte kullanıcılar her bir kelime için bir işaret yaparlar. İkinci tipte ise kullanıcılar her bir harf için bir işaret yaparlar. Parmak heceleme (finger spelling) de denilen ikinci tip işaret dili kullanımı daha çok kişi ve yer isimleri gibi kelime işareti bulunmayan veya karşıdaki kişi tarafından anlaşılamayan kelimeleri anlatmak için kullanılır.

İşaret dili tanıma sistemleri işitme engelli vatandaşların söylediklerinin anlaşılmasında normal vatandaşlar tarafından kullanılacak sistemlerin hazırlanmasında, genele açık kiosklarda dokunma olmadan etkileşimde bulunmada, robotların kumandasında kullanılabilir.

Bu tez kapsamında Türk İşaret Dili Alfabesi harflerinin parmak hecelemelemlerinin tanınması üzerinde çalışılmıştır. Bu harflerin statik görüntülerin sadece statik görüntüler yardımı ile tanınabilmesi ve sistemin yeterli hızda çalışabilmesi nedeni ile tanıma çalışması video üzerinde değil, videolardan seçilen görüntüler üzerinde yapılmıştır. Bu şekilde tanınan statik görüntüler daha sonra birleştirilerek video üzerinden tanıma da gerçekleştirilebilir.

Bu amaçla birinci aşamada Türk İşaret Dili Alfabesi harfleri parmak hecelemelemleri geliştirilen yeni bir hızlı hizalama yöntemi ile tanınmıştır. Bunun için öncelikle YTÜ Bilgisayar Müh. Bölümünde bu tez çalışması kapsamında oluşturulan parmak heceleme veritabanı videolarından *anahtar kareler*, yani o işareti karakterize eden anlık pozlar, çıkarılmıştır. Bu anahtar karelerden örnekler Şekil 1.1 ile görülebilir.



Şekil 1.1 Türk alfabesinde bulunan 29 harfin parmak hecelemelerinden çıkarılan anahtar kareler.

Anahtar kareler üzerinde YCrCb uzayında yapılan renk analizi ile el bölgeleri elde edilmiş ve bu el bölgeleri en az eylemsizlik eksenini ve y eksenini arasında kalan yönelim açısı yardımı ile aynı hizaya getirilmiştir. Hizalanan el bölgeleri küçük bir sınırlayan karenin içine sığdırılmış ve karenin piksel değerleri özellik vektörü olarak kullanılmıştır. Özellik vektörüne diğer merkezden ortalama uzaklık ve çembersellik gibi değerler de eklenerek son hali elde edilmiştir. Bu özellik vektörü çeşitli sınıflama algoritmaları ile sınıflanmış ve en iyi tanıma başarısının 1NN ve SVM sınıflama algoritmalarının kullanımı ile elde edildiği görülmüştür.

İkinci aşamada Genelleştirilmiş Hough Transformu (Beinglass ve Wolfson, 1991), DoG (Lowe, 2004) ilgi bölgeleri ve SIFT tanımlayıcısı (Lowe, 2004) kullanan bir tanıma sistemi yine aynı veritabanından elde edilen anahtar kareler üzerinde çalışacak şekilde gerçekleştirilmiştir. Basit fakat etkili bir oylama stratejisi kullanmak, Hough birikeci kova büyüklüklerini optimize etmek ve literatürde kullanılan bazı doğrulama adımlarını kullanmamak gibi iyileştirmeler sonucu sistemin başarısı yüzde 94 rakamına ulaşmıştır.

Üçüncü aşamada 12 ilgi bölgesi belirleyicisi Bölüm 4 ile anlatılan Genelleştirilmiş Hough Transformu (Beinglass ve Wolfson, 1991) ile nesne sınıfı tanıma sistemi üzerinde sağladıkları

başarı açısından karşılaştırılmıştır.

Karşılaştırılan ilgi bölgesi belirleyicileri

- EdgeLap (Mikolajczyk vd., 2006), (Mikolajczyk vd., 2003),
- SURF (Bay vd., 2008),
- HarAff (Mikolajczyk ve Schmid, 2004),
- HesAff (Mikolajczyk vd., 2005b),
- kAS (Ferrari vd., 2006b) (Ferrari vd., 2006a), (Martin vd., 2004),
- FAST (Rosten ve Drummond, 2005), (Rosten ve Drummond, 2006),
- IBR (Tuyletaars ve Van Gool, 2000), (Tuyletaars ve Gool, 2004),
- PCBR (Deng vd., 2007),
- HarLap (Mikolajczyk ve Schmid, 2005), (Mikolajczyk ve Schmid, 2001),
- HesLap (Mikolajczyk ve Schmid, 2005), (Mikolajczyk ve Schmid, 2004),
- SR (Kadir vd., 2004),
- MSER (Matas vd., 2002), (Mikolajczyk vd., 2005b),
- DoG (Lowe, 2004)

yöntemleridir.

Karşılaştırılan belirleyicilerden EdgeLap en iyi tanıma başarısını vermekte, ancak bunu büyük miktarda ilgi bölgesi ile yapmaktadır. SURF ve DoG belirleyicileri ikinci ve üçüncü en iyi tanıma başarısını vermekte ve bunu görece olarak daha az bölge ile başarmaktadır. Az sayıda bölge daha sonra bu bölgeleri kullanan algoritmaların hızlı çalışması anlamına gelmektedir.

Belirleyicilerin çıktı sayılarının daha iyi karakterize edilmesi için her bir belirleyicinin ürettiği ortalama bölge sayısı rapor edilmiştir. Genel olarak belirlenen ilgi bölgesi sayısı arttıkça tanıma başarısı da artmaktadır, ancak belli bir üst limite ulaştıktan sonra bölge sayısının artırılması başarıyı artırmamaktadır.

Bir belirleyicinin bölgelerinin ne kadar ayırt edici olduğunu sayısallaştırabilmek için yeni bir “rastgele bölgelere kıyasla ayırt edicilik” istatistiği önerilmektedir. Bu istatistiğin hesaplanması için öncelikle belirleyicinin bölgeleri ile aynı sayıda rastgele bölge üretilir ve daha sonra belirlenen ve üretilen bölgelerin sistemde ayrı ayrı kullanılması ile elde edilen başarı oranları kıyaslanır.

Yapılan deneyler incelendiğinde kör rastgele bölge üreticisinin başarısının en iyi bölge bulucusu ile eşit olduğu görülmektedir. Ayrıca ayırt edicilik açısından bakıldığında rastgele

bölgeler EdgeLap, SURF ve DoG bölgeleri haricindeki tüm bölgelerden dahi iyi değerler almaktadır. İmge veritabanından bağımsız olarak, hızlı, istenen miktarda da üretilebildikleri göz önüne alınırsa rastgele üretilen bölgelerin bir tanıma sistemi için iyi bir seçim olabileceği görülmektedir.

Bu tez çalışmasının 2. bölümünde konu ile ilgili çalışmalar özetlenmekte, 3. bölümünde en az eylemsizlik tabanlı hızlı hizalama ile katı-olmayan nesne sınıfı tanıma sistemi anlatılmakta, 4. bölümünde Genelleştirilmiş Hough Transformu ve ilgi bölgeleri tabanlı katı-olmayan nesne sınıfı tanıma sistemi anlatılmakta ve 5. bölümünde yapılan ilgi bölgeleri karşılaştırılması ve sonuçları verilmektedir. Sonuç ve gelecekte bu çalışmanın devamı olarak yapılabilecek çalışmalar Bölüm 6 ile anlatılmaktadır. Ek 1 tez çalışması kapsamında yapılan bilimsel yayınları listelerken, Ek 2 tez çalışması başlangıcında yapılması düşünülen ancak yapılmayan çalışmaları, Ek 3 ise tez çalışması sırasında yapılan ancak yayına dönüşmeyen çalışmaları anlatmaktadır.

2. KONU İLE İLGİLİ ÖNCEKİ ÇALIŞMALAR

Bu bölümde tez konusu ile ilgili önceki çalışmalara örnekler verilecektir.

2.1 El İşareti, İşaret Dili ve Parmak Heccelemesi Tanıma Çalışmaları

(Göknar ve Yıldırım, 2005) el işaretlerini Yapay Sinir Ağları ile algılamış, (Starner vd., 1998) Amerikan İşaret Dili, (Holden vd., 2005) Avustralya İşaret Dili ve (Gao vd., 2004) Çin İşaret Dili üzerine başarılı çalışmalar yapmıştır.

Türk İşaret Dili tanıma konusunda (Haberdar ve Albayrak, 2006) video üzerinden el yörüngelerini tanımak için Saklı Markov Modelleri (HMM) kullanan bir sistem yapmış, (Aran, 2008) ardışık tümleştirme yöntemi ile inanç tabanlı bir tanıma sistemi geliştirmiş, (Arı, 2008) çok-çözünürlüklü Aktif Şekil Modelleri ve Saklı Markov Modelleri (HMM) kullanan bir takip ve tanıma sistemi gerçeklemiştir.

El *duruşu* (posture) tanıma problemi daha zor bir problemin, el *jesti* (gesture) tanıma probleminin, çözümünde kullanılacak bir adım olarak görülebilir. Bir el jesti el hareketleri içerir. (Chen vd., 2008) el jesti tanımayı iki adıma yaparlar: duruşları tanıma ve duruşları jestler halinde birleştirme. İlk adımda el duruşları Haar-benzeri özellikler (Viola ve Jones, 2001) temelli bir istatistiksel yöntem ve AdaBoost öğrenme algoritması (Freund ve Schapire, 1999) yardımı ile tanınır. İkinci adımda önceden tanınmış duruşların her birine bir kelime, her bir harekete ise birer cümle olarak davranılır. Yapısal bilgiyi tanımlamak için bağlamdan-bağımsız (context-free) bir gramer kullanılır. Bulunan duruşlar gramer kuralına göre dizilere çevirilir. Giriş “cümlesi” onu üretme olasılığı en fazla olan çarpma kuralına göre tanınabilir.

Parmak hecelemesi tanımadaki kullanılan farklı yöntemler de vardır. Örneğin (Machacon ve Shiha, 2010) 41 statik Japon işaret dili hecelemesini tanımak için çok katmanlı geriye yayılma (multilayer perceptron) tabanlı bir yapay sinir ağı kullanır. Sistem bir veri eldiveni (data glove) ile elde edilen sinyaller ile eğitilir. 0.43 başarı rapor edilmiştir.

(Yang ve Lee, 2010) işaret ve hecelemeleri iki katmanlı bir şartlı rasgele alan (conditional random field), BoostMap gömmeleri (embeddings) ve parmak hecelemesi hareketlerinden yardımı ile bulan ve tanıyan bir yöntem anlatır. Veritabanları bir tek işaretçi tarafından işaretlenen, 3 ila 10 işaretten oluşan, 98 cümleden oluşur. İşaretçi eğitim verisi toplama aşamasında yeşil ve mor eldivenler giyerken test videolarının oluşturulma aşamasında bu eldivenleri giymemiştir. Veri kümesi 24 farklı işaret ve 17 farklı parmak hecelemesinden oluşur. 0.78 parmak hecelemesi belirleme ve tanıma başarı raporu edilir.

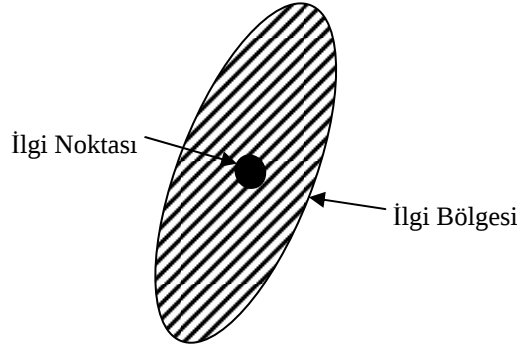
(Goh ve Holden, 2006) Avusturalya işaret dili parmak hecelemelerini 20 parmak hecelemesi buluna bir veritabanında Markov modelleri yardımı ile tanıyabilen bir system sunar. Geometrik özellikler ve optic akı (optical flow) tabanlı hareket özelliklerinin bir kombinasyonunu kullanır. 0.97 başarı rapor eder.

(Wang ve Wang, 2008) el duruşu bulma ve tanıma için SIFT tanımlayıcıları ve ayrık (discrete) AdaBoost öğrenme algoritmasını kullanır. Veritabanları üç sınıftan oluşur: “avuç”, “yumruk” ve “altı”. Tüm sınıflarca ortak olarak kullanılan SIFT özelliklerini imgenin daha baştan elimine edilmesi amacı ile kullanırlar: Ortak özellikler bulunmuyor ise imgede daha fazla arama yapılmaz. Eğer bir imge bu eleme aşamasını geçerse, paylaşılmayan özellikler yardımı ile sınıflandırılır.

Parmak hecelemesi tanıma kolay bir iş değildir, ve literatürde rapor edilen başarı oranları 0.43 ten 0.97 ye kadar değişebilmektedir. Ancak 0.43 başarı oranı rapor eden system diğerlerinden başarısız anlamına gelmemektedir. Başarı oranı farklı işaretçi sayısına, birbirinden ayrılacak sınıf sayısına, eğitim örneklerinin sayısına, eğitim ve test aşamasında renkli eldivenler kullanılıp kullanılmamasına, ışığa, arkaplana, vb. bağlıdır.

2.2 İlgi Noktaları ve İlgi Bölgeleri

İlgi noktaları (interest points) görüntüde yakın çevresindeki piksellere göre bir nedenle öne çıkan noktalardır. Bu noktalar görüntülerde nesnelerin tanınması, farklı kameralarla alınan görüntülerin birleştirilmesi, görüntü sorgulama gibi birçok farklı görüntü işleme problemünde kullanılmaktadır. Çoğu kez, belirlenen bir ilgi noktasının yakın komşuluğundan da yararlanılmaktadır. Örneğin bu komşuluk içinde kalan piksellerden yerel tanımlayıcılar çıkarılıp kümelenerek ilgi noktasına bir tip atanabilmektedir. Bu nedenle ilgi noktası kavramı genişletilip ilgi bölgesi kavramına ulaşılmıştır (Şekil 2.1). Bulunan bir ilgi noktası ve yakın komşuluğu bir ilgi bölgesi oluşturabildiği gibi, ilgi noktası bulmadan direkt bulunabilen ilgi bölgeleri de vardır. Bu noktadan sonra her iki terimin de kullanılabileceği yerlerde ilgi bölgesi terimi kullanılacaktır.



Şekil 2.1 Bir ilgi noktası ve onu destekleyen ilgi bölgesi.

Literatürde birçok farklı ilgi bölgesi belirleme yöntemi vardır. İlgi bölgesi bir imgede bulunduktan sonra çoğu kez ikinci adım ilgi bölgesinin içinden yerel tanımlayıcılar (local descriptors) çıkarmaktır.

2.3 Ölçek Uzayı

İçinde bir ağaç şekli olan bir imgeye en alt ölçekte bakıldığında ağaç yapraklarının damarları görülebilir. Biraz daha yüksek ölçekli bakıldığında damarlar kaybolur ve ağaç yaprakları küçük bölgecikler olarak kalır. Biraz daha yüksek bir ölçekte ise yapraklar kaybolur ve tüm ağaç şekli tek bir şekil olarak görülür. Başka bir deyişle düşük ölçekte nesnelerin ayrıntıları öne çıkarken, daha kaba/yüksek (coarse) ölçeklerde nesnenin geneli öne çıkmaya başlar. Aynı imgenin farklı ölçeklerdeki gösterimlerini elde edebilmek ve tüm ölçeklerde gerekli aramaları, eşlemeleri yapmak işlem sonuçlarını iyileştirmektedir. Bu nedenle ilgi bölgesi çıkarımında ve daha birçok görüntü işleme uygulamasında imgenin belirli aralıklardaki ölçek gösterimlerini (representations) içeren ölçek uzayı (scale-space) çıkarılır ve tüm ölçek uzayında arama yapılır. Uygulamada her bir yüksek ölçek bir alt ölçeğin Gauss benzeri bir fonksiyon ile yumuşatılması ile elde edilir. Ölçek uzayı çıkarımının gerçekleşmesi ve en uygun ölçeğin seçilmesi gibi ayrıntılar için T. Lindeberg (Lindeberg ve Bretzner, 2003) ve R.S. Eaton (Eaton vd., 2006) çalışmalarına bakılabilir.

2.4 İlgi Bölgesi Çıkarma Yöntemlerinin Nesne Sınıfı Tanıma Açısından Karşılaştırılması

İlgi bölgelerinin özellik eşleme açısından çok ayrıntılı karşılaştırmaları bulunmakla beraber (Mikolajczyk ve Schmid, 2005),(Mikolajczyk vd., 2005b),(Moreels ve Perona, 2007) nesne sınıfı tanıma açısından böyle ayrıntılı çalışma bulunmamaktadır. Aşağıda bilinen çalışmaların kısa bir özeti verilmektedir.

Seeman vd. (Seeman vd., 2005) ilgi bölgelerinden Harris (Harris ve Stephens, 1988),(Schmid ve Mohr, 1997) , Harris-Laplace (Mikolajczyk ve Schmid, 2001), Hessian-Laplace (Mikolajczyk ve Schmid, 2005) ve DoG (Lowe, 2004)bölgelerini yaya belirleme açısından değerlendirir. Belirleme adımı Gizli Şekil Modeli (Implicit Shape Model),(Leibe, B. ve Schiele, 2003)ile gerçekleştirilir. Hessian-Laplace bölgelerinin en başarılı bölgeler olduğunu, doğru bölge seçiminin sonuca önemli ölçüde etki ettiğini ve daha fazla bölgenin daha iyi belirleme sonuçları verdiğini rapor ederler.

Mikolajczyk vd. (Mikolajczyk vd., 2005a) ilgi bölgelerinden Harris-Laplace (Mikolajczyk ve Schmid, 2004), Hessian-Laplace (Mikolajczyk vd., 2005b), DoG (Lowe, 2004), SR (Kadir vd., 2004) ve MSER (Matas vd., 2002) ilgi bölgelerini nesne sınıfı tanıma açısından karşılaştırır. Birçok yeni nesne sınıfı tanıma sisteminin bir ara gösterim olarak ilgi bölgelerinin kümelerini kullanmasını gerekçe göstererek ilgi bölgelerinin *kümelerinin* kalitesini değerlendirir ve sonuçlarını bir yaya belirleme sistemi ile doğrularlar. Mikolajczyk ve Schmid'in nesne sınıfı tanımaya yönelik olmayan önceki karşılaştırmasından (Mikolajczyk ve Schmid, 2004) farklı bir sıralamaya ulaşırlar. Bu defa Hessian-Laplace birinci olurken, SR ikinciliği alır. Ayrıca Hessian-Laplace ve SR ilgi bölgelerinin birbirini iyi tamamladığını çünkü bunların aralarında paylaşılan kümelerin sayısının diğerlerine göre az olduğunu rapor ederler. Ayrıca çok az sayıda bölge sağladığı için MSER belirleyicisinin başarısız kaldığını belirtirler.

Stark ve Schiele (Stark ve Schiele, 2007) ilgi bölgelerinden Harris-Laplace (Mikolajczyk ve Schmid, 2005), Hessian-Laplace (Mikolajczyk ve Schmid, 2005) ve SR (Kadir vd., 2004) ilgi bölgelerini farklı şekil ve görünüm tabanlı tanımlayıcılar ile Naif Bayes (Naive Bayes) ve Yerelleştirilmiş Kelime Torbaları (Localized Bag-of-Words) tanıma algoritmalarını kullanarak karşılaştırır. Onlar da (Mikolajczyk vd., 2005a) gibi ilgi bölgesi kümelerinin kalitesini değerlendirir. Asıl amaçları şekil tabanlı (shape based) tanımlayıcılar ile görünüş tabanlı (appearance based) tanımlayıcıları karşılaştırmak olduğu için bu amaca uygun 10 sınıf içeren bir veritabanı kullanırlar. Ayrıca Caltech-101 veritabanından (Li vd., 2004)seçilen 10 sınıfı içeren ikinci bir veritabanı da kullanmaktadırlar. Genelde sıralama Hessian-Laplace, Harris-Laplace ve SR bölgeleri olarak gitmektedir. Kullandıkları veritabanı için yerel özelliklerin pozisyon bilgisini kullanmanın performansı ilgi bölgesi veya tanımlayıcı seçiminden daha çok etkilediğini rapor etmektedirler.

Nowak vd. (Nowak vd., 2006) tarafından yapılan bir çalışmada LoG (Lindeberg, 1993) ve Harris-Laplace (Lazebnik vd., 2003)ilgi bölgeleri ve rastgele bir bölge üretici tarafından

üretileen rastgele bölgeler Özelliklerin Torbası (Bag-of-Features) tabanlı bir imge sınıflama uygulaması üzerinde karşılaştırılmıştır. Rastgele bölgeler her bir ölçek düzleminin (scale plane) eşit büyüklüklerdeki ızgaralara (grid) ayrılması ve rastgele bir ızgara karesinin seçilmesi ile elde edilir. Tüm ızgara karelerinin seçilme olasılığı eşittir, dolayısı ile küçük ölçeklerden daha fazla bölge gelmektedir. Sınıflama başarısını etkileyen tek faktörün bölgelerin sayısı olduğunu rapor etmektedirler. İlgi bölgesi belirleyicileri bir rastgele bölge üreticisi kadar çok bölge belirleyemedikleri için başarıda onun gerisinde kalmaktadırlar. Daha şaşırtıcı bir sonuç ise ilgi bölgesi belirleyicileri rastgele bölgelerle benzer sayıda bölge ürettiklerinde yine de rastgele üreticinin başarısının gerisinde kalmaktadırlar.

Bu tezde Bölüm 5’de anlatılan çalışma (Nowak vd., 2006) çalışmasına rastgele bölgelerin performanslarının değerlendirilmesi açısından benzemektedir. Başlıca farklar şunlardır: Bölüm 5 ile anlatılan çalışmada çok daha fazla sayıda ilgi bölgesi değerlendirilmiştir, rastgele bölgeler farklı bir yöntemle elde edilmiştir ve bölge kalitesinin rastgele bölgelerin başarısına bağlı bir kalite ölçeği teklif edilmiştir. Başka bir fark Bölüm 5 ile anlatılan çalışmada Genelleştirilmiş Hough Transformu tabanlı ilgi bölgelerinin görelî yerlerini kullanan bir yöntem kullanılırken, (Nowak vd., 2006) çalışmasında Özelliklerin Torbası tabanlı ilgi bölgelerinin görelî yerlerini kullanmayan bir yöntem kullanılmasıdır. (Nowak vd., 2006) çalışması sonuçlarını altı farklı veritabanı üzerinden çıkarırken, Bölüm 5 ile anlatılan çalışmada sadece bir veritabanı kullanılmıştır. Ayrıca Nowak vd. çalışmasında bir imge sınıflama uygulaması, Bölüm 5 ile anlatılan çalışmada ise bir katı-olmayan nesne sınıfı tanıma çalışması ölçüt olarak kullanılmaktadır. Bu farklara rağmen Bölüm 5 ile anlatılan çalışmada Nowak vd. çalışmasında verilen başarı oranı ve bölge sayısı ile ilgili sonuçla teyit edilmektedir: ilgi bölgesi sayısı arttıkça başarı oranı da artar. Ancak, Bölüm 5 ile anlatılan çalışmada belli bir doyum noktası, yani daha fazla ilgi bölgesi eklemenin başarıyı yükseltmediği bir nokta olduğu fark edilmektedir. Ayrıca çoğu ilgi bölgesi bulucusunun aynı miktar bölge üreten bir rastgele bölge üreticisinden başarı olarak geride kaldığı sonucu da teyit edilmektedir.

2.5 İlgi Bölgelerinden Çıkarılan Tanımlayıcılar

İlgi bölgeleri içinden bölgeyi tanımlayan yerel tanımlayıcılar (local descriptor) çıkarılır. Mikolajczyk vd. (Mikolajczyk ve Schmid, 2005) literatürde görülen tanımlayıcıların farklı imge tiplerinde performanslarını karşılaştırmıştır. Karşılaştırılan tanımlayıcılar arasında çapraz-ilinti (cross-correlation), Lowe’nin SIFT tanımlayıcısı (Lowe, 2004), SIFT tabanlı tanımlayıcılardan GLOH (Mikolajczyk ve Schmid, 2005) ve PCA-SIFT (Ke ve Sukthankar,

2004), şekil bağlamı (shape context) (Belongie vd., 2002), dönen imgeler (spin images, (Lazebnik vd., 2005)), yönlendirilebilir filtreler (steerable filters, (Freeman ve Adelson, 1991)), diferansiyel değişmezler (differential invariants, (Koenderink ve Doorn, 1987)), kompleks filtreler (complex filters, (Schaffalitzky ve Zisserman, 2002)) ve moment değişmezleri (moment invariants, (Gool vd., 1996)) vardır. Genellikle GLOH ve SIFT en iyi sonuçları verirken şekil bağlamı onları izlemektedir. Ancak bol doku içeren ve belirgin kenarlar içermeyen imgelerde şekil bağlamının performansı düşmektedir. Başarı sıralaması PCA-SIFT, moment değişmezleri, çapraz-ilinti, yönlendirilebilir filtreler, dönen imgeler, diferansiyel değişmezler ve kompleks filtreler ile devam etmektedir. Özetle sonuçlar SIFT ve SIFT tabanlı tanımlayıcıların başarısını göstermektedir.

SIFT (Lowe, 2004) tanımlayıcısı bölge içindeki gradient (iki boyutlu birinci türev vektörü) dağılımını karakterize eder. Bunun için öncelikle imgedeki tüm piksellerin gradient değerleri elde edilir. Ardından SIFT tanımlayıcısı çıkarılmak istenen bölge ($4 \times 4 = 16$) alt kısma ayrılır. Her alt kısımda gradient yönelimleri 8 kovaya (sağ, sol, üst, alt, sağ-üst, sol-üst, sağ-alt, sol-alt) pay edilir. Her kovada biriken toplam gradient miktarları hesap edilir. Böylece o alt kısımdan 8 boyutlu bir özellik vektörü çıkarılmış olur. Diğer 15 alt kısımdan gelen özellik vektörleri ile birleştirilerek ($16 \times 8 = 128$ boyutlu) SIFT tanımlayıcısı elde edilir.

2.6 Tanıma Amaçlı İlgi Bölgeleri ve Tanımlayıcıları

Mikolajczyk vd. (Mikolajczyk vd., 2005a) literatürde öne çıkan 5 ilgi bölgesi bulucusu ve 5 tanımlayıcıyı nesne sınıfı tanıma problemi açısından karşılaştırmıştır. Karşılaştırılan ilgi bölgesi bulucular Harris-Affine, DoG, Hessian-Affine, SR ve MSER bulucularıdır. Karşılaştırılan tanımlayıcılar ise SIFT, GLOH, PCA-SIFT, moment değişmezleri ve çapraz-ilintidir. Bu karşılaştırmada Hessian-Affine bölgeleri üzerinden hesaplanan GLOH tanımlayıcıları sistematik olarak en iyi sonucu vermiştir. GLOH tanımlayıcısı SIFT tanımlayıcısının bir genişlemesi olduğundan bu karşılaştırma yine SIFT türü tanımlayıcıların üstünlüğünü göstermektedir. SR tanımlayıcılarının da iyi sonuçlar verdiği ve GLOH ile birbirini tamamlayacak nitelikte olduğu belirtilmektedir. MSER ve PCA-SIFT tanıma için değil eşleme için uygun görünmektedir. MSER çok ayırt edici bölgeler oluşturmaktadır, ancak tanıma amaçlı kullanım için yeterli sayıda bölge üretmemektedir.

2.7 Video Üzerinde İlgi Bölgeleri İle Çalışmalar

İlgi bölgelerinden çıkarılan tanımlayıcılar yardımı ile şekil tanıma çalışmaları videodan hareket tanımaya genişletilebilir. Bu amaçla yapılan çalışmalar iki yönde toplanmaktadır:

1 – Videoya özel ilgi noktalarının geliştirilmesi. Örneğin Laptev (Laptev, 2005), iki boyutlu Harris nokta/köşe bulucusunu (Harris ve Stephens, 1988) üç boyuta (2 uzay boyutu + zaman boyutu) genişletmiş ve bu üç boyutlu noktaların çevresinden çıkardığı özellikler ile videodan hareket tanımıştır. Video veritabanı koşan, el çırpın, yumruk atan, vb. insan hareketlerinden oluşmaktadır. Hareketi kendi detaylı matematiksel modelleri ile modellemiştir. Dollar vd. (Dollar vd., 2005), Laptev noktalarının yüz hareketleri gibi güçsüz hareketlerde iyi sonuç vermediğini rapor etmiş ve zaman boyutunda Gabor filtrelemesi yardımı ile bulunan kendi noktalarını teklif etmiştir. Noktalar etrafında çıkarılan özellikleri k-means ile kümeleyerek nokta prototiplerini elde etmiş, sonra da bu prototiplerin histogramlarını hareketi modellemek için kullanmıştır. Niebles vd. (Niebles vd., 2006), Dollar'ın noktalarını kullanmış, ancak hareketleri istatistiksel bir yaklaşımla modellemişlerdir.

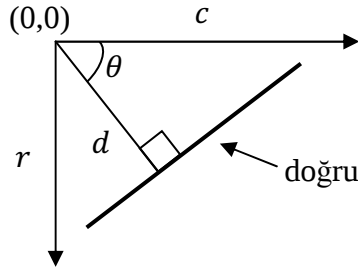
2- Video kareleri yardımı ile daha dayanıklı (örneğin birkaç karede devam eden) ilgi noktalarının bulunması. Örneğin Sivic vd. (Sivic ve Zisserman, 2003) iki uzay boyutunda Mikolajczyk ve Schmid'in (Mikolajczyk ve Schmid, 2002) afin (afine) noktalarını ve Matas vd. 'nin (Matas vd., 2002) MSER noktalarını bulmuş ve bu noktaların en dayanıklılarını bulmak için aralarından en az 3 kare devam eden noktaları seçmiştir.

2.8 Standart Hough Transformu

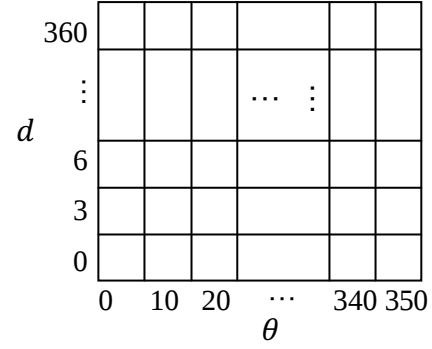
Standart Hough Transformu (SHT) olarak da adlandırılan orijinal Hough Transformunda, imge kenarları üzerindeki her (r, c) noktası polar koordinatlarda (d, θ) ikilisi ile parametrize edilen ve (2.1) ile ifade edilen doğrulara oy verir (Şekil 2.2).

$$d = r \sin \theta + c \cos \theta \quad (2.1)$$

Burada d doğrunun imge merkezine uzaklığı, θ ise imge merkezinden doğruya uzayan vektörün imge merkezi koordinat sistemi x eksenine ile yaptığı açıdır. SHT *birden çoğa* ($1 \rightarrow m$) bir algoritmadır. Yani her bir nokta olası birikeçte birçok seleye, birçok doğru için, oy verir.



(a)



(b)

Şekil 2.2 (a) Doğru denklemi (2.1) içinde kullanılan d ve θ parametreleri. (b) Hough birikeci. Kaynak: (Shapiro ve Stockman, 2001)

2.9 Genelleştirilmiş Hough Transformu

Hough Transformu imgeler üzerinden çizgi, çember, elips gibi analitik olarak tanımlanmış şekilleri bulmak için kullanılan bir yöntem iken her türlü şekli bulacak şekilde geliştirilmiştir. Beinglass ve Wolfson (Beinglass ve Wolfson, 1991) yöntemi geometrik anahtarlama (geometric hashing) yöntemine de benzer bir şekilde her türlü nokta setini modelleyecek ve tanıyacak kapsama ulaştırmıştır.

Ballard'ın çalışmasında (Ballard, 1981) Genelleştirilmiş Hough Transformu R-tablosu (R-Table) adı verilen bir anahtarlama hafıza (hash) yardımı ile düzlemde kaymış nesneleri tanıyabilmektedir. Dönen ve boyutu değişen nesneler için ise olası açı ve büyüklük değerlerinde tam kapsamlı arama (exhaustive search) gerekir. Beinglass ve Wolfson tam kapsamlı arama gerekliliğini bileşik özelliklerden (compound feature) elde edilen şekil belirteçlerinin (shape signature) kullanımı ile kaldırmıştır.

Örneğin afin (affine) dönüşüm geçiren bir nesnenin tanınmasında, aynı doğru üzerinde olmayan herhangi dört nokta bir bileşik özellik kabul edilir. Bileşik özellikten afin dönüşüm sonucu bozulmayan bir şekil belirteci çıkartılabilir. Kullanılabilecek olası şekil belirteçlerinden biri dördüncü noktanın diğer üç noktadan elde edilen afin koordinat sistemindeki yeridir (Lamdan vd., 1988).

2.10 Genelleştirilmiş Hough Transformu ile İmgelerde Şekil Tanıma Çalışmaları

Genelleştirilmiş Hough Transformu Ballard'ın versiyonunda (Ballard, 1981) R-tablosu ismi verilen bir veritabanı ve zekice bir indeksleme şeması yardımı ile çalışır. Yöntem bu

versiyonda tüm olası rotasyon ve ölçek değişimlerini kontrol etmek zorundadır.

Beinglass ve Wolfson (Beinglass ve Wolfson, 1991) nesnelerin eklem noktalarını “referans noktası” olarak seçerek Genelleştirilmiş Hough Transformunu eklemli nesneleri de tanıyabilecek şekilde iyileştirmişlerdir. Ayrıca ilgi noktası kümelerinden “şekil imzası” (shape signature) oluşturmuş ve böylece tüm rotasyon ve ölçek değerlerinin üzerinde yürüme zorunluluğunu kaldırmışlardır. Hough Transformunun gerçekleşmesini ayrıntılı olarak anlatmışlar ve makas, pusula, çekiç gibi küçük aletlerin tanınma örneklerini vermişlerdir. Modern ilgi bölgeleri ve yerel tanımlayıcılar o zaman bilinmediği için yaklaşımları yerel koordinat sistemleri ve yerel tanımlayıcıları tanımlamak için ilgi bölgelerinin *gruplarını* kullanmak zorunda kalır. Örneğin rotasyon, ölçek ve taşıma değişmezliği elde edilmek istenirse (rotation, scale, translation invariance) üç ilgi noktası yardımı ile bir yerel koordinat sistemi oluşturulur. Dördüncü bir noktanın bu koordinat sistemine göre yeri yerel tanımlayıcı olarak kullanılır. Bu durum R-tablosuna konulması gereken yerel tanımlayıcı sayısının $\Theta(n^4)$ karmaşıklığı ile artmasına (n ilgi noktası sayısıdır) neden olur. Neyse ki modern imge ilgi bölgeleri tek başlarına üzerlerine birer koordinat sistemi koymaya izin verirken, her bir ilgi bölgesinden de SIFT gibi birer yerel tanımlayıcı çıkarılabilir.

Hough Transformunun modern ilgi bölgeleri ve tanımlayıcıları kullanan bir örneği Lowe’nin çalışmasında (Lowe, 2004) verilmiştir. Lowe bu çalışmada DoG ilgi bölgeleri ve SIFT tanımlayıcıları çıkarma yöntemlerini de anlatır. SIFT tanımlayıcıları ölçek ve yönelimden bağımsızdırlar ve diğer yerel tanımlayıcılara göre daha iyi sonuçlar verdikleri gösterilmiştir (Mikolajczyk ve Schmid, 2005). Aynı çalışma Hough Transformunun bu tanımlayıcılar yardımı ile nasıl şekil eşlemede kullanılabileceğini de anlatır. Hough Transformu ile elde edilen nesne hipotezleri aynı yazarın önceki bir çalışmasında (Lowe, 2001) anlatılan bir olasılık modeli ile doğrulanır.

Hough Transformunun bir başka enteresan kullanımı Leibe ve arkadaşlarının (Leibe, Bastian vd., 2004) çalışmalarında görülebilir. Gizli Şekil Modeli (Implicit Shape Model) olarak isimlendirdikleri model hem nesne tanıma hem de bölütleme amacı ile kullanılır. Algoritmalarının performansını standart bir veritabanında (Agarwal ve Roth, 2002) gösterirler. Nesneler farklı eklemlemelerde (articulation) olabilir, farklı dokulara sahip olabilir ve bazı kısımları görünmeyebilir. Kullandıkları araba veritabanında iki sınıf vardır: “bir ya da daha fazla araba bulunur” ve “araba bulunmaz”. Metotlarının bir kısmı olasılık tabanlı bir Hough Transformu olarak görülebilir.

2.11 İlgi bölgeleri ve Genelleştirilmiş Hough Transformu

İlgi bölgelerinden çıkarılan tanımlayıcıların nesne tanıma (Ferrari vd., 2004),(Lowe, 2004) ve nesne kategorileri tanıma (Dorko ve Schmid, 2003),(Fergus vd., 2003),(Opelt vd., 2004) amaçlı kullanılması ile başarılı sonuçlar elde edilmektedir.

Lowe (Lowe, 2004), görüntü üzerinden nesne tanımak için, görüntünün ölçek-uzayında ikinci türevin en düşük ve en yüksek değerleri aldığı noktaların çevresini ilgi bölgesi olarak kullanmıştır (DoG bölgeleri). Bu noktalar etrafındaki renk değişiminin yöneliminin (gradient orientation) yerel histogramlarını ise tanımlayıcı olarak kullanmıştır. SIFT tanımlayıcısı olarak adlandırılan bu tanımlayıcı Mikołajczyk vd.'nin (Mikolajczyk vd., 2005a) tanımlayıcılar arasında yaptığı performans karşılaştırmasında en iyi sonuçları üretmiştir ve son zamanlarda yapılan çalışmalarda sıklıkla kullanılmaktadır. Ayrıca Lowe, çalışmasında ilgi noktalarının birbirine göre konumlarını Genelleştirilmiş Hough Transformu (Beinglass ve Wolfson, 1991) ile modellemiş ve etkileyici sonuçlar raporlamıştır.

Leibe vd. (Leibe, Bastian vd., 2004), (Leibe, Bastian vd., 2005) nesneleri Gizli Şekil Modeli (Implicit Shape Model) adını verdikleri olasılık değerli oylama tabanlı bir modelle modellemişlerdir. Bu model kullanılarak nesne imgede tanınabildiği gibi, imgede var olduğu alan da bölümlenebilmektedir (segmentation). Gizli Şekil Modeli incelendiğinde bunun aslında yine Genelleştirilmiş Hough Transformunun olasılık tabanlı bir gerçekleştirilmesi olduğu görülür. Leibe vd. bu modelle (Leibe, Bastian vd., 2004)'de araba gibi katı nesneleri modelledikleri gibi inek şekilleri gibi esnek nesneleri modellemişler, (Leibe, Bastian vd., 2005) da ise insan şekillerini modellemiş ve son derece karmaşık - kalabalık cadde sahnelerinde insanları bulmuşlardır.

2.11.1 Hough Transform Modeli Oluşturma - Doğrulama

Bir nesne sınıfı tanıma sisteminde öncelikle nesne sınıfını en iyi temsil edebilecek bir model tanımlanır, sonra tanımlanan model elde bulunan birden fazla eğitim nesnesi yardımı ile oluşturulur ve nesnenin imgedeki yerine dair hipotezler model yardımı ile doğrulanır. Aşağıda Hough Transformu tabanlı nesne sınıfı modellerinden örnekler verilmiştir:

Sehgal ve Desai (Sehgal ve Desai, 2002) geometrik anahtarlama (geometric hashing) (Shapiro ve Stockman, 2001) ve poz sınıflama (pose clustering) (Shapiro ve Stockman, 2001) yöntemlerini birleştirdikleri çalışmalarında aynı sınıfa ait eldeki N eğitim nesnesinin tamamından çıkarılan özellikleri (noktaları) kapsayan bileşik özellik haritasını model olarak kullanmıştır. Araştırmacılar bu modeli oluşturmak için eğitim nesnelerini poz kümeleme

(pose clustering) ile hizalamıştır. Değişmezleri ve nesne referans noktasını da yeni bileşik model üzerinden çıkarmıştır.

Lowe (Lowe, 2004)git gide daha kısıtlı şartlar kontrol ederek en sona kalan hipotezleri doğrulanmış kabul etmektedir: Hough Transformu birikecinde, eğer bir seledede 3 den fazla oy var ise o seleye oy veren noktaların yardımı ile hipotez ile model arasındaki dönüşüm parametrelerini bulunur. Dönüşüm gerçekleştirilir ve seledeki noktalardan olması gereken yere yeterince yakın olmayan noktalar (outlier) elenir. Eleme sonucunda geriye 3 değerinden az nokta kalırsa bu defa tüm hipotez elenir. Bu yaklaşım (önce noktaların sonra hipotezin elenmesi) istenen sayıda veya değişim devam ettiği sürece tekrarlanır. Her defasında noktaların modele daha yakın olması istenir. Daha sonra beklenen yanlış eşleme sayısı ve doğru eşleme sayısına bağlı olasılıksal bir eleme daha yapılır ve geriye kalan hipotezler doğrulanmış kabul edilir.

2.11.2 Gürültülü Değerler ile Baş Etme

İmge – video üzerinde bulunan noktaların yerleri (koordinatları) son derece gürültülü değerlerdir. R-Tablosu ve birikeç gürültülü değerlerin nicemlenmesi ile elde edilmiş sele değerlerine sahiptir. Dolayısı ile sık sık yanlış seleye oy verilmektedir. Sele değerlerinin çok büyük tutulması yer değerlerine oy atmayı anlamsız kılmakta, çok küçük tutulması ise yanlış oy atma sayısını artırmaktadır. Literatürde bu problem ile baş etme amacı ile kullanılan yöntemlerin arasında birikeçteki oyların kümelenmesi / filtrelenmesi ve oylamada olasılıksal yaklaşımların kullanılması görülmektedir.

Anelli vd. (Anelli vd., 2007) nesne modeli ile test örnekleri arasındaki şekil farkı ve gürültü ile baş edebilmek için üç önlem alır. Öncelikle birbirine benzeyen ancak aynı olmayan şekillerin karşılık gelen noktaları arasında büyük yönelim farklılıkları olduğu gerekçesi ile R-Tablosu indekslemesinde yönelim kullanmaz. İkinci bir önlem olarak R-Tablosunu nicemlemez. Uzaklıkların gerçel sayı değerlerini (selelere ayırmadan) bir diziye atar. Ve her bir test noktası için tüm dizide arama yapıp en yakın değişmez değerlerinin oy vermesini sağlar. Son olarak oylama bitince bir oy dağıtma penceresi (vote dispersion window) yardımı ile her seleye komşuluğundaki ve kendisindeki oyların toplamını değer olarak atarlar. Böylece $2l + 1$ boyutundaki kare şeklindeki bir dağıtma penceresi yardımı ile bir noktanın l birim uzağındaki noktalar da aynı merkez için oy kullanabilir. Oy dağıtma penceresi kullanımı aslında 2 boyutta 1'lerden oluşan bir kernel ile konvolüsyona eşdeğerdir. Aynı zamanda hızlı bir kümeleme işlemi olarak da görülebilir. Anelli vd. işlemin hızlı yapılabilmesi için dinamik programlama temelli bir yaklaşım da sunar.

3. EN AZ EYLEMSİZLİK EKSENİ TABANLI HIZLI HIZALAMA İLE TÜRK İŞARET DİLİ TANIMA SİSTEMİ

Tez çalışmasının bu aşamasında Türk İşaret Dili parmak hecelemelerini tanıyan bir sistem gerçekleştirilmiştir. Bu amaçla işaret videolarından anahtar çerçeveler (kareler) çıkarılmış, bu anahtar çerçevelerden ten rengindeki bölgeler elde edilerek işaret nesneleri elde edilmiş ve işaret nesneleri yönelim açısı tabanlı hızlı hizalama ile hizalanmıştır. Hizalanan nesneler bir minimum sınırlayan kare (minimum bounding square) içerisine yerleştirilip sabit bir boyuta indirgenmiştir. İndirgenmiş boyuttaki minimum sınırlayan karenin ikili piksel değerleri özellik olarak kullanılırken, bu özelliklere ortalama merkezden uzaklık (mean radial distance) ve çembersellik (circularity) gibi özellikler eklenmiştir. Elde edilen özellikler ile farklı sınıflama yöntemleri üzerinde denemeler yapılmış ve tanıma başarıları rapor edilmiştir.

3.1 Anahtar Kare Çıkarımı

Türk İşaret Dili kelime işaretlerinden farklı olarak Türk İşaret Dili harf hecelemelerinin büyük kısmı sadece statik şekil yardımı ile (hareket olmaksızın) tanınabilir. Bu özellikten yararlanmak ve işlem hızını artırmak amacı ile videolardan anahtar kareler çıkarılır. Şekil 1.1 29 Türk Alfabesi harfi için örnek anahtar kareler göstermektedir.

Anahtar kareler bir Türk İşaret Dili harf işaret videosunda üzerinde en az hareket bulunan karelerdir. Bu nedenle arkasından gelenle arasındaki, (3.1),(3.2),(3.3),(3.4) denklemleri ile verilen, fark (uzaklık) değeri en az olan çerçeve anahtar kare olarak seçilir.

$$D^f = \sum_n |\Delta R_n^f| + |\Delta G_n^f| + |\Delta B_n^f| \quad (3.1)$$

$$\Delta R_n^f = R_n^{f+1} - R_n^f \quad (3.2)$$

$$\Delta G_n^f = G_n^{f+1} - G_n^f \quad (3.3)$$

$$\Delta B_n^f = B_n^{f+1} - B_n^f \quad (3.4)$$

(3.1),(3.2),(3.3),(3.4) denklemlerinde f kare numarasını, D^f terimi f ve $f+1$ kareleri arasındaki farkı, n piksel numarasını gösterirken R , G ve B piksel renk bileşenlerini göstermektedir.

3.2 Renk Kullanarak Ten Belirleme

Bir sonraki adımda anahtar kareler üzerinde ten bölgeleri ten rengi ile belirlenir. İmgede ten bölgeleri haricinde ten renginde nesne olmadığı varsayılır. Ten belirleme amacı için YCrCb renk-uzayının diğer renk uzaylarına göre daha başarılı olduğu gösterilmiştir (Sazonov vd., 2003). Bu nedenle imgeler öncelikle (3.1),(3.5) ve (3.6) denklemleri kullanılarak YCrCb renk-uzayına geçirilir.

$$Y = 0.299R + 0.587G + 0.114B \quad (3.5)$$

$$C_r = B - Y \quad (3.6)$$

$$C_b = R - Y \quad (3.7)$$

Gürültünün azaltılması için Y , C_r ve C_b bileşenlerinin her biri (3.8) ile verilen iki boyutlu Gauss filtresi ile yumuşatılır.

$$F(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.8)$$

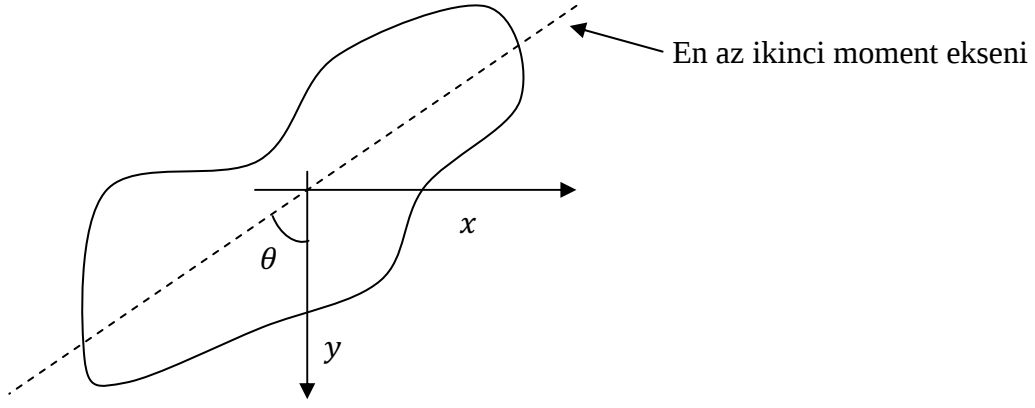
Chai ve Bouzerdom (Chai ve Bouzerdom, 2000) ten bölgelerine ait olan piksellerin benzer C_r ve C_b değerlerine sahip olduklarını bildirir ve ten piksellerinin C_r - C_b düzlemindeki dağılımını verir. Bu bilgiye dayanarak bir pikselin Y , C_r ve C_b değerleri $135 < C_r < 180$, $85 < C_b < 135$ ve $Y > 80$ aralıklarında ise ten rengi olarak değerlendirilir.

Çok küçük ten rengindeki bölgeler morfolojik açma (morphological opening) ile temizlendikten sonra ten rengi belirleme adımı tamamlanmış olur. Şekil 3.2a ve b bir ten rengi belirleme örneği vermektedir. Videolar arka planda ten renginde nesne olmamasına dikkat edilerek çekilmesine rağmen %2 civarında hatalı bölge bulunan imge olmaktadır. Bu imgelerde el bölgesi tamamen bulunurken fazladan arka plan bölgeleri de el olarak döndürülmektedir.

3.3 Hızlı Hizalama

Ten rengi ile elde edilen el bölgelerinin sınıflama algoritmalarına verilecek özellik vektörlerine dönüştürülmeden önce hizalanması gerekmektedir.

Bu çalışmada en az ikinci moment eksen (Axis of least second moment, (Umbaugh, 1998)) tabanlı bir hızlı hizalama yöntemi teklif edilmektedir. Yöntem nesnenin en az ikinci moment eksen ile y eksen arasındaki açının (yönelim açısı, *angle of orientation*) sıfırlanacak şekilde döndürülmesi ile çalışır (Şekil 3.1).



Şekil 3.1 En az ikinci moment eksen ve yönelim açısı θ .

Şekil 3.1 ile betimlenen yönelim açısı θ (3.9),(3.10),(3.11),(3.12) ve (3.13) denklemleri yardımı ile elde edilebilir.

$$2\theta = \arctan\left(\frac{2M_{11}}{M_{20} - M_{02}}\right) \quad (3.9)$$

$$M_{11} = \sum_x \sum_y xyI(x, y) \quad (3.10)$$

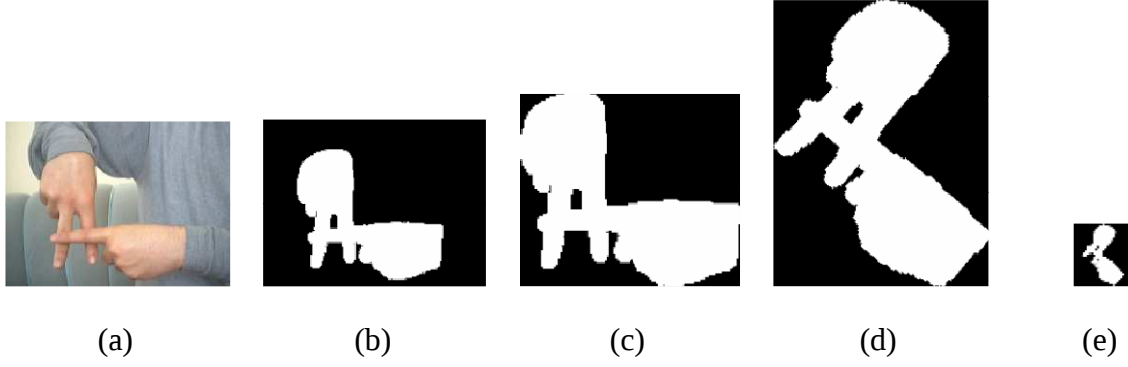
$$M_{20} = \sum_x \sum_y x^2 I(x, y) \quad (3.11)$$

$$M_{02} = \sum_x \sum_y y^2 I(x, y) \quad (3.12)$$

$$I(x, y) = \begin{cases} 1, & (x, y) \text{ pikseli nesne üzerinde ise} \\ 0, & \text{aksi halde} \end{cases} \quad (3.13)$$

Yönelim açısı tabanlı hizalama küçük yönelim farklarını kompanse edebilen, ancak büyük yönelim farklarına cevap verebilen bir yöntemdir. Bu durum bazı işaretleri, örneğin U ve C harflerinin işaretleri, şekil olarak birbirine benzeyen ancak yönelim bilgisi ile ayrılabilen Türk İşaret Dili harfleri için olumlu bir özelliktir.

Bir nesnenin tüm piksellerini kapsayan kareye *sınırlayan kare* denilsin. İmgeleri bir sınırlayan karenin içine koyarak ve sınırlayan kareyi sabit, küçük bir çözünürlüğe indirgeyerek hızlı hizalama işlemi tamamlanır. Sınırlayan karenin piksel değerleri özellik vektörü olarak kullanılır.



Şekil 3.2 Hızlı hizalama işleminin aşamaları. (a) Orijinal kare. (b) Belirlenen ten bölgeleri. (c) Nesnenin içinde bulunduğu dikdörtgen. (d) Yönelimi yönelim açısı yardımı ile düzenlenmiş nesne. (e) Nesneyi içeren yeniden boyutlandırılmış sınırlayan kare.

3.4 Ek Nesne Özellikleri

Özellik vektöründe sınırlayan karenin piksel değerleri yanında ek nesne özellikleri de kullanılmıştır. Bu özellikler arasında alan (*area*, (Umbaugh, 1998)), alan merkezi (*center of area*, (Umbaugh, 1998)), perimetre (*perimeter*, (Umbaugh, 1998)), Bölüm 3.3 ile anlatılan yönelim açısı ve çembersellik ($\frac{perimeter^2}{alan}$) vardır. Ek olarak, (3.14) ile verilen merkezden ortalama uzaklık (mean radial distance) çıkarılır.

$$\mu_r = \frac{1}{N} \sum_n \|(x_n, y_n) - (\bar{x}, \bar{y})\| \quad (3.14)$$

(3.14) de n piksel numarası, N piksel sayısı, (x_n, y_n) numarası n olan pikselin koordinatları, $(\bar{x}, \bar{y})$ alan merkezinin koordinatları ve $\| \cdot \|$ iki piksel arasındaki Öklit uzaklığıdır. Diğer bir özellik ise (3.15) ile verilen merkezden olan uzaklığın standart sapmasıdır (standart deviation of radial distance).

$$\sigma_R = \left(\left(\frac{1}{N} \right) \left(\sum_n (\|(x_n, y_n) - (\bar{x}, \bar{y})\| - \mu_R)^2 \right) \right)^{\frac{1}{2}} \quad (3.15)$$

Son nesne özelliği olarak (3.16) ile verilen çembersellik (circularity) değeri kullanılır.

$$C = \frac{\mu_R}{\sigma_R} \quad (3.16)$$

Özet olarak en küçük sınırlayan karenin 30x30 boyutlu özellik vektörüne 9 özellik daha eklenir.

3.5 Video Veritabanı

Eğitim ve test videoları bir adet Philips PCVC840K CCD web kamerası yardımı ile alınmıştır. Yakalama çözünürlüğü 320x240 değeri olarak ayarlanırken saniyede 15 kare yakalanmıştır. 29 Türk İşaret Dili Alfabesi harfinin (Şekil 1.1) her biri için 10 eğitim ve 6 test videosu kullanılmıştır. İşaretler 3 farklı işaretçi tarafından gerçekleştirilmiştir.

3.6 Sınıflama Karşılaştırması

Çıkarılan ikili özellikler farklı yedi sınıflama algoritması ile sınıflanmış ve algoritmaların başarı değerleri Çizelge 3.1 ile özetlenmiştir. Başarı yüzdesi $\frac{\text{başarılı sınıflama sayısı}}{\text{toplam test örneği sayısı}}$ olarak tanımlıdır.

Çizelge 3.1 Türk İşaret Dili Alfabesi parmak hecelemeleri üzerinde yedi sınıflama algoritmasının başarı yüzdeleri

Sınıflayıcı	Başarı Yüzdesi
1NN (Aha vd., 1991)	99.43
SVM (Keerthi vd., 2001)	98.85
Rastgele Orman (Random Forest, (Breiman, 2001))	97.13
RBF Ağı (RBF Network, (Fritzke, 1994))	96.55
MNB (John ve Langley, 1995)	88.51
J48 (Quinlan, Ross, 1993)	85.63

En başarılı sınıflayıcılar bir en yakın komşu (1NN, one nearest neighbor) ve destek vektör makinesi (SVM, support vector machine) olmuştur. Çizelge 3.1 ile gösterildiği gibi bu yöntemler %98 ve üzeri başarı sağlamışlardır. Sınıflamada zorluk çıkaran harflerin araştırılması sonucu en büyük problemin S/Ş ve G/Ğ harflerinin karıştırılması sonucu ortaya çıktığı görülmektedir. Şekil 3.3 ile gösterildiği gibi bu harflerin işaretleri birbirine çok benzemektedir.



(a)

(b)

(c)

(d)

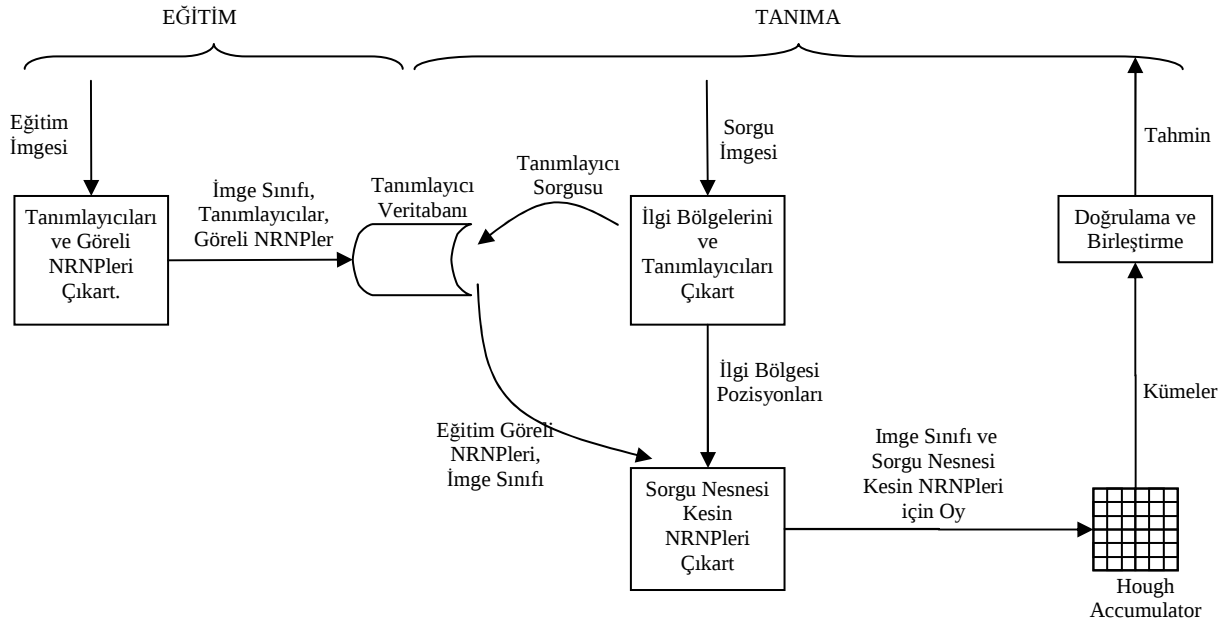
Şekil 3.3 Metodumuzun başarısız olabildiği iki örnek: (a) Ğ (b) G (c) Ş (d) S.

4. GENELLEŞTİRİLMİŞ HOUGH TRANSFORMU, İLGİ BÖLGELERİ VE YEREL TANIMLAYICILAR İLE TÜRK İŞARET DİLİ PARMAK HECELEMESİ TANIMA SİSTEMİ

Tez çalışmasının bu aşamasında Türk İşaret Dili parmak hecelemelerini Genelleştirilmiş Hough Transformu (Beinglass ve Wolfson, 1991), DoG ilgi bölgeleri (Lowe, 2004) ve SIFT (Lowe, 2004) tanımlayıcıları yardımı ile tanıyabilen bir sistem gerçekleştirilmiştir. Yaptığımız iyileştirmeler ile sistem yüzde 94 başarıya ulaşmıştır. Yaptığımız iyileştirmeler arasında daha basit ancak etkili bir oylama politikası kullanmak, Hough birikeci kova büyüklüklerini optimize etmek ve çok fazla noktayı elimine eden doğrulama adımlarından vazgeçmek vardır.

4.1 Genelleştirilmiş Hough Transformu ve İlgi Bölgeleri İle Nesne Tanımaya Ayrıntılı Bir Bakış

Bu bölümde Şekil 4.1 ile özetlenen, Genelleştirilmiş Hough Transformu, modern ilgi noktaları ve yerel tanımlayıcılar kullanan, katı-olmayan nesne sınıfı tanıma sisteminin gerçekleştirilmesi ayrıntıları ile anlatılmaktadır.



Şekil 4.1 Hough Transformu eğitim ve tanıma aşamalarının özeti. NRNP nesne referans noktası pozisyonu anlamına gelmektedir.

4.2 İlgi Bölgeleri ve Yerel Tanımlayıcılar

Hızlı bir Hough Transformu gerçekleştirmek için kullanılan ilgi bölgesinin kendi üzerinde bir koordinat sistemi oluşturmaya yetecek bilgiyi verebilmesi gerekmektedir. İlgi bölgeleri

üzerine bir karşılaştırma için (Mikolajczyk vd., 2005b) çalışmasına bakılabilir. Tez çalışmasının bu bölümünde ilgi bölgesi olarak DoG (Mikolajczyk vd., 2005b), (Lowe, 2004) ilgi bölgeleri kullanılmıştır.

İlgi bölgelerinin kullanıldığı bilgisayarla görü sistemlerinde birçok defa ilgi bölgelerinden yerel tanımlayıcılar çıkarılır. Mikolajik vd. çalışmaları arasında farklı yerel tanımlayıcıların (Mikolajczyk ve Schmid, 2005) ve diğer bir çalışmalarında ilgi bölgesi saptayıcıların ve yerel tanımlayıcıların beraber (Mikolajczyk vd., 2005a) karşılaştırmaları vardır. Bu çalışmada yerel tanımlayıcı olarak SIFT tanımlayıcısı (Lowe, 2004), (Mikolajczyk ve Schmid, 2005), (Mikolajczyk vd., 2005a) kullanılmıştır.

Lowe, (Lowe, 2004) tipik bir 500x500 boyutlarındaki imgeden yaklaşık 2000 ilgi bölgesinin çıktığını rapor etmektedir. Bizim çalışmamızda aşağıda anlatılan eleme yapıldıktan sonra imge başına yaklaşık 80 DoG bölgesi kalmaktadır. Bu ilgi bölgesi azlığı aşağıda anlatılacak farklı değer ve yaklaşımlar için temel bir nedendir.

4.3 Ten Rengi Tabanlı İlgi Bölgesi Eliminasyonu

Parmak hecelemeleri el ile yapıldığı için imgedeki el bölgeleri renk ile ten belirleme yöntemleri ile belirlenir ve sadece el bölgeleri içindeki ilgi bölgeleri kullanılır. Bölüm 3.2 ile anlatılan yöntem kullanılmıştır. Şekil 4.2 örnek bir renk ile ten bulma operasyonu sonucunu göstermektedir. Arka planda ten rengine yakın nesneler olduğunda bu kadar başarılı sonuçlar çıkmayabilmektedir.

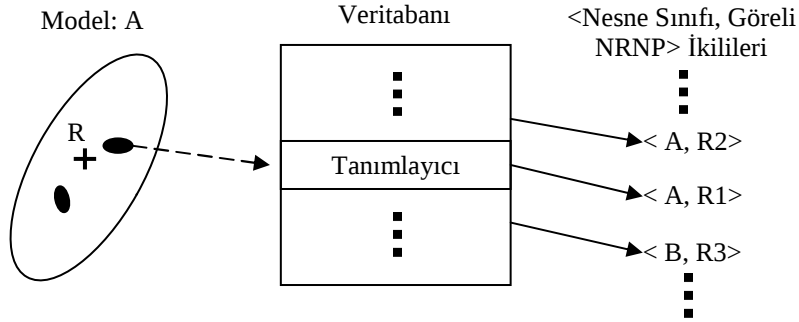


Şekil 4.2 Ten rengi ile el bölgesi çıkarma işlemi sonucu

4.4 Hough Transformu

Hough Transformu genel olarak bir imgede bulunabilecek doğru ve çemberler (Torii ve Imiya, 2007) gibi parametrik şekilleri aramak için kullanılır, ancak aslında bir imgede

bulunabilecek her türlü şekli aramak için kullanılabilir. Metot model ve imgede birbiri ile eşleşebilen yerel özellikleri bulmaya dayanır. Böyle eşleşebilen özelliklerin miktarı çok olduğunda imgenin aranan şeklin – modelin bir örneğini içermesi olasılığını yüksek olduğu açıktır. Ancak Hough Transformu bunun ötesine geçer: Eşleşen özelliklerin göreceli yerleşimleri de birbiri ile benzeşmelidir. Bu nesnedeki bir referans noktasının (reference point) göreceli yerinin eşleşen tüm noktalarda aynı olmasına dikkat edilerek sağlanır. Şekil 4.3 metodun eğitim aşamasını özetlemektedir.

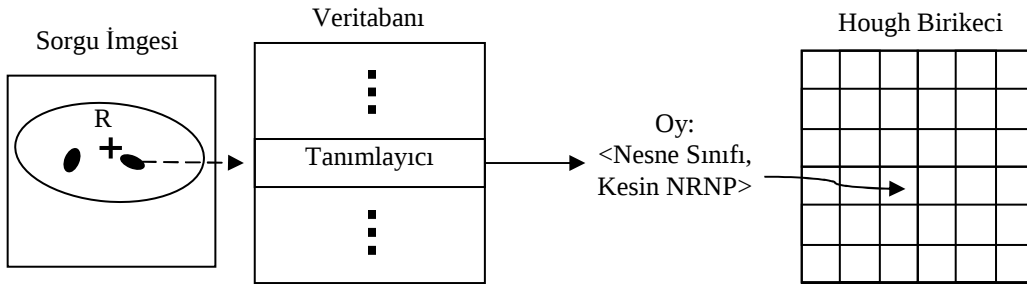


Şekil 4.3 Genelleştirilmiş Hough Transformu, ilgi bölgeleri ve yerel tanımlayıcılar ile şekil tanımda eğitim aşaması

Şekil 4.3'te en soldaki büyük elips "A" sınıfına ait bir modeli veya eğitim objesini göstermektedir. Küçük siyah dolgulu elipsler ilgi bölgelerini, mesela DoG bölgelerini göstermektedir. Küçük artı işareti nesne referans noktasını temsil etmektedir. Şeklin ortasındaki büyük dikdörtgen bir veritabanını göstermektedir. Eğitim aşamasında model ilgi bölgelerinden çıkarılan yerel tanımlayıcılar bu veritabanına aktarılır. Veritabanındaki her bir tanımlayıcı model sınıfına ve bu tanımlayıcının çıkarıldığı ilgi bölgesine göre referans noktasının göreceli yerine işaret etmektedir. Hough Transformunun eğitim aşamasının . (Beinglass ve Wolfson, 1991) çalışmasından adapte edilen sözde kodu (pseudocode) aşağıdadır:

- Her bir model için:
 - o Modelin üzerindeki ilgi bölgelerini (örneğin Harris, Hessian, DoG (Mikolajczyk vd., 2005b) ilgi bölgelerini) bulunuz. Her bir ilgi bölgesi üzerine yerel bir koordinat sistemi koymamıza elverecek bilgiyi içermelidir (örneğin yönelim, ölçek ve x-y pozisyonları gibi).
 - o İlgi bölgelerinden tanımlayıcıları çıkarınız (Örneğin SIFT (Mikolajczyk ve Schmid, 2005) tanımlayıcısını).
 - o Model üzerinde herhangi bir noktayı “nesne referans noktası” olarak atayınız.
 - o Model üzerindeki her bir ilgi bölgesi için:
 - İlgi bölgesinin üzerinde oluşturduğumuz koordinat sisteminde Nesne referans noktası pozisyonunu (NRNP) bulunuz.
 - Yerel tanımlayıcıyı model sınıfı ve görel NRNP ye işaret edecek şekilde bir veritabanına yerleştiriniz.

Şekil 4.4 Hough Transformu ile şekil tanıma yönteminin tanıma aşamasını özetlemektedir. En soldaki dikdörtgen bir sorgu/test imgesini temsil etmektedir. İçindeki büyük elips bulmak/tanımak istediğimiz bir nesneyi göstermektedir. Nesne içerisindeki küçük siyah dolgulu elipsler ilgi bölgelerini, artı işareti ise nesne referans noktasını göstermektedir. Şeklin ortasındaki büyük dikdörtgen eğitim aşamasında yerel tanımlayıcıların konulduğu veritabanını göstermektedir. En sağdaki grid şekli Hough Birikecidir. Bu birikeç obje sınıfları ve NRNP’ler için verilen oyları kümelemek için kullanılmaktadır. Tanıma aşamasında sorgu imgelerinden çıkarılan tanımlayıcılar veritabanındaki tanımlayıcılar ile eşlenir. Veritabanındaki eşlenen tanımlayıcılar nesne sınıfı ve nesne referans noktası pozisyonu ile ilgili oy verir. Bu oylar Hough birikecinde kümelenir.



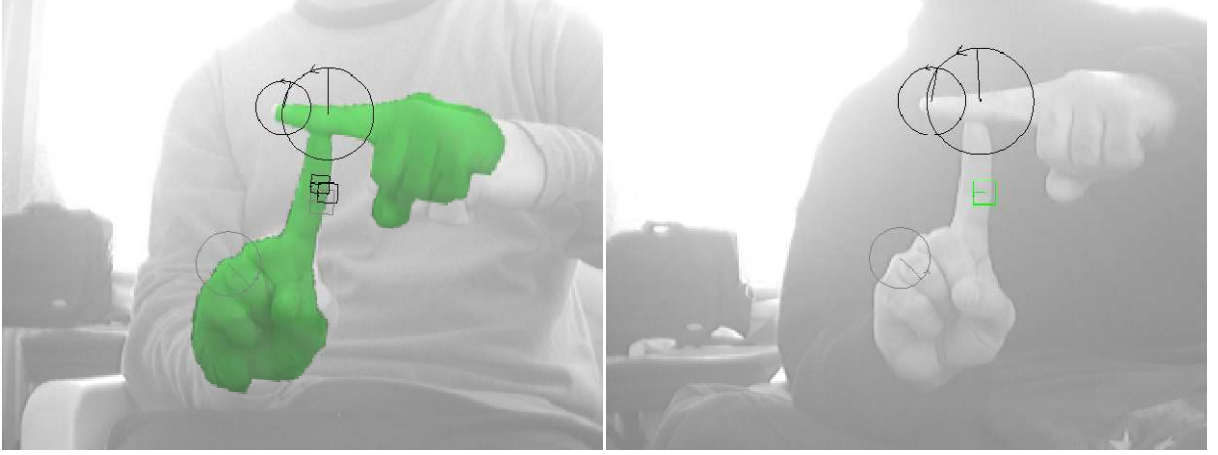
Şekil 4.4 Genelleştirilmiş Hough Transformu, ilgi bölgeleri ve yerel tanımlayıcılar ile şekil tanımda tanım aşaması.

Hough Transformu ile şekil tanımadaki tanıma aşamasının (Beinglass ve Wolfson, 1991) çalışmasından adapte edilen sözde kodu aşağıdadır:

- Her bir test imgesi için:
 - o İlgi bölgelerini bulunuz.
 - o İlgi bölgelerinden yerel tanımlayıcıları çıkarınız
 - o Her bir ilgi bölgesi için:
 - Veritabanındaki en benzer n tanımlayıcıyı bulunuz.
 - Bulunan her bir yerel tanımlayıcının işaret ettiği bir model sınıfı ve görelili NRNP ikilisi vardır. Sorgu imgesi ilgi bölgesi pozisyonlarından yararlanarak görelili NRNP'leri imge üzerinde kesin NRNP'lere dönüştürünüz. Nesne sınıfı, kesin NRNP ikilisini Hough birikecine bir oy olarak atınız.
 - o Hough birikeci kovalarını inceleyiniz. Herhangi bir kovada üç ya da daha fazla oy varsa bu oyları o model için bir hipotez kabul ediniz.
 - o Hipotezleri doğrulayınız, doğrulayamadığınız hipotezleri elimine ediniz.
 - o Doğrulanmış hipotezleri modelin şekildeki örnekleri olarak kabul ediniz.

Yerel bölgelerin pozisyonlarındaki büyük belirsizlik nedeni ile Lowe (Lowe, 2004) her bir oyu aslında her boyuttaki en yakın iki kovaya atmaktadır. Aynı yaklaşım bu çalışmada da kullanılmıştır, böylece her bir oy (x-y, ölçek ve açıdan ibaret olan dört boyut olması nedeni ile) 16 kovaya atılmaktadır.

Şekil 4.5 Hough Transformunun bazı bileşenlerini doğru bir eşleme üzerinde göstermektedir. Sol imge sorgu/test imgesidir. Sağ imge ise eşlenen eğitim imgesidir. Sol imgedeki yeşil kısım aslında sağ imgede bulunan eşlenen nesnedir. Eşlenen DoG bölgelerinin pozisyonları kullanarak dönüşümü gerçekleştirilmiştir. Sağ imgedeki küçük kare eğitim nesnesi referans noktasıdır. Sağ imgedeki üç küçük kare Hough birikecine konan SIFT tanımlayıcıları tarafından nesne referans noktası pozisyonu için atılan oylardır. Birbirlerine yakın oldukları için aynı kova içerisinde kümelenmişler ve dolayısı ile bu nesne hipotezini üretmişlerdir. Her bir imgedeki üç çember eşleşen SIFT noktalarının çıkartıldığı DoG ilgi bölgeleri, dolayısı ile oyların geldiği yerlerdir.



Şekil 4.5 Doğru bir eşlemenin sorgu ve eğitim imgeleri üzerinde gösterilen Hough Transformu bileşenleri

Şekil 4.6 Hough Transformu bileşenlerini yanlış bir eşleme üzerinde göstermektedir. Tıpkı Şekil 4.5 gibi okunabilir. Ayrıca bu şekil tanıma metodunun nasıl başarısız olabileceğini de göstermektedir. Eğitim nesnesi sorgu nesnesinin *bir bölümü* ile mükemmel bir şekilde uyushmaktadır, ancak sorgu nesnesinde eşlenmeyen kısımlar da kalmaktadır.



Şekil 4.6 Yanlış bir eşleme üzerinde Hough Transformu bileşenleri

4.5 Oylama tabanlı sınıflama

Bu bölümdeki uygulamada her bir test imgesinde sadece bir parmak hecelemesini varsayılabilir. Bu nedenle sorgu imgesi sınıfı bir oylama yöntemi ile tahmin edilebilir. Her bir Hough birikeci kovası hipotezi kendi sınıfı için oy kullanır. En çok oyu kullanan sınıf kazanır.

En çok oy alan sınıflar arasında bir beraberlik varsa sınıfların benzerlik miktarları kullanılır. Bir sınıfın benzerlik miktarı o sınıf için oy veren toplam benzerlikleridir. Kümelerin toplam benzerlikleri ise (4.1) ile verilir.

$$S = \prod_i \frac{1}{d_i + e} \quad (4.1)$$

(4.1) de i kümenin elemanlarından birini, d_i o eleman için veritabanındaki tanımlayıcı ile eşleşen model tanımlayıcısı arasındaki uzaklığı, e ise sıfırla bölmeyi engelleyen küçük bir sabit sayıyı göstermektedir.

4.6 Veritabanı, Protokol ve Optimizasyon

Bu bölüm için Şekil 1.1’da gösterilen 29 Türk alfabesi harfinin her birinden 16 örnek içeren bir veritabanı kullanılmıştır. Veritabanı eğitim, test ve doğrulama (validation) kümelerine ayrılmıştır. Eğitim kümesi Hough Transformunu eğitmek, doğrulama kümesi parametre değerlerinde optimizasyon yapmak ve test kümesi ise algoritmanın performansını ölçmek için kullanılmıştır.

4.7 Yapılan İyileştirmeler

Bu bölümde Genelleştirilmiş Hough Transformunun performansının iyileştirilmesi için yapılan optimizasyonlar ele alınacaktır. Mümkün olduğunda kendi yaklaşımımız ile diğerlerinin yaklaşımları tanıma başarısı açısından karşılaştırılacaktır. Yapılan deneylerin hepsinde ten bölgesi tabanlı ilgi bölgesi eleme ve oylama tabanlı sınıflama kullanılmıştır.

4.7.1 Oy Kullanan Veritabanı Eşleşmelerinin Sayısı

Genelleştirilmiş Hough Transformu ve ilgi bölgeleri ile şekil tanımda tanıma aşamasında her bir ilgi bölgesi için veritabanındaki benzer tanımlayıcılar bulunur ve benzer tanımlayıcılar için Hough Transformuna oylar atılır.

Bu adımda Lowe veritabanından en benzer iki tanımlayıcı alır ve eğer ancak ikinci ile birinci arasındaki fark yeterince büyükse birincinin oy atmasına izin verir. Bu yaklaşım bizim uygulamamızda 0.05 değerinde bir başarı oranını sonuç vermektedir.

Bu çalışmadaki uygulama için sabit sayıda (4 tane) tanımlayıcıya oy verdirtmenin daha iyi bir sonuç verdiği görülmüştür. Bu yaklaşım ile başarı oranı 0.17 ye çıkmaktadır.

4.7.2 Hough Birikeci Kova Boyutu

Optimizasyon sırasında Hough birikeci kova boyutunun optimize edilmesinin önemli olduğu görülmüştür.

Bu çalışmadaki kova büyüklükleri diğer çalışmalar için en iyi değerler olmayabilir, ancak

optimizasyonun sonuca olan etkisi bu değerlerin optimizasyonunun önemini göstermektedir.

Bu çalışmada Hough birikeci kovalarının x-y boyutlarındaki büyüklükleri o boyuttaki arama bölgesinin boyutu cinsinden ifade edilmektedir. Arama bölgesi ten bölgesini sınırlayan en küçük dikdörtgen alandır. Arama bölgesinin Lowe'nin kullandığı $1/4$ ü yerine $1/15$ i gibi küçük kova boyutları kullanmak tanıma başarısını 0.05 ten 0.19 a çıkarmaktadır.

Açı boyutunda $2\pi/6$ yerine $2\pi/14$ gibi daha küçük bir kova boyutu kullanma başarıyı 0.05 ten 0.10 a çıkarmaktadır.

Ölçek boyutunda 2 yerine 0.75 gibi küçük bir kova boyutu kullanma başarıyı 0.05 ten 0.07 ye çıkarmaktadır.

4.7.3 Doğrulama Adımları

Lowe (Lowe, 2004)Hough birikeci kovalarındaki kümelerin oluşturulmasından sonra iki doğrulama adımı kullanmaktadır.

Birinci doğrulama adımında, bir döngü içerisinde, model ilgi bölgelerinin imge düzlemine afin dönüşümü bir en az kareler (least squares) yaklaşımı ile bulunur, dönüşüm gerçekleştirilir, önceki döngü iterasyonundaki pozisyon hata payının yarısından daha fazla pozisyon hatasına sahip olan ilgi bölgeleri elenir (bu Hough Transformu kova boyutunu her iterasyonda yarıya düşürme gibi görülebilir), afin dönüşüme uyan yeni ilgi bölgeleri varsa onlar tekrar iyi ilgi bölgelerinin arasına eklenir. Bu adım bu çalışmada anlatılan uygulamada çok fazla kümeyi elimine etmektedir. Bu doğrulama adımının kaldırılması ile başarı 0.05 değerinden 0.44 değerine yükselmektedir.

İkinci doğrulama adımı (Lowe, 2004) de verilmiştir. Beklenen yanlış eşleme ve gerçekleşen eşleme miktarlarına dayanan olasılık tabanlı bir yöntemdir. Bu doğrulama adımı da bu çalışmada anlatılan uygulamada çok fazla kümeyi elimine etmektedir. Öyle ki, bu adımın kullanılmaması ile başarı oranı 0.05 değerinden 0.39 değerine yükselmektedir.

Bu doğrulama adımları imgelerden üretilen ortalama ilgi bölgesi sayısı çok olduğunda (yaklaşık 2000) ve Hough birikecinde geniş kovalar kullanıldığında faydalı olabilir. Bu çalışmada anlatılan uygulamada ten bölgesi tabanlı ilgi bölgesi eleme sonucunda elde imge başına ortalama 80 ilgi bölgesi kalmaktadır. Ayrıca daha dar Hough birikeci kovaları kullanılmakta, dolayısı ile ekstra doğrulama adımlarına olan gereksinim azalmaktadır.

4.7.4 Tüm Optimizasyon Sonuçlarının Birleştirilmesi

Bu bölümde anlatılan tüm optimizasyon adımları kullanıldığında, yani veritabanında en

benzer 4 tanımlayıcı oy kullandığında, x-y boyutlarında arama bölgesinin 1/15 i büyüklüğünde, açı boyutunda $2\pi/14$ büyüklüğünde ve ölçek boyutunda 0.75 büyüklüğünde Hough birikeci kovaları kullanıldığında ve yukarıda anlatılan iki doğrulama adımı kullanılmadığında Türk Alfabesi harfleri işaret dili parmak hecelemesi tanıma uygulamasının başarısı 0.94 değerine ulaşmaktadır. Çizelge 4.1 toplamda 145 test imgesinden 8 tanesinin yanlış tanındığını göstermektedir. Boş sınıflamaların, sınıflayıcının bir karar vermediği durumların, sayısı sıfırdır, yani her test örneği için bir karar verilmiştir. Bunun nedeni fazla miktarda hipotez elimine eden doğrulama adımlarının kullanılmamış olmasıdır.

Çizelge 4.1 Tüm optimizasyonlar bir arada kullanıldığında doğru, yanlış ve boş sınıflamaların sayısı

Doğru sayısı	Yanlış sayısı	Boş sayısı	Başarı oranı
137	8	0	0.94

4.8 Yanlış Sınıflanan Parmak Hecelemeleri

Şekil 4.7 karıştırılan sınıfların yan yana konulmuş rastgele birer örneğini göstermektedir. Görüldüğü gibi karıştırılan sınıflar birbirine çok benzemektedir. Ç ve D sadece alttaki elin küçük bir hareketi ile ayrılırken diğerleri sadece birkaç parmak ile ayrılmaktadır. Bu parmaklar tüm alana kıyasla çok küçük bir alanı kapsamaktadır.



a) Ç ve D



b) G ve Ğ



c) M ve N



d) S ve Ş

Şekil 4.7 Karıştırılan sınıflardan örnekler

Çizelge 4.2 bir *kısmi sınıf karıştırma matrisi* vermektedir. Kısmi sınıf karıştırma matrisi normal sınıf karıştırma matrisinden, yerden kazanmak ve daha kolay anlaşılır bir çizelge elde etmek amacı ile hatalı sınıflama göstermeyen satır ve sütunlar silinerek elde edilmektedir.

Çizelge 4.2 Deneyden çıkarılan kısmi sınıf karıştırma matrisi. Test kümesinde her sınıftan 5 örnek vardır.

		Tahmin Edilen Sınıflar				
		D	G	Ğ	N	S
Gerçek Sınıflar	Ç	1	-	-	-	-
	G	-	-	3	-	-
	Ğ	-	1	-	-	-
	M	-	-	-	2	-
	Ş	-	-	-	-	1

5. YEREL İLGİ BÖLGELERİNİN KATI-OLMAYAN NESNE SINIFI TANIMA AÇISINDAN BİR KARŞILAŞTIRMASI

Tez çalışmasının bu aşamasında önceki aşamada anlatılan Genelleştirilmiş Hough Transformu ve ilgi bölgeleri ile Türk İşaret Dili parmak hecelemelri tanıma sistemindeki önemli bir bileşen olan ilgi bölgelerinin bir değerlendirilmesi yapılmıştır.

5.1 Karşılaştırılan İlgi Bölgesi Bulucuları

Değerlendirmede aşağıdaki ilgi bölgesi bulucuları karşılaştırılmıştır. Karşılaştırılan her bir ilgi bölgesi bulucu için gerekli kodlar dipnotlarda belirtilen web adreslerinden temin edilerek uygulamada kullanılmıştır.

HarLap (Mikolajczyk ve Schmid, 2005),(Mikolajczyk ve Schmid, 2001): Harris-Laplace bölgeleri ölçek uzayında hem x-y düzleminde Harris fonksiyonunun (Harris ve Stephens, 1988)hem de ölçek doğrultusunda LoG (Lindeberg, 1998) fonksiyonunun maksima (en yüksek değer) yaptığı noktalar tarafından belirlenir. Çalıştırılabilir dosya http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/extract_features2.tar.gz web adresinden alınmıştır.

HarAff (Mikolajczyk ve Schmid, 2004): Harris-Affine bölgeleri önce Harris-Laplace bölgelerinin bulunması, daha sonra bu bölgelere bir afin adaptasyon işlemi (Lindeberg ve Garding, 1997) uygulanması sonucu elde edilir. Çalıştırılabilir dosya http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/extract_features2.tar.gz web adresinden alınmıştır.

HesLap (Mikolajczyk ve Schmid, 2005),(Lowe, 2004) ,(Mikolajczyk ve Schmid, 2004): Hessian-Laplace bölgeleri x-y düzleminde Hessian determinant değerinin, ölçek doğrultusunda ise LoG (Lindeberg, 1998) fonksiyonunun maksima yaptığı noktalar ile belirlenir. Çalıştırılabilir dosya http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/extract_features2.tar.gz web adresinden alınmıştır.

HesAff (Mikolajczyk vd., 2005b): Hessian-Affine bölgeleri önce Hessian-Laplace bölgelerinin bulunması, daha sonra bu bölgelere bir afin adaptasyon işlemi (Lindeberg ve Garding, 1997) uygulanması sonucu elde edilir. Çalıştırılabilir dosya http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/extract_features2.tar.gz web adresinden alınmıştır.

EdgeLap (Mikolajczyk vd., 2006),(Mikolajczyk vd., 2003): Edge-Laplace bölgeleri çok ölçekli (multi-scale) Canny kenar bulma algoritmasının çıktısı kullanılarak elde edilir. Kenar imgesindeki her kenar pikseli bir x-y düzleminde bölge merkezi adaydır. Bu adaylar arasında ölçek doğrultusunda Laplacian değerinin belirgin şekilde extrema (en düşük veya en yüksek değerler) yaptığı noktalar kabul edilir. Laplace extremum (en düşük veya en yüksek değer) değeri aynı zamanda o bölgenin ölçeği kabul edilir. Çalıştırılabilir dosya http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/extract_features2.tar.gz web adresinden alınmıştır.

IBR (Tuyletaars ve Van Gool, 2000),(Tuyletaars ve Gool, 2004): Parlaklık Tabanlı Bölgeler (Intensity Based Regions) bölgeleri parlaklık (intensity) extremasından çıkıp her yöne ilerleyen doğrular üzerinde parlaklığı inceleyerek bulunur. Parlaklıkta ani bir değişim o yönde bölgenin sonunu ipucu verir. Bu bölge sonunu belirleyen noktalar kapalı bir alan belirleyecek bir şekilde birbiri ile birleştirilir ve daha sonra bir elipse dönüştürülür. Çalıştırılabilir dosya http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/ibr.ln.gz adresinden alınmıştır.

SR (Kadir vd., 2004): Belirgin bölgeler (Salient Regions) elipsoidal bölgelerde önce parlaklık değerlerinin olasılık yoğunluk fonksiyonunun (pdf, probability density function), sonrada olasılık yoğunluk fonksiyonlarının entropisinin hesaplanması ile elde edilir. Ölçek doğrultusunda entropi extreması aday kabul edilir. Ölçek doğrultusunda daha büyük bir entropi türevine sahip olan noktalar Belirgin bölge olarak kabul edilir. Çalıştırılabilir dosya <http://www.robots.ox.ac.uk/~timork/salscale.html> adresinden alınmıştır.

MSER (Matas vd., 2002),(Mikolajczyk vd., 2005b): Maksimum Kalıcı Extrama Bölgeleri (Maximally Stable Extremal Regions) bölgeleri parlaklık imgesinden su dökümü (watershed) algoritması yardımı ile elde edilen birleşik bileşenlerdir (connected components). Sadece geniş bir parlaklık eşik değeri aralığında kalıcı kalabilen bölgeler kabul edilmektedir. Çalıştırılabilir dosya http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/msr.tar.gz adresinden alınmıştır.

PCBR (Deng vd., 2007): Ana Kavis Tabanlı Bölgeler (Principal Curvature Based Regions) MSER bölgelerine su dökümü algoritması yardımı ile bulunan bölgeler olmaları bakımından benzerler, ancak *ana kavis imgelerinden* elde edilirler. Ana kavis imgeleri her bir pikselde Hessian matrisinin özdeğerlerinin (eigenvalues) maksimasını alarak elde edilir. Kalıcılık ölçek aralıklarında beklenir. Çalıştırılabilir dosya

<http://web.engr.oregonstate.edu/~tgd/software/pcbrRun.zip> adresinden alınmıştır

DoG (Lowe, 2004): Gaussyanların Farkı (Difference of Gaussians) bölgeleri LoG fonksiyonunun hem x-y düzleminde hem de ölçek doğrultusunda extrema yaptığı noktalar ile belirlenir. LoG fonksiyonu işlemin hızlandırılması amacı ile Gaussyanların Farkı yöntemi ile elde edilir. Çalıştırılabilir dosya <http://www.cs.ubc.ca/~lowe/keypoints/> adresinden alınmıştır

kAS (Ferrari vd., 2006b) (Ferrari vd., 2006a),(Martin vd., 2004): K Bitişik Segment (k Adjacent Segments) yaklaşık olarak düz kenar parçalarının (contour segments) kümeleridir. K kümedeki parça sayısını verir. Tekil parçalar 1AS, L gibi şekiller 2AS, Z ve F gibi şekiller ise 3AS olarak kabul edilir. kAS bölgelerinin merkezi parça merkezlerinin ortalaması, ölçeği ise birbirine en uzak parçalar arası uzaklıktır. Çalıştırılabilir dosya http://www.vision.ee.ethz.ch/~calvin/kas_sources_V1.03.tgz adresinden alınmıştır.

SURF (Bay vd., 2008): Hızlandırılmış Gürbüz Özellik (Speeded Up Robust Feature) bölgeleri Hessian matrisinin determinantının hem x-y düzleminde hem de ölçek doğrultusunda extrema yaptığı noktalar tarafından belirlenir. Hessian determinantının hızla elde edilebilmesi için toplama imgeleri (integral images) ve kutu filtreleri (box filters) tabanlı bir Hessian determinantı yakınsaması kullanılır. Çalıştırılabilir dosya <http://www.vision.ee.ethz.ch/~surf/index.html> adresinden alınmıştır.

FAST (Rosten ve Drummond, 2005), (Rosten ve Drummond, 2006): Hızlandırılmış Parça Testinden Özellikler (Features from Accelerated Segment Test) belirleyicisi de hız için tasarlanmış bir belirleyicidir. İlgili noktaları bir parça (segment) testi ile bulunur: Verilen bir pikselin ilgili noktası olup olmadığına karar vermek için bu pikselin etrafındaki bir çember üzerindeki pikseller kontrol edilir. Eğer n bitişik piksel (bir parça) merkez pikselin parlaklığından belli bir eşik değeri kadar farklı ise merkez piksel bir ilgili noktası olarak kabul edilir. Algoritmanın ilk versiyonlarında n değeri 11 den fazla olmalıdır. Daha sonraki versiyonlarında makine öğrenmesi teknikleri (ID3 (Quinlan, J. R., 1986)) kullanılarak bu kısıt kaldırılmıştır. İlgili bölgesi olmayan noktaların daha hızlı elenmesi için bazı akıllı kontroller uygulanır. FAST belirleyicisi ölçek bilgisi vermemektedir. Bu çalışmada kullanılan Genelleştirilmiş Hough Transformu ile şekil tanıma uygulamasında ölçek bilgisi zorunlu olduğundan FAST belirleyicisi tarafından verilen noktalar üzerinden Laplasyan ölçek çıkarımı (Lindeberg, 1998), (Mikolajczyk ve Schmid, 2001) yapılmıştır. Çalıştırılabilir dosya <http://mi.eng.cam.ac.uk/~er258/work/fast.html> adresinden alınmıştır.

5.2 Ayırt Edicilik

Farklı bölge belirleyicilerin değerlendirilmesinde ilk akla gelen kriterlerden biri tanıma başarısıdır. Ancak bir tanıma uygulamasında kullanılan belirleyicinin en yüksek tanıma başarısını en az bölge sayısı ile ulaşması gerekmektedir. Başka bir deyişle bölgeler olabildiğince “ayırt edici” olmalıdır. Çünkü ilgi bölge belirleme işlemi genelde sistemin çalışma zamanının ön sıralarında gerçekleşir ve eğer çok fazla sayıda bölge belirlenirse bu bölgelerin kullanıldığı diğer aşamalar daha yavaş çalışır. Gereksiz bölgeler tanıyıcı algoritmayı şaşırtarak başarının düşmesine de neden olabilir.

İdeal ilgi bölgesi bulucusu sadece “gerekli” veya “ayırt edici” bölgeleri bulur.

Her bir ilgi bölgesi bulucusu farklı sayıda bölge bulduğundan ve bölge sayısı başarıyı çok etkilediğinden dolayı ayırt edicilik tanıma başarısı ve ortalama ilgi bölgesi sayısı ile belirlenemez. Bu çalışmada ayırt edicilik rastgele sayıların başarısına kıyasla tanımlanmıştır.

Ayırt edicilik (5.1) ile tanımlanır:

$$q = \frac{d}{r} \quad (5.1)$$

(5.1) denkleminde q ayırtedicilik, d ilgi bölgesi belirleyicisinin başarısı, r ise rastgele bölge üreticinin (aynı sayıda rastgele ilgi bölgesi için) başarısıdır. Rastgele bölge üreticisinin ortalamada ilgi bölgesi belirleyicisinin bölge sayısı kadar bölge üretmesi sağlanır. Rastgele üreticinin başarısı her çalıştırmada değişebileceğinden ((Nowak vd., 2006) takip edilerek) 6 çalıştırmanın ortalaması kullanılır. Bu çalışma için geliştirilen ExpRand rastgele bölge üreticisinin standart sapmaları Çizelge 5.1 de görülebilir.

Çizelge 5.1 Ortalama bölge sayısına kıyasla rastgele bölgelerin (ExpRand bölgelerinin) başarısı. Üçüncü sütun rastgele bölgelerin standart sapmasını vermektedir. Ortalama ve standart sapmalar 6 çalıştırma sonucunda elde edilmiştir.

Ortalama Bölge Sayısı	ExpRand Başarısı Ortalaması	ExpRand Başarısı Standart Sapması
37	0.0000	0.0000
67	0.0046	0.0056
325	0.7057	0.0438
384	0.7575	0.0384
428	0.8195	0.0452
444	0.8287	0.0379
482	0.8322	0.0399
555	0.8471	0.0450
862	0.8759	0.0265
875	0.8575	0.0369
876	0.8736	0.0307
1577	0.8897	0.033
1658	0.8713	0.032
2282	0.8667	0.0113

Tanım gereği rastgele bölge üreticisinin ayırt ediciliği 1'e eşittir.

Ayırt edicilik bir bölge belirleyicisinin kör bir rastgele üreticiye kıyasla ne kadar iyi olduğunu göstermektedir. İyi bir belirleyicinin en azından rastgele üreticiyi geçmesini, dolayısı ile ayırt edicilik değerinin 1'in üzerinde olmasını bekleriz. Tersine ayırt edicilik değerinin 1'in altında olması belirleyicinin kör bir rastgele üreticiden başarısız olması anlamına gelir ki, bu da belirleyicinin bu tip bir uygulamada kullanılmasını sorgulanabilir hale getirir.

Dikkat edilmesi gereken bir nokta rastgele üreticinin çok az sayıda bölge ürettiği, bu nedenle başarısının çok düşük kaldığı ve hatta sıfır olduğu durumlarda "ayırt edicilik" istatistiğine değer verilmemesi gereğidir. Rastgele bölge üreticilerinin başarısı küçük bölge sayılarında ilgi bölgesi belirleyicilerden daha hızlı azalmaktadır. Bu ise belirleyicinin ayırt edicilik değerini aşırı artırmaktadır. Bunun bir örneği IBR ve PCBR belirleyicilerinin ayırt edicilik değerlerinde görülebilir.

5.3 ExpRand: Eksponensiyal Rastgele Bölge Üreticisi

ExpRand rastgele bölge üreticisi bölge üretme işlemine rastgele bir s ölçeği üretmekle başlar. Bunun için (5.2) kullanılır.

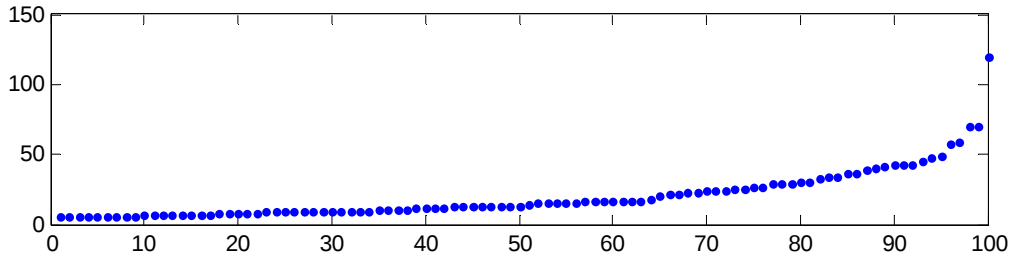
$$s = s_{min} + rand \quad (5.2)$$

(5.2) de s_{min} en küçük kabul edilebilir ölçek, $rand$ ise (5.3) ile verilen Eksponensiyal olasılık dağılımı ile üretilen rastgele bir sayıdır.

$$f(x|\mu) = \begin{cases} \frac{1}{\mu} e^{-\frac{x}{\mu}}, & x \geq 0 \\ 0, & \text{aksihalde} \end{cases} \quad (5.3)$$

Her ne kadar (5.2) s_{min} ile sonsuz arasında ölçek değerleri verse de, büyük ölçekler çok küçük olasılıklara sahiptir, dolayısı ile nadir olarak gelmektedirler. Eğer imge boyutlarına sığmayacak büyüklükte bir ölçek gelirse kullanılmaz. s_{min} değeri için 5, μ değeri için ise 15 seçilmiştir. Bu değerler rastgele üreticinin ölçek çıktılarını MSER ve HesLap gibi belirleyicilere benzetmek için gözlem yoluyla seçilmiştir, daha sonraki versiyonlarda bu değerler imge boyutları yardımı ile bulunacaktır.

Şekil 5.1 Bu şekilde üretilen 100 adet ölçeğin dağılımını göstermektedir.



Şekil 5.1 ExpRand tarafından üretilen 100 adet rastgele ölçek. Ölçekler grafiğe aktarılmadan önce dağılımın anlaşılabilir hale gelmesi için sıralanmıştır. Yatay eksen ölçek numarasını, dikey eksen ise ölçeği vermektedir

ExpRand düzgün olasılık dağılımından (Walpole vd., 2002) bir x koordinatı ve yine düzgün olasılık dağılımından bir y koordinatı üreterek işlemi bitirir.

Bu yöntem yardımı ile herhangi bir imge için istenen her sayıda bölge üretilebilir.

ExpRand üreticisi tasarlanırken ayrık (discrete) dağılımlar yerine sürekli (continuous) olasılık dağılımları tercih edilmiştir çünkü bu bölümde kullanılan tanıma algoritmasında kullanılan ölçek, x ve y değerleri sürekli değerler almaktadır.

5.4 Veritabanı ve Protokol

Bu çalışma için Bölüm 3.5 da anlatılan video veritabanından Bölüm 3.1 ile anlatılan yöntemle çıkarılan anahtar kareler veritabanı olarak kullanılmıştır. Her bir sınıf için altı eğitim örneği kullanılırken beş test örneği seçilmiştir.

5.5 Başarı Oranı

Nesne sınıfı tanıma sistemimizin çıktıları doğru, yanlış ve boş olarak sınıflandırılabilir. Doğru sınıflandırma sistemin sorgu imgesi ile aynı sınıfı vermesi, yanlış sınıflandırma sistemin farklı bir sınıf vermesi, boş sınıflandırma ise sistemin herhangi bir sınıfa karar vermemesi durumlarında oluşmaktadır. Başarı oranı olarak (5.4) ile verilen oran kullanılmıştır.

$$B = \frac{t - f}{a} \quad (5.4)$$

(5.4) denkleminde t doğru sınıflamaların sayısı, f yanlış sınıflamaların sayısı, a ise tüm sınıflama denemelerinin sayısıdır. Daha çok kullanılan $\frac{t}{a}$ yerine (5.4) denkleminin tercih edilmesinin nedeni (5.4) denkleminin boş sınıflamaları yanlış sınıflamalara tercih etmesidir.

5.6 Değerlendirme ve Sonuçlar

Bir ilgi bölgesi belirleyicisini değerlendirmek için o belirleyici ile elde edilen bölgeler Bölüm 4 ile anlatılan Genelleştirilmiş Hough Transformu ve ilgi bölgeleri ile şekil tanıma sisteminde ilgi bölgesi olarak kullanılır ve sistem tanıma başarı oranı (5.4) denklemindeki gibi hesap edilir. Bu sırada belirleyicinin imge başına ortalama kaç bölge çıkardığı da kaydedilir. Daha sonra ExpRand rastgele bölge üreticisi bölge belirleyici ile aynı sayıda rastgele bölge üretir ve bu bölgeler sistemde ilgi bölgesi olarak kullanılarak bir başarı oranı elde edilir. ExpRand başarı oranları 6 bağımsız çalıştırma sonrasında elde edilir. Belirleyici ve ExpRand başarı oranları kullanılarak belirleyicinin ayırt edicilik değeri (5.1) denklemindeki gibi bulunur.

Çizelge 5.2 belirleyici isimlerini, belirleyicilerin başarı oranlarını, belirleyicilerin ayırt edicilik değerlerini ve belirleyicinin imge başına belirlediği ortalama bölge sayılarını listelemektedir. İyi bir belirleyicinin hem yüksek başarı oranına hem de yüksek ayırt edicilik değerlerine sahip olması beklenir. EdgeLap, DoG ve SURF böyle belirleyicilerdir.

Çizelge 5.2 Farklı ilgi bölgesi belirleyicileri kullanılarak yapılan nesne sınıfı tanıma denemelerinin sonuçları

Belirleyici	Belirleyici Başarı Oranı	Ayır Edicilik	Ortalama Bölge Sayısı	Ortalamada Bir İmgeyi İşleme Süresi (saniye)
Edgelap	0.8897	1.000	1577	1.38
Surf	0.8690	1.1472	384	2.75
Dog	0.8552	1.2117	325	0.51
Mser	0.8207	0.9688	555	0.19
Haraff	0.8207	0.9370	862	0.69
3As	0.8069	0.9237	876	29.14
Harlap	0.7931	0.9249	875	0.46
SR	0.7724	0.9425	428	1.14
Fast	0.6897	0.7916	1658	0.48
4As	0.6690	0.7719	2282	33.38
Hesaff	0.6069	0.7323	444	0.61
Heslap	0.5862	0.7044	482	0.44
Ibr	0.3724	Sonsuz	37	0.15
Pcbr	0.0138	3.0000	67	0.55

Çizelge 5.3’de EdgeLap ve 1577 bölgeyi ExpRand’ın başarı oranı bakımından başta geldiği görülmektedir. Bu EdgeLap’ın ayır ediciliğini 1 e eşit yapmaktadır. ExpRand ile EdgeLap arasındaki berabere kalma durumunu bozma daha sonraki çalışmalara bırakılmıştır.

Başarı oranında ikinci ve üçüncü yerler SURF ve DoG belirleyicilerine kalmıştır. Bu belirleyiciler görece olarak az sayıda bölge ile oldukça iyi tanıma başarısına ulaşmaktadırlar. Az sayıda bölge daha sonra bu bölgeleri kullanan algoritmaların hızlı çalışması anlamına gelmektedir. Bu belirleyicilerin kendileri de hızlı çalışmak üzere dizayn edilmişlerdir. Özellikle DoG belirleyicisinin en hızlı belirleyicilerden olduğu görülmektedir.

Hessian-Laplace, Bölüm 2.4 ile anlatılan literatürdeki diğer çalışmalarda önlerde görülürken, bu çalışmada arka sıralarda kalmaktadır. Bu veritabanında yeterince bölge belirleyemediği gibi bölgeleri de yeterince ayır edici değildir. Bu durum (Nowak vd., 2006) çalışmasında verilen bir argümanını desteklemektedir: bir rastgele bölge üreticisi bir belirleyiciye tercih

edilebilir, çünkü veritabanından bağımsız olarak gerektiğinde bölge üretebilir. Aksine bir belirleyici bir veritabanında yeterince bölge belirleyebilirken bir başkasında yeterince bölge belirleyemeyebilir.

MSER, Bölüm 2.4 ile anlatılan literatürdeki diğer nesne sınıfı tanıma karşılaştırmalarında arka sıralarda görülmekte iken, bu çalışmada yapılan karşılaştırmada önlerde yer almaktadır. MSER gerçekleştirmesinin geçerli ayarlarının belirleyicinin tanım kapasitesini gizlediği görülmektedir. Ayarlar belirlenen bölge sayısını artıracak şekilde düzenlendiğinde çok daha iyi sonuçlar elde edilebilmektedir. Ayrıca MSER çok hızlı bir şekilde çalışmaktadır.

Ayırt edicilik açısından bakıldığında sadece SURF ve DoG özelliklerinin gerçekten yüksek değerler aldığı, EdgeLap özelliklerinin ise ancak rastgele bölgeler kadar ayırt edici olduğu görülmektedir. PCBR ve IBR bölgelerinin ayırt edicilik değerlerini dikkate almamak gerekmektedir, çünkü o bölge sayılarında rastgele sayıların başarısı kıyaslama yapmak için çok düşüktür.

Genel olarak başarı oranı artan bölge sayısı ile artmaktadır. Ancak bu durum belli bir optimal bölge sayısına kadar devam etmektedir, bu sayıdan sonra başarı oranı düşmeye başlamaktadır. Bu durum Çizelge 5.2’de belirleyiciler 4AS ve FAST için ve Çizelge 5.1’de rastgele bölge üretici ExpRand için gözlenebilir.

Çok düşük bölge sayılarında hem bölge belirleyicilerinin (örneğin IBR ve PCBR’nin) hem de rastgele bölge üreticilerin başarısı düşmektedir, rastgele bölge üreticiler daha hızlı olmak üzere. Bu durum da yine Çizelge 5.1 ve Çizelge 5.2 ile görülmektedir.

Bölge belirleyicilerin bir imgeyi işleme zamanları çizelgede verilmiştir. kAS belirleyicisinin diğerlerine kıyasla çok yavaş olduğu görülürken MSER belirleyicisinin hızlı olduğu görülmektedir. Çalışma hızı ile ilgili ölçümler sadece bir fikir vermek amacı ile konulmuştur. Bazı gerçekleştirmeler C veya MATLAB gibi farklı dillerdedir, bazı gerçekleştirmeler Microsoft Windows veya Linux gibi farklı işletim dillerine yöneliktir ve bazı gerçekleştirmeler bölge belirlemenin yanında başka işler (örneğin tanımlayıcıları çıkarmak) yapmaktadır. Ayrıca her bir belirleyici farklı sayıda bölge belirlemektedir. Özetle çalışma zamanlarının doğru bir şekilde değerlendirilmesi zordur.

6. SONUÇLAR VE ÖNERİLER

Bu tez kapsamında birinci aşamada Bölüm 3 ile anlatılan en az eylemsizlik eksenli tabanlı hızlı hizalama ile Türk İşaret Dili parmak hecelemleri tanıma sistemi gerçekleştirilmiştir. Bölüm 3.5 ile anlatılan veritabanı üzerinde yapılan denemelerde sistemin yüzde 99 üzerinde başarı sağladığı görülmektedir. Bu hizalama yöntemi benzer yönelimlere sahip nesneleri aynı hizaya getirirken büyük yönelim farkına sahip nesneleri farklı hizalara getirmektedir. Bu istenen bir sonuçtur, çünkü parmak hecelemleri tanıma probleminde, örneğin U ve C harfleri için, yönelim temel ayırıcı olabilir. Sistem tüm video yerine sadece bir kare kullanması, hızlı hizalama ve sınırlayan karenin küçük çözünürlüğe indirgenmesi nedeni ile çok hızlı çalışabilmektedir. Farklı sınıflama algoritmalarının başarısı çıkarılmış ve bir en yakın komşu (1NN, (Aha vd., 1991)) ve destek vektör makinesi (SVM, (Keerthi vd., 2001)) algoritmalarının en başarılı algoritmalar olduğu görülmüştür. Her ne kadar hızlı hizalama yöntemi el şekli tanıma bağlamında tanıtılsa da yöntem diğer şekil tanıma problemlerinde kullanılabilir.

Tez çalışmasının ikinci aşamasında Bölüm 4 ile anlatılan Genelleştirilmiş Hough Transformu, DoG ilgi bölgeleri ve SIFT yerel tanımlayıcıları kullanan Türk işaret dili harf parmak hecelemleri tanıma sistemi gerçekleştirilmiştir. Bu çalışmada Hough Transformu tabanlı tanımda algoritma parametrelerinin iyi optimize edilmesinin başarıya büyük etki yaptığı gözlemlenmiş ve önemli parametreler ve bu parametrelerin bizim uygulamamızda aldığı optimum değerler verilmiştir. Sistem yüzde 94 üzeri başarı sağlamıştır.

Daha başarılı bir Hough Transformu gerçeklemek için daha karmaşık sıfır-veya-bir oy şemasının yerine sabit sayıda (örneğin 4) oy şeması kullanılmıştır. Bu yaklaşım eğer uygulamanız daha fazla sayıda doğru sınıflama karşılığında birkaç yanlış sınıflamayı kompanse edebiliyor ise daha iyi çalışır. Böyle bir uygulamaya örnek olarak el şekillerini Gizli Markov Modelleri (Hidden Markov Models, (Xie vd., 2004)) yardımı ile birleştirerek el hareketi tanımaya çalışan bir sistem verilebilir.

Optimizasyon çalışmalarının sonucunda küçük Hough birikeci kovaları kullanılmıştır. Bu sistemin başarısında büyük bir fark oluşturmakta ve ekstra doğrulama adımlarını gereksiz kılmaktadır.

İlgi bölgesi başına birden fazla oy atmanın ve daha küçük kovalar kullanmanın bir yan etkisi algoritmanın çalışma zamanının artmasıdır. Zaman önemli ise bu yaklaşımlar kullanılmayabilir.

Burada kullanılan veritabanı aynı amaç için oluşturulmuş 464 örnekten ibarettir. Bu nedenle yapılan optimizasyonlar diğer uygulamalar için geçerli olmayabilir.

Yöntemin doğru sınıflayamadığı örnekler listelenmiştir. Bu örnekler birbirine çok benzeyen G ve Ğ gibi sınıfların işaretleridir. Bu birbirinden ayırt edilmesi zor örnekler ile baş etmenin yollarından biri iki aşamalı bir sınıflama stratejisi kullanmak olabilir: ilk adımda G ve Ğ gibi birbirine benzer sınıflar bir üst sınıfın, mesela G-Ğ sınıfının birer örneği olarak sınıflanır. İkinci adımda G-Ğ sınıfının altındaki örnekler bu sınıfı ayırt etmek üzere özel olarak oluşturulmuş bir sınıflayıcı tarafından ayırt edilebilir. Bu çalışmada bu ikinci sınıf için hangi algoritmanın kullanılacağı konusu araştırılmamış, gelecekteki bir çalışma konusu olarak bırakılmıştır.

Tez çalışmasının üçüncü aşamasında EdgeLap, SURF, HarAff, HesAff, kAS, FAST, IBR, PCBR, HarLap, HesLap, SR, MSER, ve DoG ilgi bölgeleri yine tez kapsamında oluşturulan ve Bölüm 4'te anlatılan katı-olmayan nesne sınıfı tanıma uygulaması yardımı ile karşılaştırılmıştır.

EdgeLap bölgelerinin en iyi tanıma başarısını yakaladığı ve SURF ve DoG bölgeleri tarafından takip edildiği görülmektedir. EdgeLap'ın başarısı çok miktarda bölge belirlemede ve bölgelerinin oldukça ayırt edici olmasına, SURF ve DoG yöntemlerinin başarısı ise bölgelerinin gerçekten ayırt edici olmasına verilebilir, çünkü SURF ve DoG yöntemleri göreceli olarak az sayıda bölge ile bunu başarmaktadır.

Bu çalışmada ilgi bölgelerinin ayırt ediciliklerini sayısallaştırmak için yeni bir ayırt edicilik istatistiği geliştirilmiştir. Bu istatistik belirleyici tarafından belirlenen bölgelerin tanıma başarısını rastgele üretilen bölgelerin tanıma başarısına kıyaslar.

Nowak vd.'nin çalışmasından (Nowak vd., 2006) ilgi bölgelerinin sayısının artması ile tanıma başarısının arttığını biliyoruz. Ancak bu çalışmada yüksek de olsa bir en uygun sayı değerinin olduğu, bu değerden sonra başarının artmadığı görülmektedir.

Bu çalışmada ayrıca rastgele üretilen bölgelerin nesne sınıfı tanıma sistemlerinde gerçekten kullanılabileceği değerlendirilmektedir: Rastgele bölgeler veritabanından bağımsız olarak, çok hızlı bir şekilde, istenen sayıda üretilebilmektedirler, çoğu belirleyiciden daha ayırt edicidirler ve başarı oranları en iyi belirleyiciler ile aynı seviyededir.

Genelleştirilmiş Hough Transformu, ilgi bölgeleri ve yerel tanımlayıcılar ile Türk işaret dili harfleri parmak hecelemeleri tanıma çalışması yöntemi birbirine çok benzeyen G ve Ğ gibi sınıfların işaretlerini tanıyamamaktadır. Bu birbirinden ayırt edilmesi zor örnekler ile baş

etmenin yollarından biri iki aşamalı bir sınıflama stratejisi kullanmak olabilir: ilk adımda G ve Ğ gibi birbirine benzer sınıflar bir üst sınıfın, mesela G-Ğ sınıfının birer örneği olarak sınıflanır. İkinci adımda G-Ğ sınıfının altındaki örnekler bu sınıfı ayırt etmek üzere özel olarak oluşturulmuş bir sınıflayıcı tarafından ayırt edilebilir. Bu çalışmada bu ikinci sınıf için hangi algoritmanın kullanılacağı konusu araştırılmamış, gelecekteki bir çalışma konusu olarak bırakılmıştır.

Bu tez kapsamında ilgi bölgeleri değerlendirilmiştir. İlgi bölgeleri kullanan birçok tanıma sisteminde diğer bir önemli adım yerel tanımlayıcıların çıkarılmasıdır. Bu çalışmada tüm deneyler için SIFT tanımlayıcısı kullanılmıştır. Bu çalışmanın devamı olarak yapılabilecek bir çalışma diğer yerel tanımlayıcıların ilgi bölgeleri ile eşleşimlerinin başarısını değerlendirmektir. Bazı ilgi belirleyicilerin (örneğin kAS ve SURF ilgi bölge belirleyicilerin) ilgi bölgesi belirleme yanında yerel tanımlayıcılar da çıkarması bu fikri güçlendirmektedir.

Bir başka çalışma yönü ilgi bölgelerinin birbirini ne kadar tamamladığını görmektir. Bu önemlidir çünkü ilgi bölgelerinin bir kombinasyonu ile herhangi bir ilgi bölgesinden daha iyi bir başarı elde edilebilir.

Bu çalışma ve Nowak vd. (Nowak vd., 2006) rastgele bölgelerin tanıma amaçlı kullanılabileceğini göstermektedir. Bu durumda rastgele bölgeler üzerine daha iyi bir inceleme yapılması gerekmektedir. İncelenmesi gereken sorular arasında şunlar vardır: Verilen bir imge için kaç bölge üretilmelidir? En iyi rastgele bölge üretme yöntemi nedir? Rastgele bölgeleri bölge belirleyicilerin bölgeleri ile kombine etmek iyi bir fikir midir? Eğer öyle ise hangi belirleyiciler ile kombine edilmelidir?

KAYNAKLAR

- Agarwal, S. ve Roth, D., (2002), "Learning a Sparse Representation for Object Detection", European Conference on Computer Vision, 2002, Copenhagen.
- Aha, D. W., Kibler, D. ve Albert, M. K., (1991), "Instance-Based Learning Algorithms", Machine Learning 6(1): 37-66.
- Altun, O. ve Albayrak, S., (2006), "Increasing the Effect of Fingers in Fingerspelling Hand Shapes by Thick Edge Detection and Correlation with Penalization", Advances in Image and Video Technology, Chang ve Lie, Springer Berlin / Heidelberg, 4319: 1133-1141.
- Anelli, M., Cinque, L. ve Sangineto, E., (2007), "Deformation tolerant generalized Hough transform for sketch-based image retrieval in complex scenes", Image and Vision Computing.
- Aran, O., (2008), "Vision Based Sign Language Recognition: Modeling and Recognizing Isolated Signs with Manual and Non-manual Gestures". Doktora Tezi, Boğaziçi Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Arı, İ., (2008), "Facial Feature Tracking and Expression Recognition for Sign Language". Doktora Tezi, Boğaziçi Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Ballard, D. H., (1981), "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition 13(2): 111-122.
- Bay, H., Ess, A., Tuytelaars, T. ve Gool, L. V., (2008), "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding.
- Beinglass, A. ve Wolfson, H. J., (1991), "Articulated Object Recognition or: How to Generalize the Generalized Hough Transform", IEEE Computer Vision and Pattern Recognition, 1991, Maui, HI, USA
- Belongie, S., Malik, J. ve Puzicha, J., (2002), "Shape matching and object recognition using shape contexts", IEEE Transactions on Pattern Analysis and Machine Intelligence 24(4): 509-522.
- Breiman, L., (2001), "Random forests", Machine Learning 45(1): 5-32.
- Chai, D. ve Bouzerdom, A., (2000), "A Bayesian Approach To Skin Colour Classification", TENCON-2000.
- Chen, Q., Georganas, N. D. ve Petriu, E. M., (2008), "Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar", IEEE Transaction on Instrumentation and Measurement 57(8): 1562-1571.
- Deng, H., Zhang, W., Mortensen, E., Dietterich, T. ve L, S., (2007), "Principal Curvature-based Region Detector for Object Recognition", IEEE Conference on Computer Vision and Pattern Recognition.
- Dollar, P., Rabaud, V., Cottrell, G. ve Belongie, S., (2005), "Behavior Recognition via Sparse Spatio-Temporal Features", 2nd Joint IEEE International Workshop on VS-PETS, Beijing.
- Dorko, G. ve Schmid, C., (2003), "Selection of Scale-Invariant Parts for Object Class Recognition", International Conference on Computer Vision.
- Eaton, R. S., Stevens, M. R., McBride, J. C., Foil, G. T. ve Snorrason, M. S., (2006), "A Systems View of Scale Space", Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006).
- Fergus, R., Perona, P. ve Zisserman, A., (2003), "Object Class Recognition by Unsupervised

- Scale-Invariant Learning", IEEE Conference on Computer Vision and Pattern Recognition.
- Ferrari, V., Fevrier, L., Jurie, F. ve Schmid, C., (2006a), "Groups of Adjacent Contour Segments for Object Detection", INRIA technical report.
- Ferrari, V., Tuytelaars, T. ve Gool, L. V., (2004), "Simultaneous Object Recognition and Segmentation by Image Exploration", European Conference on Computer Vision.
- Ferrari, V., Tuytelaars, T. ve Gool, L. V., (2006b), "Object Detection by Contour Segment Networks", European Conference on Computer Vision.
- Freeman, W. T. ve Adelson, E. H., (1991), "The Design and Use of Steerable Filters", IEEE Transactions on Pattern Analysis and Machine Intelligence 13(9): 891-906.
- Freund, Y. ve Schapire, R. E., (1999), "A short Introduction to Boosting", Journal of Japanese Society for Artificial Intelligence 14(5): 771-780.
- Fritzke, B., (1994), "Fast Learning with Incremental RBF Networks", Neural Processing Letters 1(1): 2-5.
- Gao, W., Fang, G. L., Zhao, D. B. ve Chen, Y. Q., (2004), "A Chinese sign language recognition system based on SOFM/SRN/HMM", Pattern Recognition 37(12): 2389-2402.
- Goh, P. ve Holden, E.-J., (2006), "Dynamic Fingerspelling Recognition using Geometric and Motion Features", Image Processing, 2006 IEEE International Conference on Atlanta, GA
- Gool, L. V., Moons, T. ve Ungureanu, D., (1996), "Affine/photometric invariants for planar intensity patterns", Fourth European Conf. Computer Vision.
- Gökmar, G. ve Yıldırım, T., (2005), "El İşaretlerinin Yapay Sinir Ağları ile Algılanması", 13. Sinyal İşleme ve İletişim Uygulamaları, Kayseri, IEEE.
- Haberdar, H. ve Albayrak, S., (2006), "Vision Based Real Time Isolated Turkish Sign Language Recognition", International Symposium on Methodologies for Intelligent Systems, Bari, Italy.
- Harris, C. ve Stephens, M., (1988), "A Combined Corner and Edge Detector", Alvey Vision Conference.
- Holden, E. J., Lee, G. ve Owens, R., (2005), "Australian sign language recognition", Machine Vision and Applications 16(5): 312-320.
- John, G. H. ve Langley, P., (1995), "Estimating Continuous Distributions in Bayesian Classifiers", Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, Morgan Kaufmann.
- Kadir, T., Zisserman, A. ve Brady, M., (2004), "An affine invariant salient region detector", European Conference on Computer Vision, Springer-Verlag.
- Ke, Y. ve Sukthankar, R., (2004), "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", Conf. Computer Vision and Pattern Recognition.
- Keerthi, S. S., Shevade, S. K., Bhattacharyya, C. ve Murthy, K. R. K., (2001), "Improvements to Platt's SMO algorithm for SVM classifier design", Neural Computation 13(3): 637-649.
- Kimura, A. ve Watanabe, T., (1998), "Fast generalized Hough Transform: rotation, scale and translation invariant detection of arbitrary shapes", Trans IEICE.
- Kiryati, N., Kälviäinen, H. ve Alaoutinen, S., (2000), "Randomized or Probabilistic Hough Transform: Unified Performance Evaluation", Pattern Recognition Letters.
- Koenderink, J. ve Doorn, A. v., (1987), "Representation of Local Geometry in the Visual

System", *Biological Cybernetics* 55.

Lamdan, Y., Schwartz, J. T. ve Wolfson, H. J., (1988), "Object recognition by Affine Invariant Matching", *IEEE Conf. on Computer Vision and Pattern Recognition*.

Laptev, I., (2005), "On space-time interest points", *International Journal of Computer Vision* 64(2-3): 107-123.

Lazebnik, S., Schmid, C. ve Ponce, J., (2003), "Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition", *International Conference on Computer Vision*.

Lazebnik, S., Schmid, C. ve Ponce, J., (2005), "A sparse texture representation using local affine regions", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8): 1265-1278.

Leibe, B., Leonardis, A. ve Schiele, B., (2004), "Combined Object Categorization and Segmentation with an Implicit Shape Model", *European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision*.

Leibe, B. ve Schiele, B., (2003), "Interleaved Object Categorization and Segmentation", *British Machine Vision Conference*.

Leibe, B., Seeman, E. ve Schiele, B., (2005), "Pedestrian Detection in Crowded Scenes", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

Li, F. F., Fergus, R. ve Perona, P., (2004), "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories", *IEEE Conference on Computer Vision and Pattern Recognition*.

Lindeberg, T., (1993), "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*.

Lindeberg, T., (1998), "Feature Detection with Automatic Scale Selection", *Int'l J. Computer Vision*.

Lindeberg, T. ve Bretzner, L., (2003), "Real-time scale selection in hybrid multi-scale representations", *Scale Space Methods in Computer Vision, Proceedings* 2695: 148-163.

Lindeberg, T. ve Garding, J., (1997), "Shape-Adapted Smoothing in Estimation of 3-D Shape Cues from Affine Deformations of Local 2-D Brightness Structure", *Image and Vision Computing*.

Lowe, D. G., (2001), "Local Feature View Clustering for 3D Object Recognition", *IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society*.

Lowe, D. G., (2004), "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision* 60(2): 91-110.

Machacon, H. T. C. ve Shiha, S., (2010), "Recognition of Japanese Finger Spelling Gestures Using Neural Networks", *Journal of Medical Engineering & Technology* 34(4): 254-260.

Martin, D., Fowlkes, C. ve Malik, J., (2004), "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues", *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Matas, J., Chum, O., Urban, M. ve Pajdla, T., (2002), "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", *British Machine Vision Conference*.

- Matas, J., Galambos, C. ve Kittler, J., (2000), "Robust detection of lines using the progressive probabilistic hough transform", Computer Vision Image Understanding.
- Mikolajczyk, K., Leibe, B. ve Schiele, B., (2005a), "Local Features for Object Class Recognition", Tenth IEEE International Conference on Computer Vision (ICCV'05), IEEE Computer Society.
- Mikolajczyk, K., Leibe, B. ve Schiele, B., (2006), "Multiple Object Class Detection with a Generative Model", CVPR.
- Mikolajczyk, K. ve Schmid, C., (2001), "Indexing Based on Scale Invariant Interest Points", Eighth Int'l Conf. Computer Vision.
- Mikolajczyk, K. ve Schmid, C., (2002), "An Affine Invariant Interest Point Detector", ECCV, Springer-Verlag.
- Mikolajczyk, K. ve Schmid, C., (2004), "Scale & affine invariant interest point detectors", International Journal of Computer Vision 60(1): 63-86.
- Mikolajczyk, K. ve Schmid, C., (2005), "A performance evaluation of local descriptors", Ieee Transactions on Pattern Analysis and Machine Intelligence 27(10): 1615-1630.
- Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. ve van Gool, L., (2005b), "A comparison of affine region detectors", International Journal of Computer Vision 65(1-2): 43-72.
- Mikolajczyk, K., Zisserman, A. ve Schmid, C., (2003), "Shape Recognition with Edge Based Features", British Machine Vision Conference.
- Moreels, P. ve Perona, P., (2007), "Evaluation of Features Detectors and Descriptors based on 3D Objects", Int. J. Comput. Vision 73(3): 263-284.
- Niebles, J. C., Wang, H. ve Fei, L. F., (2006), "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words", British Machine Vision Conference.
- Nowak, E., Jurie, F. ve Triggs, B., (2006), "Sampling Strategies for Bag-of-Features Image Classification", European Conference on Computer Vision.
- Opelt, A., Fussenegger, M., Pinz, A. ve Auer, P., (2004), "Weak Hypotheses and Boosting for Generic Object Detection and Recognition", European Conference on Computer Vision.
- Quinlan, J. R., (1986), "Induction of decision trees", Machine Learning: 81-106.
- Quinlan, R., (1993), "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, San Mateo, CA.
- Rosten, E. ve Drummond, T., (2005), "Fusing points and lines for high performance tracking", IEEE International Conference on Computer Vision.
- Rosten, E. ve Drummond, T., (2006), "Machine learning for high-speed corner detection", European Conference on Computer Vision.
- Sazonov, V., Vezhnevetsi, V. ve Andreeva, A., (2003), "A survey on pixel based skin color detection techniques", Graphicon-2003.
- Schaffalitzky, F. ve Zisserman, A., (2002), "Multi-view matching for unordered image sets, or 'how do I organize my holiday snaps?'"", European Conference on Computer Vision.
- Schmid, C. ve Mohr, R., (1997), "Local Grayvalue Invariants for Image Retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Seeman, E., Leibe, B., Mikolajczyk, K. ve Schiele, B., (2005), "An Evaluation of Local

Shape-Based Features for Pedestrian Detection", British Machine Vision Conference.

Sehgal, A. ve Desai, U. B., (2002), "3D object recognition using Bayesian geometric hashing and pose clustering", Pattern Recognition.

Shaked, D., Yaron, O. ve Kiryati, N., (1996), "Deriving stopping rules for the probabilistic Hough transform by sequential analysis.", Computer Vision Image Understanding.

Shapiro, L. G. ve Stockman, G. C., (2001), "Computer Vision", Prentice Hall.

Sivic, J. ve Zisserman, A., (2003), "Video Google: A Text Retrieval Approach to Object Matching in Videos", Ninth IEEE International Conference on Computer Vision (ICCV'03).

Stark, M. ve Schiele, B., (2007), "How Good are Local Features for Classes of Geometric Objects", International Conference on Computer Vision (ICCV).

Starnier, T., Weaver, J. ve Pentland, A., (1998), "Real-time American sign language recognition using desk and wearable computer based video", IEEE Transactions on Pattern Analysis and Machine Intelligence 20(12): 1371-1375.

Suetake, N., Uchino, E. ve Hirata, K., (2006), "Generalized Fuzzy Hough Transform for Detecting Arbitrary Shapes in a Vague and Noisy Image", Soft Computing - A Fusion of Foundations, Methodologies and Applications.

Torii, A. ve Imiya, A., (2007), "The Randomized-Hough-Transform-based Method for Great-Circle Detection on Sphere", Pattern Recognition Letters 28(10): 1186-1192.

Tuytelaars, T. ve Gool, L. V., (2004), "Matching Widely Separated Views based on Affine Invariant Regions", International Journal on Computer Vision.

Tuytelaars, T. ve Van Gool, L., (2000), "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions", 11th British Machine Vision Conference.

Umbaugh, S. E., (1998), "Computer Vision and Image Processing: A Practical Approach Using CVPTools", Prentice Hall.

Viola, P. ve Jones, M., (2001), "Robust real-time object detection", Tech. Rep. CRL2001/01, Cambridge Res. Lab.

Walpole, R. E., Myers, R. H., Myers, S. L. ve Ye, K., (2002), "Probability and Statistics For Engineers and Scientists", Prentice Hall.

Wang, C.-C. ve Wang, K.-C., (2008), "Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction", Lecture Notes in Control and Information Sciences.

Xie, L., Xu, P., Chang, S.-F., Divakaran, A. ve Sun, H., (2004), "Structure Analysis of Soccer Video with Domain Knowledge and Hidden Markov Models", Pattern Recognition Letters 25(7): 767-775.

Yang, H.-D. ve Lee, S.-W., (2010), "Simultaneous Spotting of Signs and Fingerspellings Based on Hierarchical Conditional Random Fields and Boostmap Embeddings ", Pattern Recognition 43(8): 2858-2870.

Yla-Jaaski, A. ve Kiryati, N., (1994), "Adaptive termination of voting in the probabilistic hough transform", IEEE Trans. Pattern Analysis Machine Intelligence.

EKLER

- Ek 1 Tez çalışması kapsamında yapılan bilimsel yayınlar.
- Ek 2 Tez çalışması başında yapılması düşünülen, ancak yapılmayan çalışmalar
- Ek 3 Tez çalışması sırasında yapılan ancak yayına dönüşmeyen çalışmalar.

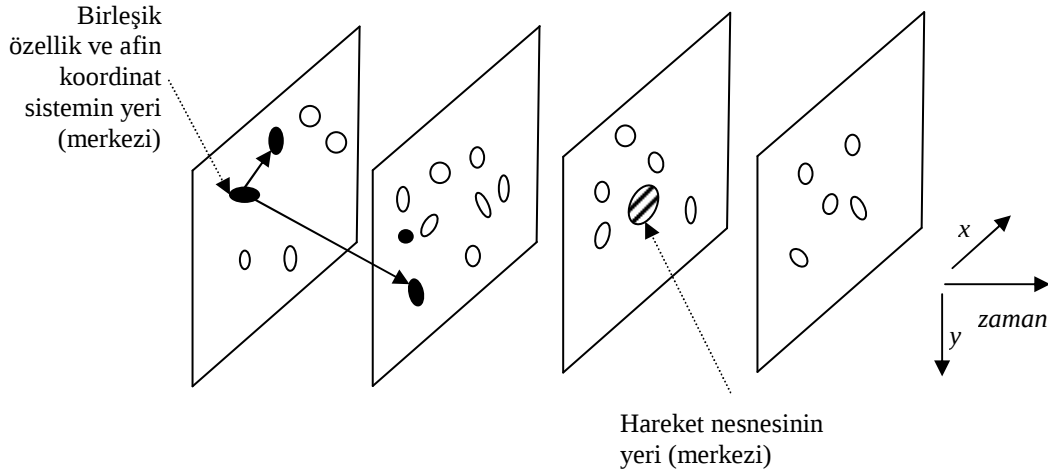
Ek 1 Tez Çalışması Kapsamında Yapılan Bilimsel Yayınlar

1. Altun, O. and Albayrak, S., (2006), "Increasing the Effect of Fingers in Fingerspelling Hand Shapes by Thick Edge Detection and Correlation with Penalization", Advances in Image and Video Technology, Chang ve Lie, Springer Berlin / Heidelberg, 4319: 1133-1141.
2. Altun, O., Albayrak, S., Ekinci, A. and Bükün, B., (2006), "Turkish Fingerspelling Recognition System Using Axis of Least Inertia Based Fast Alignment", AI 2006: Advances in Artificial Intelligence, Sattar ve Kang, Springer Berlin / Heidelberg, 4304: 473-481.
3. Altun, O. and Albayrak, S., (2007), "Hızlı Hizalama, Kalın Kenar Bulma ve Cezalandırılmalı Korelasyon ile Türk İşaret Dili Alfabesi Tanıma", Sinyal İşleme ve İletişim Uygulamaları Kurultayı SİU 2007, Eskişehir.

Ek 2 Tez Çalışması Başlangıcında Yapılması Düşünülen Ancak Yapılmayan Çalışmalar

Uzay-Zaman Genelleştirilmiş Hough Transformu

Videodan hareket tanımda, ilgi noktalarının zaman yönünde birbirine göre konumlarından henüz yararlanılmadığı görülmektedir. Bu çalışmada Genelleştirilmiş Hough Transformunun (Beinglass ve Wolfson, 1991) zaman boyutuna genişletilmesi ile noktaların hem uzay boyutunda hem de zaman boyutunda birbirine göre konumları değerlendirilebilir ve böylece tanımda başarı artırılabilir (Şekil 2).



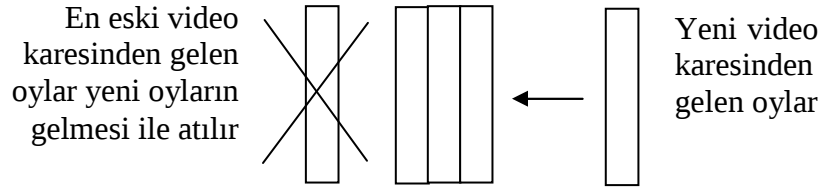
Şekil 2. Uzay Zaman Genelleştirilmiş Hough Transformunda bileşik özellik, bileşik özellik afin koordinat merkezi ve hareket nesnesinin merkezinin yeri

Şekil 2 da küçük elipsler ilgi noktalarını ve noktaların çevresindeki ilgi bölgelerini göstermektedir. İçerideki elipsler bir bileşik özellik oluşturmaktadır. Noktalardan bir şekil belirteci çıkarılır. Ayrıca noktalardan birini merkez alan bir afin koordinat sistemi oluşturulur. Afin koordinat sisteminde hareket nesnesinin yeri R-Tablosuna yazılır. Bileşik özellikteki noktalardan biri farklı bir video karesinden seçileceğinden yöntem otomatik olarak zaman boyutuna genişletilmiş olur.

Sürekli Uzay-Zaman Genelleştirilmiş Hough Transformu

Ayrıca Hough transformu video kameradan gelmekte olan (devamlı, continuous/online) kareler yardımı ile hareket tanıyabilecek şekilde iyileştirilebilir. Böylece devamlı hareketlerin (örneğin işaret dili ile anlaşılan işitme engellinin ardı ardına kesintisiz hareketlerinin) daha hareket yapılırken tanınması başarılabilecektir.

Bu genişletme, Hough transformu birikecinin (accumulator) değiştirilmesi ile yapılabilir. Yeni tip birikeç, son K video karesinden gelen ikili oyları hatırlayabilen bir birleşik birikeçtir (Şekil 3). Şekilde son $K=4$ video karesinden gelen oyları hatırlayan bir system görülmektedir. Uygulamada K daha büyük olabilir.

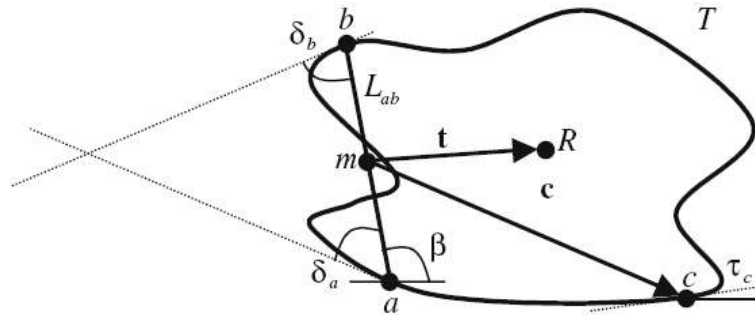


Şekil 3. Sürekli Uzak-Zaman Genelleştirilmiş Hough Transformu için bileşik birikeç.

Bileşik birikeçin K bileşeni olsun, her bileşen bir video karesinden gelen oyları biriktirebilir. Her yeni video karesinin gelmesi ile en eski bileşen silinecek, yerine yeni karenin oylarını içeren bileşen eklenecek ve yeniden sayım yapılacaktır. İşlemin hızlanması için her N video karesinde bir sayım yapılabilir veya bileşenler tek video karesinden değil de N video karesinden gelen oyları biriktirebilir. K sayısı tanınmak istenen hareketlerin sürdüğü ortalama zaman miktarına göre seçilmelidir. Hough transformunun eksik sayıda nokta ile nesne tanıyabilme özelliği, zamanda K kareden fazla yayılan hareketlerin de tanınabilmesine olanak verecektir.

GHT ve Kontrol Noktası

Suetake vd. (Suetake vd., 2006), Kimura ve Watanabe'nin (Kimura ve Watanabe, 1998) Fast Generalized Hough Transform (FGHT) yaklaşımındaki oylama işlemini olasılıksal hale getirmişlerdir. Kimura ve Watanabe'nin (Kimura ve Watanabe, 1998) FGHT algoritması kenarları çıkarılmış imgeler (contour images) üzerinde çalışır: T model imgesi olsun. T kare şeklinde bloklara ayrılır. Her bloğun içinde kalan kenar kısmı bir çizgi parçası (line segment) ile yaklaşıkları. Bu parçalardan yeterli uzunlukta olanların oluşturduğu küme T ile gösterilsin. Öncelikle modelin referans noktası R Şekil 4 ile de gösterildiği gibi rasgele bir nokta olarak seçilir.



Şekil 4 Bir T modeli ve üzerinde tanımlı parametreler. Kaynak: (Suetake vd., 2006)

Daha sonra *genişletilmiş C-tablosu* (bu çalışmada R-Tablosu yerine bu terim kullanılmıştır) Şekil 5 ile betimlendiği gibi oluşturulur. Bunun için T içinden seçilen iki parça olan T_a ve T_b nin orta noktaları a ve b kullanılır. Noktalar arası uzaklık L_{ab} belirli bir eşikten büyük olmalıdır. Genişletilmiş C-tablosuna tüm olası a ve b değerleri için değişmezler δ_a , δ_b , kontrol bilgileri τ_c , c_x , c_y , L_{ab} , β ve referans noktasının yeri t_x , t_y yazılır. δ_a , δ_b parametreleri T_a , T_b ve ab çizgisinin oluşturduğu üçgenin a ve b deki iç açılarıdır. kontrol noktası (checkpoint) c , T_a ve T_b haricinde seçilen bir T_c parçasının orta noktasıdır. τ_c kontrol noktası c deki teğet çizginin açısıdır. c , ab çizgisinin merkezinden c ye uzanan vektördür. β , ab çizgisinin bir açısıdır. t , ab çizgisinin merkezinden R ye uzanan vektördür.

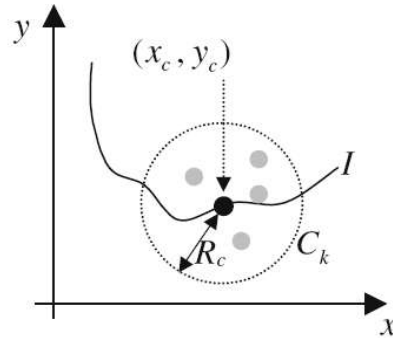
No.	Invariant feature	Check point		Line segment $\overline{ab}$		Vector to R
		Angle	Vector	Length	Angle	
1	$\delta_a(1), \delta_b(1)$	$\tau_c(1)$	$(c_x(1), c_y(1))$	$L_{ab}(1)$	$\beta(1)$	$(t_x(1), t_y(1))$
2	$\delta_a(2), \delta_b(2)$	$\tau_c(2)$	$(c_x(2), c_y(2))$	$L_{ab}(2)$	$\beta(2)$	$(t_x(2), t_y(2))$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Şekil 5 Genişletilmiş C-Tablosu. Kaynak: (Suetake vd., 2006)

Tanıma aşamasında test imgesi önce modelin geçtiği işlemlere benzer işlemler geçirilir: İmge bloklara bölünür, bloklardan yeterince uzun çizgi parçaları çıkarılır, çizgi parçalarının kümesinden rasgele a ve b ikilileri seçilir ve değişmezler δ_a ve δ_b hesaplanır. Daha sonra genişletilmiş C-Tablosunda yakın değişmez değerlerine sahip satırlar bulunur, bu satırlar için kontrol noktası c nin imgede olması gereken yer hesaplanır. İmgede bu yerin yeterince yakınında bir çizgi parçası varsa ve τ_c açısı modeldeki değere yakın ise referans noktasının yeri için oy kullanılır.

Suetake vd. (Suetake vd., 2006) genelleştirilmiş bulanık Hough transformu (generalized fuzzy Hough transform) adını verdikleri yaklaşım ile yukarıda anlatılan algorithmada

Şekil 6 de gösterilen değişikliği yaparlar.



Şekil 6 Suetake vd. (Suetake vd., 2006) nın genelleştirilmiş bulanık Hough Transformu için C_k alanı. Kaynak: (Suetake vd., 2006)

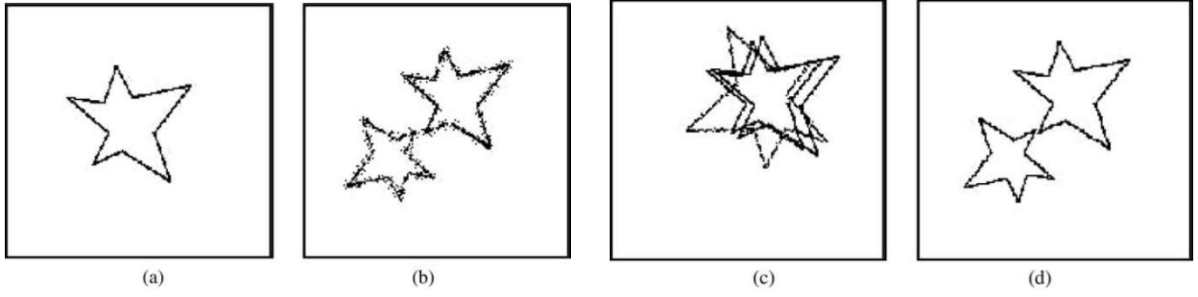
Kontrol noktası c nin imgede olması gereken yer (x_c, y_c) olarak hesaplanmış olsun. (x_c, y_c) merkezli ve R_c yarıçaplı C_k alanında bulunan noktaların sonuca etki etmesi isteniyor. Bunun için C_k içindeki her noktaya (x_c, y_c) ye uzaklığına bağlı bir üyelik değeri verilir. Uzaklık yüksek olduğunda üyelik değerinin az, uzaklık kısa olduğunda ise üyelik değerinin çok olması gayet doğal olacaktır. Bu nedenle üyelik değeri m (1) ile verilen Gauss benzeri formülle hesaplanır:

$$m(r) = \exp\left(\frac{-r^2}{s^2}\right) \quad (1)$$

Bu formülde r , noktanın (x_c, y_c) ye olan uzaklığı, s ise bir sabittir. Oylama işleminde atılacak oyun miktarı ise (2) ile hesaplanır:

$$\text{Oy miktarı} = \sum (C_k \text{ içindeki özellik noktasının üyelik değeri}) \quad (2)$$

Şekil 7 kendi iyileştirmelerinin yaptığı pozitif etkiyi anlatmaktadır.



Şekil 7 Gürültülü bir imgede bozulmuş bir yıldız şeklinin bulunması. (a) Model imgesi (128 x 128 piksel). (b) Gürültülü bir hedef imgesi. (c) Kimura ve Watanabe (Kimura ve Watanabe, 1998) yaklaşımı sonucu. (d) Suetake vd. (Suetake vd., 2006) yaklaşımı sonucu. Kaynak: (Suetake vd., 2006)

Genelleştirilmiş Hough Transformu ile parmak heceleme tanımada da benzer yaklaşımlar kullanılabilir.

Performansın İyileştirilmesi

Hough transformu kaba güç (brute force) kullanan bir yaklaşımdır. Noktaların tümünün oyları değerlendirilmektedir. Algoritma kuvantalama ve R-Tablosu / birikeç gibi yapıların verimli (efficient) bir şekilde gerçekleştirilmesi gibi artılarına rağmen kaba güç kullanan yaklaşımların verimsizliğini paylaşmaktadır.

Yeterli Kalitede Olmayan Noktaların Kullanılmaması

Bu sorunun çözümü ile ilgili ilk tedbir sadece en kaliteli noktaların kullanılması, başka bir deyişle gürültü olabilecek noktaların R-Tablosu oluşturma ve oylamada kullanılmaması olabilir. Hangi noktalar daha kalitelidir? Nokta çıkarma yöntemine ve uygulama alanına bağlı olarak farklı kalite tanımları yapılabilir. Örneğin Sivic ve Zisserman iki boyutlu noktaları kullandıkları “Video Google” isimli çalışmalarında (Sivic ve Zisserman, 2003) bir noktanın değerlendirmeye alınması için varlığının arka arkaya en az üç video çerçevesinde devam etmesini şart koşar. Harris ve Hessian nokta bulucusu gibi bir kısım algoritmalar (Mikolajczyk ve Schmid, 2002) her nokta için bir kalite sayısı zaten üretirler.

Noktaların Rasgele Bir Altkümesinin Kullanılması

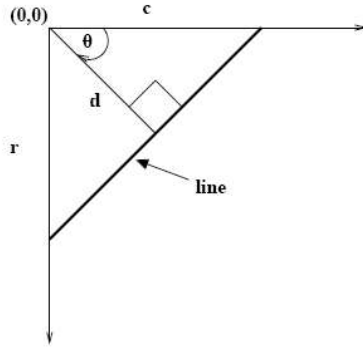
Literatürde incelenmiş bir yaklaşım ise tanıma işlemi sırasında test imgesinden çıkan tüm noktaların kullandığı oylar yerine, rasgele seçilen bir alt kümenin kullandığı oylar ile karar vermedir. Bu yaklaşımı kullanan algoritmalar arasında Probabilistic Hough Transform (PHT) ve Randomized Hough Transform (RHT) vardır. Bu iki algoritma aslında imgeler üzerinde doğru bulma amacı ile kullanılan standart Hough transformunun geliştirmesidir. Bu nedenle önce standart Hough transformu ile doğru bulma yaklaşımı özetlenecektir.

Standart Hough Transformu

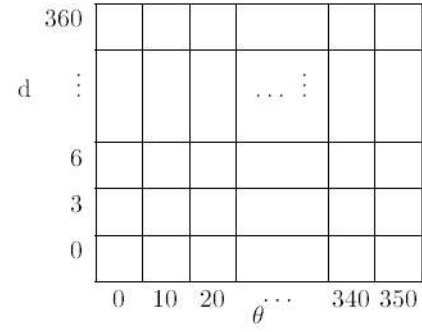
Standart Hough transformu (SHT) olarak da adlandırılan orijinal Hough transformunda, imge kenarları üzerindeki her (r, c) noktası polar koordinatlarda (d, θ) ikilisi ile parametrize edilen ve (3) ile ifade edilen doğrulara oy verir.

$$d = -r \sin \theta + c \cos \theta \quad (3)$$

Burada d noktanın imge merkezine uzaklığı, θ ise imge merkezinden noktaya uzayan vektörün imge merkezi koordinat sistemi x eksenine yaptığı açıdır. SHT *birden çoğa* ($1 \rightarrow m$) bir algoritmadır. Yani her bir nokta olası birikeçte birçok seleye, birçok doğru için, oy verir.



(a)



(b)

Şekil 8 (a) Doğru denklemi $d = -r\sin\theta + c\cos\theta$ da kullanılan d ve θ parametreleri. (b) 256 x 256 boyutlarındaki imgelerde doğru bulmak için kullanılan birikeç örneği. Kaynak: (Shapiro ve Stockman, 2001)

Olasılıksal (Probabilistic) ve Rasgeleleştirilmiş (Randomized) Hough Transformatları

Olasılıksal (Probabilistic) Hough Transformatı (PHT) şu gözleme dayanır: bir çok kere, karar vermek için birikeçteki tüm oyların birikmesine gerek kalmaz. Toplam nokta sayısının çok daha küçük bir kısmının verdiği oylar (yani bir anket) genel bir oylamada verilecek sonucu verebilir. Daha az oy verilmesi ise hız konusunda büyük iyileşmeler kaydetmek anlamına gelir. PHT de SHT gibi birden çoğa bir algoritmadır.

Rasgeleleştirilmiş (Randomized) Hough Transform (RHT) nin PHT den temel farkı *çoktan bire* ($m \rightarrow 1$) bir algoritma olmasıdır. İki yada daha fazla noktadan oluşan bileşik özellikler yardımı ile *bir* doğrunun parametreleri d ve θ belirlenir ve sadece o birikeç selesine oy atılır. Bu algoritmada da az sayıdaki rasgele nokta tüm nokta kümesinin yerine kullanılır, dolayısı ile hız yönünden kazanç sağlanır.

Kiryati (Kiryati vd., 2000) RHT ve PHT yi karşılaştırır ve tanıma başarısı açısından RHT nin az gürültülü ve kaliteli, PHT nin ise çok gürültülü ve kalitesiz imgelerde kullanılmasını tavsiye eder.

RHT ve PHT de kritik soru ne büyüklükte bir alt kümenin seçileceği sorusudur. Kiryati (Kiryati vd., 2000), PHT için (Yla-Jaaski ve Kiryati, 1994) ve (Shaked vd., 1996) çalışmalarında bu konu üzerinde durulduğunu, RHT de ise birikeçteki en yüksek zirve bir eşiğe ulaştığında oylamanın sona erdiğini bildirir.

Kademeli Olasılıksal (Progressive Probabilisti) Hough Transformatı

Matas vd. (Matas vd., 2000) noktaların rasgele bir alt kümesini seçmek yerine her oy atımında birikeci kontrol etmek ve aranan nesnenin bir örneğini bulur bulmaz imgede o nesneye ait noktaları oy verecek noktalar listesinden çıkarmak yoluna gitmişlerdir. Böylece bir çok nokta hiç oy vermeden bir nesneyle ilişkilendirilir, ve oy verecek noktaların sayısı hızla azaltılabilir. Standart Hough Transformatı gibi birden çoğa olan ve doğru bulma amacı ile kullanılan algoritma

Ek 3 Tez Çalışması Sırasında Yapılan Ancak Yayına Dönüşmeyen Çalışmalar

Türev Operatörleri

Kodlama dili olarak Matlab seçildiğinden bu dilin konvolüsyon işlemini yapma şekli çok önem kazanmaktadır. Ölçek uzayı çıkarılırken hangi türev operatörleri kullanılmalıdır?

Eaton vd. de dikkat çeken bir nokta türev operatörü olarak $[-1 \ 1]/2$ kullanılmasıdır. Halbuki geleneksel olarak bu amaçla $[-1 \ 0 \ 1]/2$ operatörü kullanılmaktadır. Bu nedenle bu iki operatörün sistemin başarısına olası etkileri karşılaştırılmıştır.

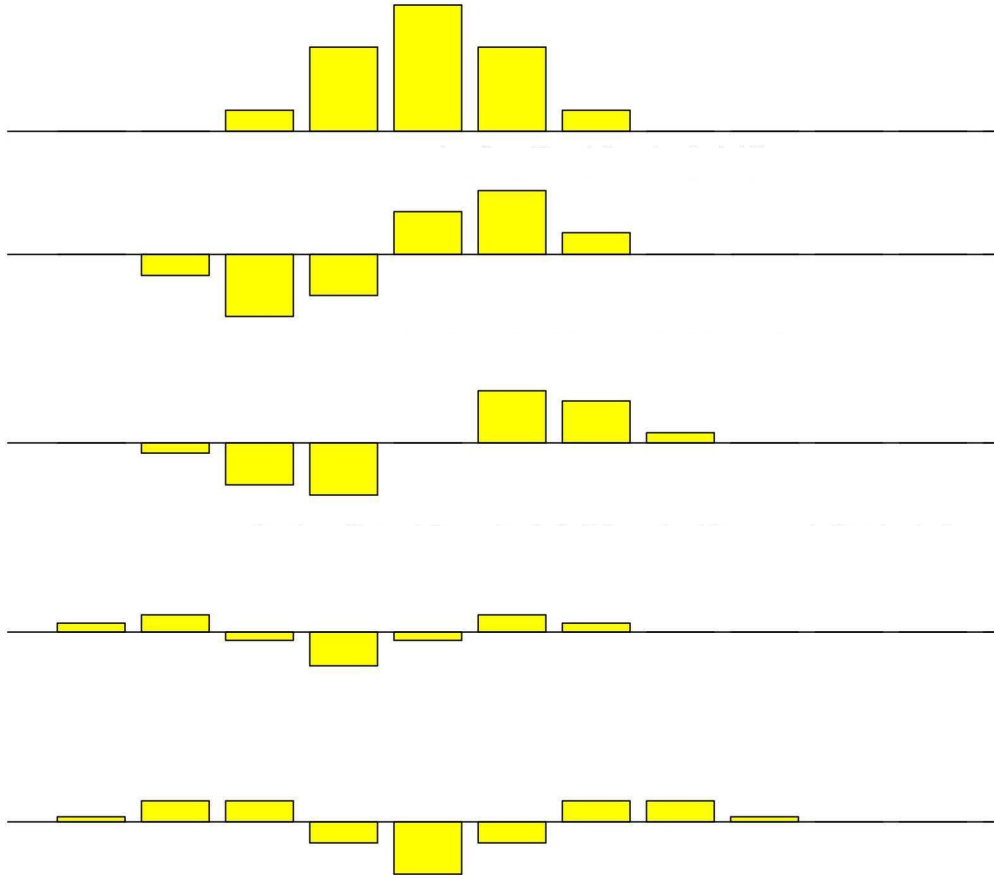
Yapılan incelemeler sonucunda $[-1 \ 1]/2$ operatörü ile ilgili bir büyük bir de küçük sorun göze çarpmıştır: Büyük sorun, bu operatörün imge ilgi noktasının yerini yarım piksel kaydırmasıdır. Başka bir deyiş ile bu operatörün arka arkaya iki kere uygulanması ile elde edilen ikinci türevlerde ilgi noktaları bir piksel kaymaktadır. Ölçek uzayının yüksek düzlemlerinde, yani sonraki oktavlardaki düzlemlerde, bir piksellik bir hata oldukça büyük bir hata anlamına gelmektedir. Çünkü bu oktavlardaki bir piksel uzaklık başlangıçtaki oktavlardaki 4 piksellik uzaklığa denk gelmektedir. Küçük sorun ise $[-1 \ 1]$ in gereksiz yere ikiye bölünmesidir. $[-1 \ 0 \ 1]$ operatörünün ikiye bölünmesi mantıklıdır çünkü bu operatör ile iki adımlık bir uzaklıktan elde edilen parlaklık farkı ikiye bölünerek tek bir adımlık uzaklıktan oluşan parlaklık farkı tahmin edilmeye çalışılmaktadır. $[-1 \ 1]$ operatörü ise zaten tek adımlık bir uzaklıkta oluşan parlaklık farkını vermektedir.

Sonuçta yer kaydırma sorunu nedeni ile $[-1 \ 1]$ operatöründen vazgeçilmiş ve $[-1 \ 0 \ 1]/2$ operatörü tez çalışmasında türev çıkarmak amacı ile kullanılmak üzere seçilmiştir.

Bu seçimler yapılan bir kısım gözlem ve karşılaştırmalara dayanmaktadır. Bu karşılaştırmalar ve elde edilen bazı çıktılar aşağıda sunulmuştur.

Basit Gauss sinyali üzerinde türev operatörlerinin karşılaştırılması

Türev operatörünün davranışı basit bir nokta ile gözlemlenebilir. Ancak ani kenar değişimi nedeni ile operatör çıktıları gerçek imgelerde karşılaştırılması zor cevaplar vermektedir. Bu nedenle operatörlerin davranışı öncelikle bir Gauss sinyali ile gözlemlenmiştir. Şekil 9 bu davranışı betimlemektedir.



Şekil 9 Basit bir Gauss sinyali üzerinde $[-1 \ 1]$ ve $[-1 \ 0 \ 1]/2$ operatörlerinin çıktıları.

Şekil 9 de yukardan aşağı birinci grafik orijinal sinyal “a” yı, $[-1 \ 1]$ ile elde edilen birinci türevi, $[-1 \ 0 \ 1]/2$ ile elde edilen birinci türevi, ardışık $[-1 \ 1]$ ’ler ile elde edilen ikinci türevi ve ardışık $[-1 \ 0 \ 1]/2$ ler ile elde edilen ikinci türevi göstermektedir. $[-1 \ 1]$ birinci türevde ekstremayı her zaman “ideal” çizginin bir piksel solunda verirken, $[-1 \ 0 \ 1]/2$ “ideal” çizginin merkeze bakan tarafında vermektedir. Bu nedenle $[-1 \ 1]$ asimetrik çıktılar vermekte, $[-1 \ 0 \ 1]/2$ ise simetrik çıktılar vermektedir.

Yapay imgeler üzerinde türev operatörlerinin karşılaştırılması

Türev operatörleri yapay imgelere uygulanarak çok kullanılan çıktı ve çıktı kombinasyonları karşılaştırılmıştır.

Her bir karşılaştırma adımı için üç değişkenden söz edilebilir: Birinci türev operatörü, ikinci türev operatörü, yapay imge. Bu değişkenlere farklı değerler verilerek bir çok çıktı dizisi oluşturulmuştur. Bu çıktı dizilerinden bazıları aşağıdaki şekillerde okurun dikkatine sunulmuştur.

Her bir çıktı dizisinde orijinal imgeden sonra sırası ile x yönündeki birinci türev (dx), y yönündeki birinci türev (dy), x yönündeki ikinci türev (dxx), y yönündeki ikinci türev (dyy), dxx ve dyy ’nin karşılıklı elemanlarının toplamı ($dxx+dyy$), dxx ve dyy ’nin karşılıklı elemanlarının çarpımı ($dxxmyy$), x yönünde birinci türev alınması sonrası y yönündeki ikinci türev alınması ile elde edilen dxy ’nin elemanlarının karesi (dxy^2), ve ilgi noktası bulmada kullanılan Laplacian of Gaussian (lofg), Hessian matrisi determinantı (hes) (Eaton vd., 2006) ve Harris fonksiyonu (har) (Mikolajczyk vd., 2005b) çıktıları verilmektedir. Bu nokta bulucu fonksiyonlar tanıma çalışmalarında çok kullanıldıkları için çıktılar arasına dahil edilirken, türev kombinasyonları nokta bulucuların yapı taşları oldukları için dahil edilmiştir.

LoG, Hessian ve Harris fonksiyonları (4), (5) ve (6) ile verilmiştir. Bu denklemlerde s ölçek (scale) değerini gösterir. $[-1 \ 0 \ 1]/2$ ve $[-1 \ 1]$ in incelendiği çalışmamızda imgeler bir ölçek uzayının düzlemleri değildir. Bu nedenle çıktılar elde edilirken standart sapma $s = 1$ kabul edilmiş ve notasyonun kısa tutulması için yazılmamıştır.

$$|LoG(x, y, s)| = s^2 |dxx(s) + dyy(s)| \quad (4)$$

$$Hessian(x, y, s) = \det \begin{bmatrix} dxx(s) & dxy(s) \\ dxy(s) & dyy(s) \end{bmatrix} \quad (5)$$

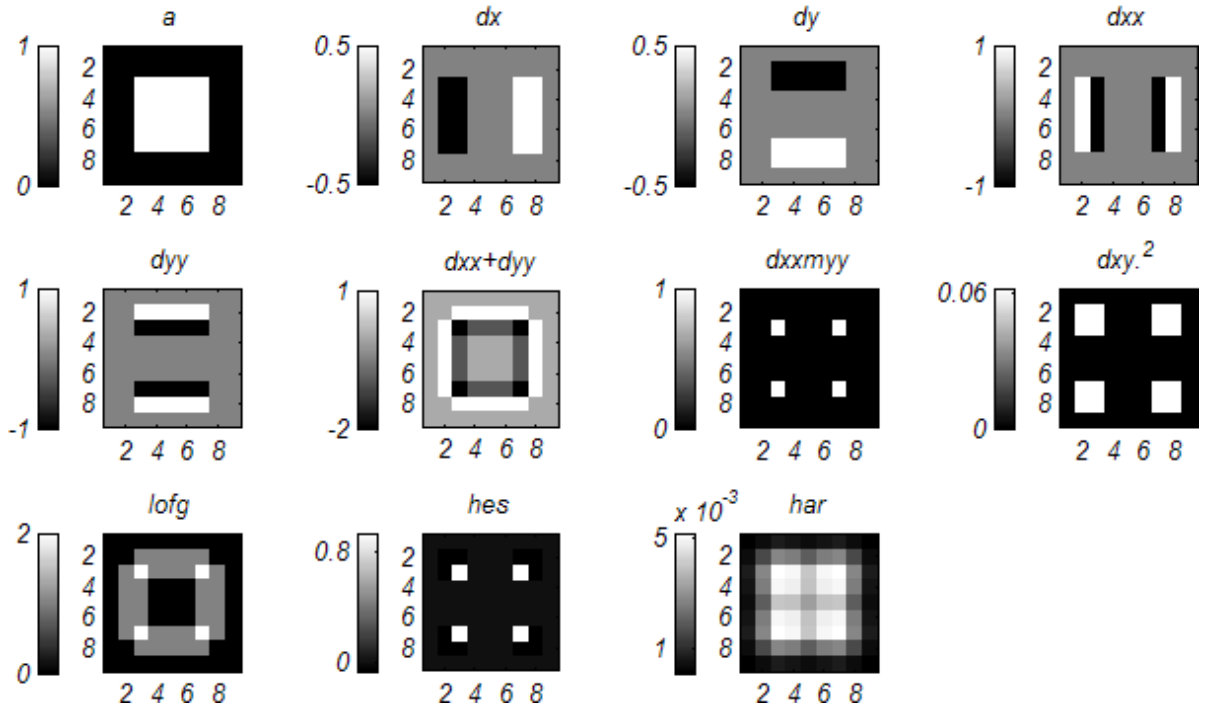
$$Harris(x, y, i, s) = \det(m) - k \ tr(m) \quad (6)$$

(4), (5) ve (6) da $\det(\cdot)$ determinant, $tr(\cdot)$ trace, k bir sabit, m ise (7) ile verilen şekil matrisidir. Şekil matrisi tanımında i integrasyon Gauss kernelinin standart sapmasıdır.

$$m(x, y, i, s) = s^2 g(i) * \begin{bmatrix} dx^2(s) & dx(s)dy(s) \\ dx(s)dy(s) & dy^2(s) \end{bmatrix} \quad (7)$$

Burada kaydedilmelidir ki lofg, hes ve har ile gösterilen çıktılar bu fonksiyonlara karşılık gelen nokta bulucuların sonuçlarını değil, sadece fonksiyonun ilgi noktaları etrafındaki değer dağılımını göstermektedir.

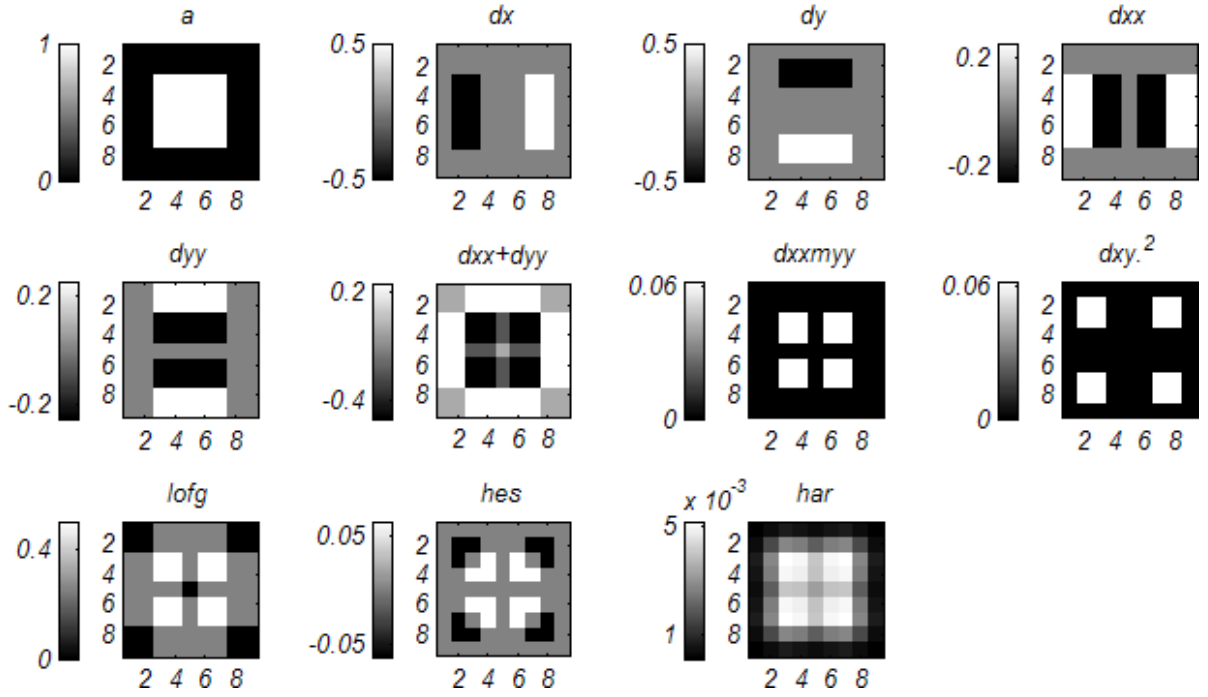
Kullanılan operatör ikilisi imge dizisinin üzerinde hx ve hxx ile gösterilmektedir. hx , x yönündeki birinci türevi, hxx ise aynı yöndeki ikinci türev operatörünü göstermektedir. hy ve hyy operatörleri öncekilerin transpozesi olduğundan ayrıca gösterilmemiştir.



Şekil 10 “a” orijinal imgesi üzerinde $hx=[-1 \ 0 \ 1]/2$ ve $hxx=[1 \ -2 \ 1]$ operatörlerini karşılaştıran çıktı dizisi

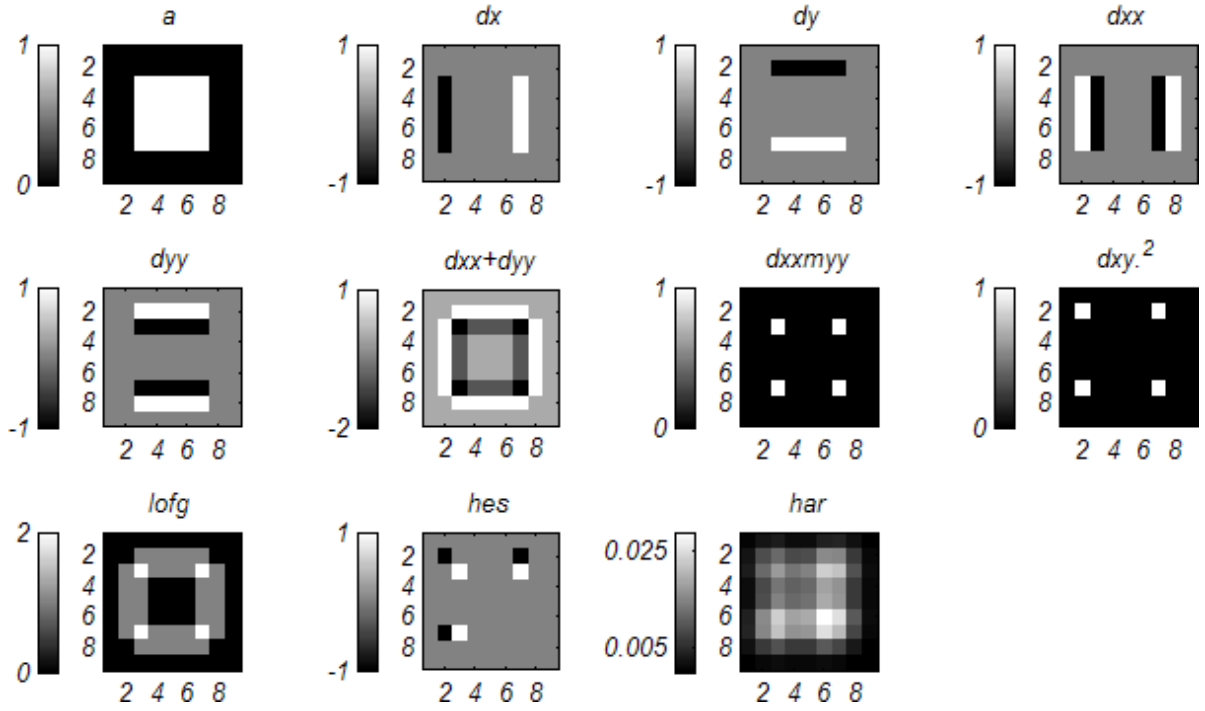
Şekil 10 de “a” imgesi 9x9 luk sıfır “deniz”inde yüzen 5x5 lik bir 1 “ada”sı şeklindedir. Türev

operatörlerinin davranışlarının daha kolay gözlenebilmesi için özellikle küçük oluşturulmuştur. “dx” x yönündeki birinci türevdir. Beklendiği gibi türev y doğrultusundaki kenarlarda ekstrema vermektedir. Eğer 1 lerden oluşan adanın “sahil”deki son pikselleri kenar kabul edilirse türevin hem bu kenar piksellerinde hem de denizin ilk piksellerinde aynı ekstremayı verdiği görülür. Bu $[-1 \ 0 \ 1]/2$ nin orta bileşenindeki 0 ile açıklanabilir. Hem adaya ait kenarlarda hem de denize ait kenarlarda 0 hedef pikselin değerinin değerlendirilmesini engellemekte, diğer piksellerden gelen değerler de aynı olduğundan aynı ekstremayı üretmektedir. Başka bir deyişle kenarları simetrik bir şekilde yaydığı söylenebilir. Benzer değerlendirmeler y yönündeki türev “dy” için de yapılabilir. “dxx” ve “dyy” yine simetriyi korumuştur. Çünkü $[1 \ -2 \ 1]$ tek boyutlu bir operatördür. Hem “dx” hem de “dxx” simetrik olunca onların kombinasyonları da simetriyi korumuştur. Nokta bulucu fonksiyonların ekstremalarının tam olması gerektiği yerde olması dikkat çekmektedir.



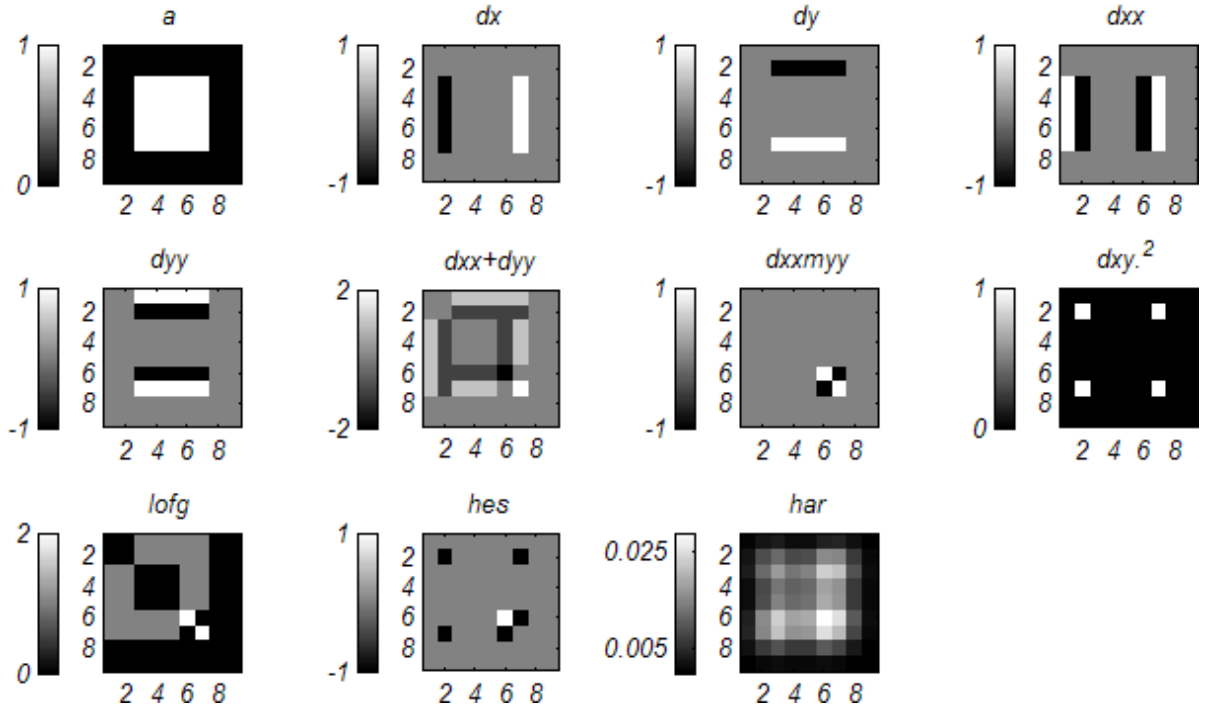
Şekil 11 “a” orijinal imgesi üzerinde $hx=[-1 \ 0 \ 1]/2$ ile birinci türev ve hx ’in arka arkaya iki kere uygulanması

Şekil 11 teki karşılaştırmanın bir öncekinden farkı ikinci türevin direkt bir operatör uygulanarak değil, birinci türeve $[-1 \ 0 \ 1]/2$ nin tekrar uygulanarak elde edilmesidir. Operatör simetrik olduğu için tüm çıktılar simetriktir. Efektif ikinci türev operatörü $[1 \ 0 \ -2 \ 0 \ 1]$ dir. Bu nedenle ikinci türev ekstremaları geniş çıkmış, sonuçta tüm çıktılarda ekstremalar daha geniş alanlara yayılmıştır. Buradan direkt $[1 \ -2 \ 1]$ uygulamanın avantajlı olduğu çıkarımı yapılabilir. Ancak ikinci türevin birinci türevden edilmesi çok daha doğal bir yaklaşımdır. Aynı noktada “dxy” ikinci türevi için mecburen birinci türev operatörü kullanılacağından $[1 \ -2 \ 1]$ in kullanımı sorun çıkarabilir. Bu konunun daha fazla incelenmesi gerekmektedir. Bu tez çalışmasında ikinci türev için bu çıktı dizisinde betimlenen yaklaşım tercih edilmiştir.



Şekil 12 “a” orijinal imgesi üzerinde $hx=[-1 \ 1]$ ve $hxx=[1 \ -2 \ 1]$.

Şekil 12 de birinci türev için $[-1 \ 1]$ kullanılması nedeni ile dx , LoG ve Harris simetriyi kaybetmişlerdir. Harris alt sağdaki köşeyi tamamen yitirmiştir. Bölüm 0 deki Şekil 9 incelendiğinde bu sonucun neden anlaşılabilir. $[-1 \ 1]$ operatörü birinci türevi asimetrik bir şekilde dağıtmaktadır. “ dx ”, “ dy ” ve bunların kullanıldığı “ dxy^2 ”, “ hes ” ve “ har ” bozulmuştur.



Şekil 13 “a” orijinal imgesi üzerinde $hx=[-1 \ 1]$ uygulanması.

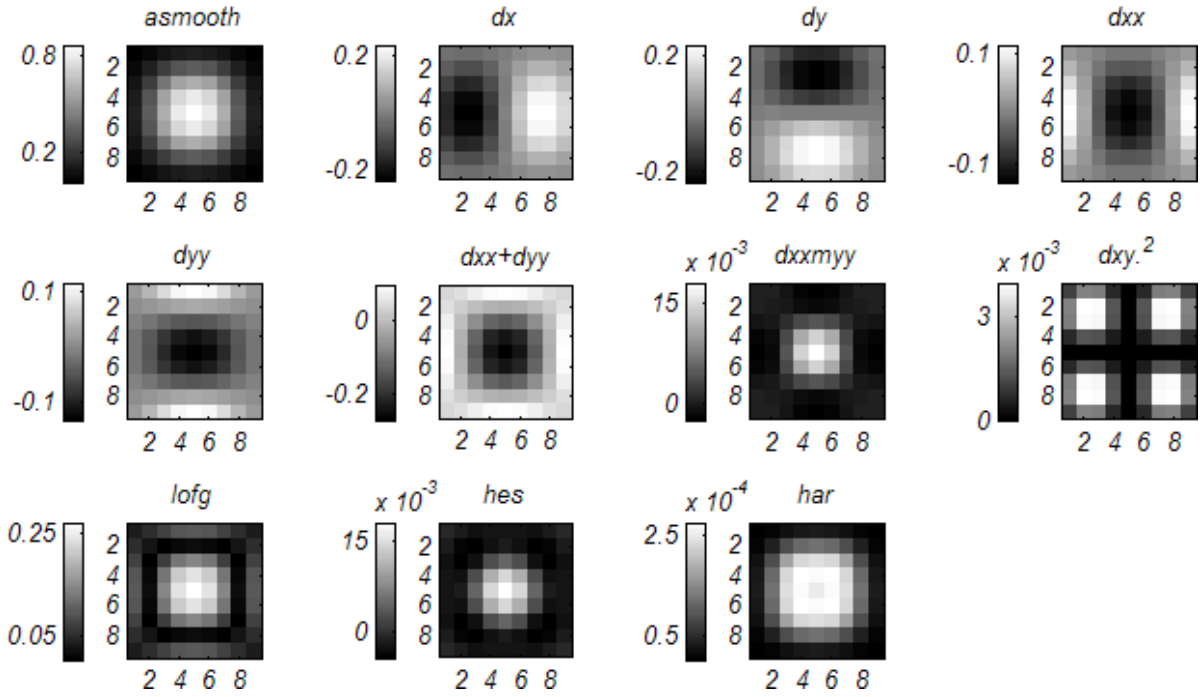
Şekil 13 te hem birinci, hem de ikinci türev için $[-1 \ 1]$ kullanımı görülmektedir. İkinci türevler

“dxx” ve “dyy” kayma miktarının bir piksele ulaştığını göz önüne sermektedir. Tüm çıktılar simetriyi kaybetmiş, nokta bulucu fonksiyonlar noktaların yerlerini yanlış belirlemiştir.

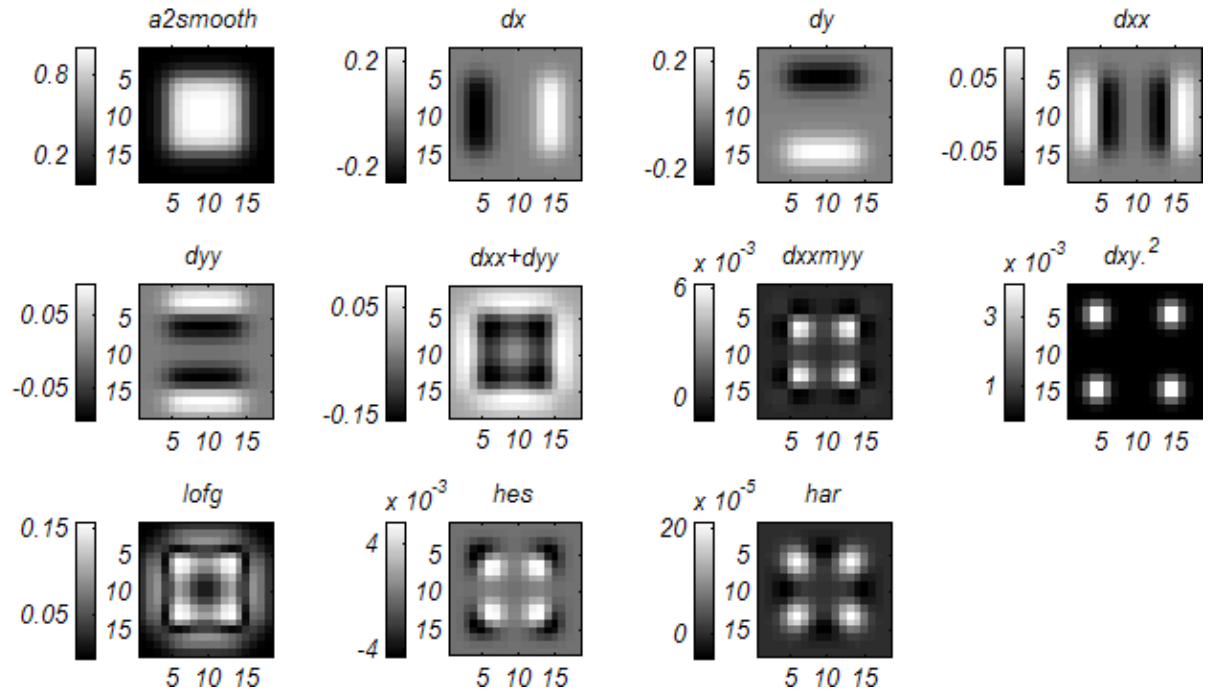
Çıktıların incelenmesi ile ortaya çıkan bir sonuç $[-1 \ 1]$ operatörünün simetriyi bozması nedeni tanıma amacı için uygun bir türev operatörü olmadığıdır. Birinci türev için $[-1 \ 0 \ 1]/2$ kullanılmalıdır.

Çıktıların incelenmesi ile ortaya çıkan ilgi çekici bir sonuç ise operatörlerin özellikleri incelenmek istediğinde konvolüsyon fonksiyonlarından yararlanırken çift operatörlerin son kısmına sıfırlar eklenerek simetrik hale getirilmesi gerektiğidir. Yukarıdaki örnekte $[-1 \ 1]$ ile iki kere konvolüsyon $[-1 \ 1]*[-1 \ 1] = [1 \ -2 \ 1]$ e değil $[-1 \ 1 \ 0]*[-1 \ 1 \ 0] = [1 \ -2 \ 1 \ 0 \ 0]$ e denk çıktı vermektedir.

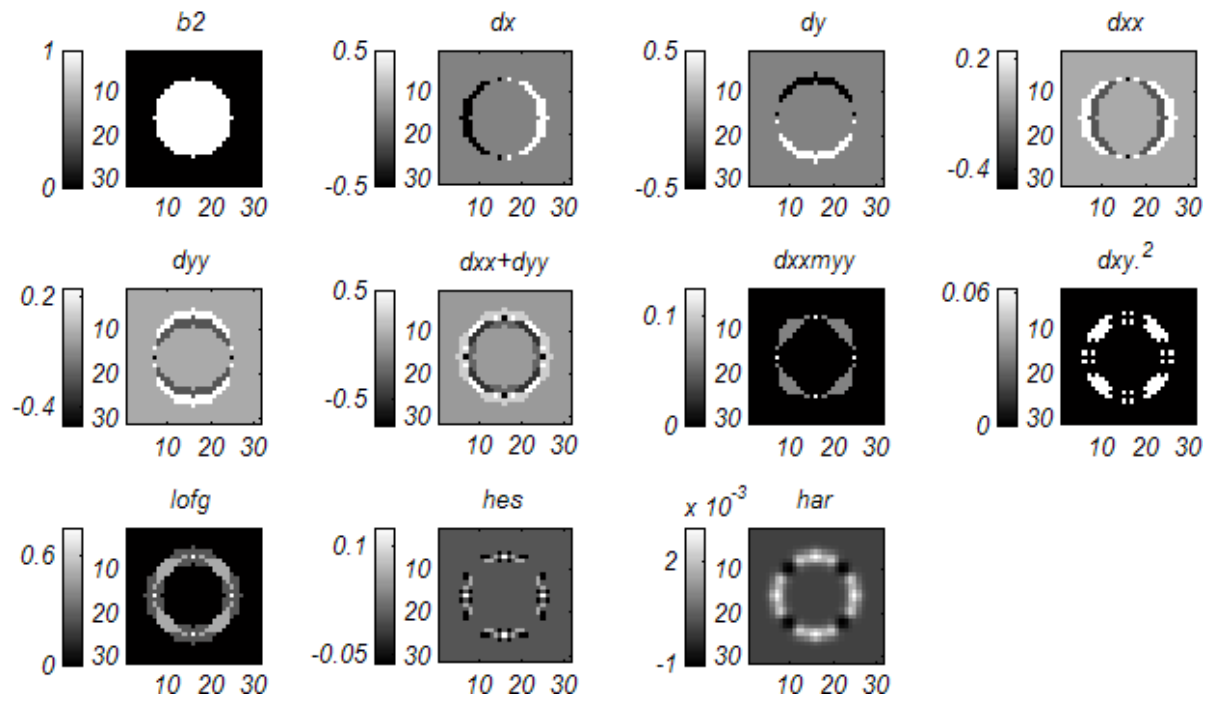
Aşağıda ilgi görebileceği düşüncesi ile bir kısım imgelerin bu operatörlere verdiği cevaplar görülmektedir. Bu şekillerde kullanılan orijinal imgelerden “a2” a’nın iki kat kenar uzunluğuna sahip bir kopyası, “b2” a2 ile aynı boyutlarda ancak içinde bir daire bulunan bir resim, “asmooth”, “a2smooth” ve “b2smooth” ise bunların bir Gaussian ile yumuşatılmış halleridir.



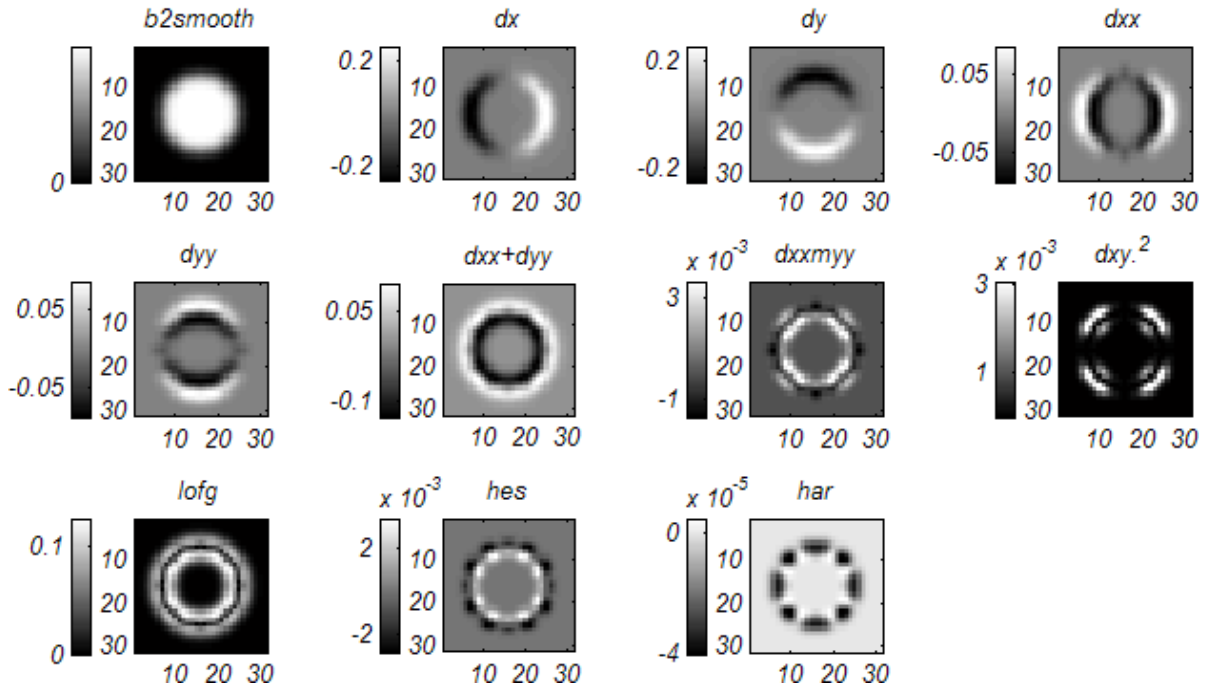
(a)



(b)



(c)



(d)

Şekil 14 Farklı suni imgelerin farklı türev operatörleri ile elde edilen çıktıları.

Şekil 14d deki “har”, yani Harris fonksiyonu çıktısı dikkat çekmektedir. Piksel değerleri bilinmeden bakıldığında fonksiyonun daire kenarları haricinde nerede ise her pikselde nokta bulunduğu düşünülebilir. Ancak piksel değerlerine bakıldığında maksimanın sıfıra çok yakın değerler olduğu görülür. Maksima belirlemeden önce pozitif bir eşik uygulanması halinde imgede hiçbir ilgi noktası bulunmayacaktır, ve bu orijinal imge bir daire olduğu için doğrudur.

Dolgu Yöntemleri

Birçok bilgisayarla görü uygulamasında imgelerin ölçek uzaylarının çıkarılması gerekmektedir. İmgelerin ölçek uzayları Gauss benzeri filtrelerle bir çok ardışık konvolüsyon ile elde edilir. Konvolüsyon işlemi ise sonucu oluşan çıktıların kenarlarını bozmaktadır.

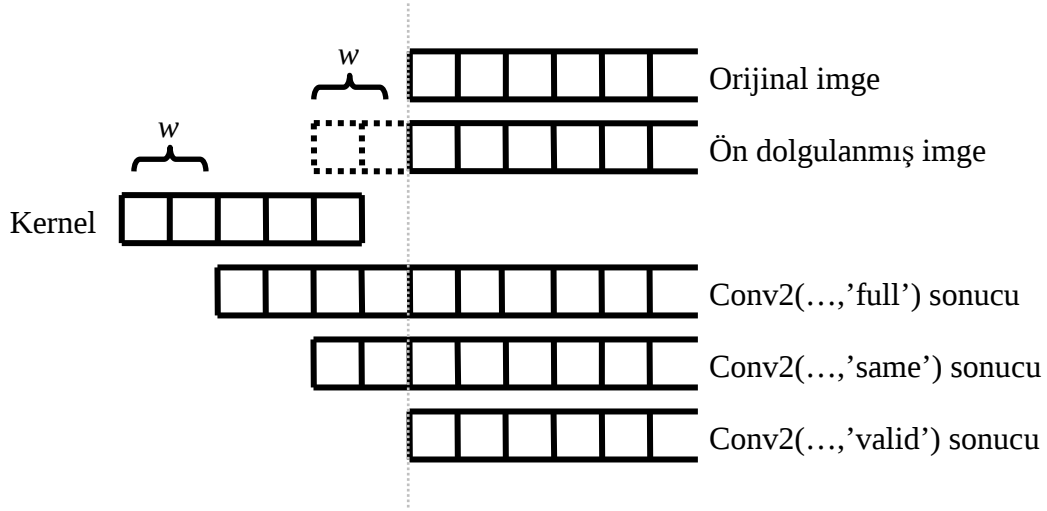
Ölçek uzayının üst katmanlarında imgeler küçülmekte, bu nedenle küçük filtrelerin bile kenarlardan getirdikleri bozukluklar imgeyi büyük oranda bozmaktadır. Ayrıca bu bozukluklar en alt katmandan itibaren birbirine eklenmekte ve en üst katmanlardaki resimlerin tamamen bozulmalarına neden olmaktadır. Bu nedenle imge *dolgulama yöntemleri* incelenmelidir.

Matlab’da dolgu yöntemini belirleme

Tez çalışmasında gerçekleştirme Matlab ile yapıldığı için Matlab konvolüsyon fonksiyonunun çalışma şekli önem kazanmaktadır. Matlab imgeyi konvolüsyon için geçerli sıfır ile dolgular ve, bizdeki versiyonunda, farklı bir değer ile dolgulama seçeneği vermez. Bu kısıt Matlab konvolüsyon fonksiyonuna (conv2) parametre olarak verilecek imgenin önceden istenen yöntem ile dolgulanması ve ‘shape’ parametresi olarak ‘valid’ seçilmesi ile aşılabılır:

conv2 fonksiyonunda konvolüsyonun döndüreceği kısım ‘shape’ parametresi ile belirlenir. Bu parametre ‘full’ değerini aldığı anda Matlab tüm konvolüsyonu döndürürken, ‘same’ değeri için konvolüsyonun orijinal imgesi ile aynı büyüklükte olan merkezi kısmını, ‘valid’ değeri için ise sadece kenarlar nedeni ile bozulmayan merkezi kısmını döndürür. İmge her kenarda kullanılacak operatörün o yöndeki boyutunun yarısı kadar dolgulanır ve sonra ‘valid’ seçeneği

kullanılırsa imgenin istenen değer ile dolgulanması sağlanmış olur. Şekil 15 bu ön dolgulamayı betimlemektedir.



Şekil 15 Ön dolgu ile konvolüsyon. w ön dolgunun genişliğidir

Dolgu yöntemlerinin karşılaştırılması

Bir imge konvolüsyona hazırlık amacı ile birkaç farklı şekilde dolgulanabilir. En çok kullanılan yöntemler sıfırla dolgulama, kenarın simetriğini kullanma ve kenar piksellerinin tekrarlanmasıdır. Bu yöntemlerin hangisinin imgeyi kenar bozulmasından daha iyi koruduğunu öğrenmek amacı ile karşılaştırmalar yapılmıştır. Karşılaştırmalarda sıfırla dolgulama “zero”, kenar simetriğini kulanma “sym”, kenar piksellerinin tekrarlanması ise “edge” ismi ile belirtilmiştir. Bu yöntemlerin yanı sıra dolgu değerlerinin arkadaki piksel değerlerinden polinomsal ekstrapolasyon ile eldesi de değerlendirilmiştir. İsim olarak “slope” 1. dereceden (doğrusal), “poly_2” 2. dereceden, “poly_3” ise 3.dereceden polinomal fit kullanılarak yapılan dolgulama işlemini belirtmek için kullanılmıştır.

Dolgu yöntemlerinin orijinal imge yardımı ile karşılaştırılması

Öncelikle aynı imgenin farklı dolgulama yöntemleri yardımı ile ölçek uzayları çıkarılmış, ve uzaylarda oluşan düzlemler (plane) ile orijinal imgenin farkı elde edilmiştir. En az farkı sağlayan en başarılı yöntem kabul edilebilir. Sonuçlar Çizelge 1 ile özetlenmiştir. “sym” ve “edge” yöntemleri bu karşılaştırma için tamamen aynı sonuçları vermişlerdir. Bunun nedeni ölçek uzayı çıkaran kod parçasının düzlemleri birbirinden 3×3 ’lük filtreler üretmesidir. Filtre 3×3 olduğu için dolgu genişliği $(3-1)/2=1$ olmakta, dolayısı ile kenarın tekrarı ile simetriğinin alınması arasında bir fark olmamaktadır. Yöntemlerin başarı sırası “edge”/“sym”, “slope”, “zero”, “poly_2”, “poly_3” olmuştur.

Çizelge 1 Farklı dolgu teknikleri ile elde edilen ölçek uzayı düzlemlerinin (plane) orijinal imgeden piksel başına farkları. Küçük değerler daha iyidir.

	edge	zero	sym	slope	poly_2	poly_3
plane_2	0.0058516	0.0062797	0.0 58516	0.0059926	0.0064509	0.01067
plane_3	0.012177	0.014801	0.01217	0.012118	0.011891	0.023864
plane_6	0.019903	0.024067	0.019903	0.020233	0.019729	0.02798
plane_7	0.026927	0.037159	0.026927	0.025674	0.026763	0.027433
plane_10	0.037642	0.057531	0.037642	0.03953	0.037941	0.034999
plane_11	0.046174	0.077065	0.046174	0.049069	0.044247	0.1334
plane_14	0.058266	0.09732	0.058266	0.06695	0.059606	0.19979
plane_15	0.06783	0.11839	0.06783	0.077941	0.071428	0.22963
plane_18	0.088828	0.14636	0.088828	0.10415	0.11849	0.31354
plane_19	0.10075	0.17248	0.10075	0.13273	0.26301	0.31394

Aynı karşılaştırma sırasında her bir dolgu yöntemi ile ölçek uzayı hazırlamada geçen zamanlar da not tutulmuştur. Çizelge 2 sonuçları özetlemektedir:

Çizelge 2 Ölçek uzayı hazırlanması için piksel başına geçen zaman.

edge	zero	Sym	slope	poly_2	poly_3
0.71734	0.2988	0.27181	0.37379	33.7822	33.7046

Sonuç olarak “sym” yöntemi hem daha hızlı çalışması hem de en az bozmaya neden olması ile öne çıkmaktadır. Bu aşamada kesin bir seçim olarak görülmektedir. Aksine “poly_2” ve “poly_3” hem çok yavaş çalışmakta hem de büyük bozulmaya neden olmaktadır. Bu nedenle bundan sonraki karşılaştırmalarda bu yöntemlerin değerlendirilmemesine karar verilmiştir.

Dolgu yöntemlerinin ideal düzlemler yardımı ile karşılaştırılması

Bir önceki karşılaştırmada ölçek uzayı düzlemlerinin *orijinal imge* ile farklarına bakılması eleştirilebilir. Ölçek uzayı düzlemleri ideal (kenar bozulmasına uğramamış) düzlemler ile nasıl karşılaştırılabilir? Öncelikle böyle ideal düzlemlerin elde edilmesi gerekmektedir. Bunu başarmak için öncelikle büyük bir imge ve ölçek uzayı oluşumunda kenar bozulmalarının ulaşamayacağı merkezi kısmı iki ayrı imge olarak alınmıştır. Her iki imgeden de aynı yöntemler ile dolgularak ölçek uzayları oluşturulmuştur. Küçük imgeden oluşturulan düzlemlerin büyük imgeden oluşturulan düzlemlerde karşılık gelen bölge ile farkı hesaplanmıştır. Başka bir deyişle büyük imge düzlemlerinin merkezleri küçük imge düzlemleri için groundtruth görevini üstlenmiştir. Yöntemlerin başarı sıralaması Çizelge 1 de gösterilen önceki karşılaştırmadan farklı çıkmamıştır. Bu sonuçlar Çizelge 2 de özetlenmiştir. Küçük değerler daha iyidir.

Çizelge 3 Farklı dolgu teknikleri ile elde edilen ölçek uzayı düzlemlerinin (plane) bozulmamış (groundtruth) ölçek uzayı düzlemlerinden piksel başına farkları

	poly_2	edge	zero	sym	slope
plane_2	6.7333e-005	5.6315e-005	0.0027989	5.6315e-005	5.3294e-005
plane_3	0.00015217	0.00010798	0.0044998	0.00010798	0.00013242
plane_6	0.0091629	0.0090336	0.01579	0.0090336	0.0091305
plane_7	0.0094148	0.009146	0.01871	0.009146	0.0093254
plane_10	0.023089	0.01021	0.02482	0.01021	0.0106
plane_11	0.028535	0.011177	0.028982	0.011177	0.011897
plane_14	0.040048	0.034028	0.053781	0.034028	0.035348
plane_15	0.064391	0.036817	0.054608	0.036817	0.039807
plane_18	0.046668	0.039707	0.053048	0.039707	0.04551
plane_19	0.10682	0.061959	0.067425	0.061959	0.07635
plane_22	0.26628	0.093748	0.081783	0.093748	0.15117
plane_23	0.40324	0.094574	0.076656	0.094574	0.21392
plane_26	0.34646	0.13731	0.14359	0.13731	0.24207
plane_27	0.38537	0.14359	0.1446	0.14359	0.2432

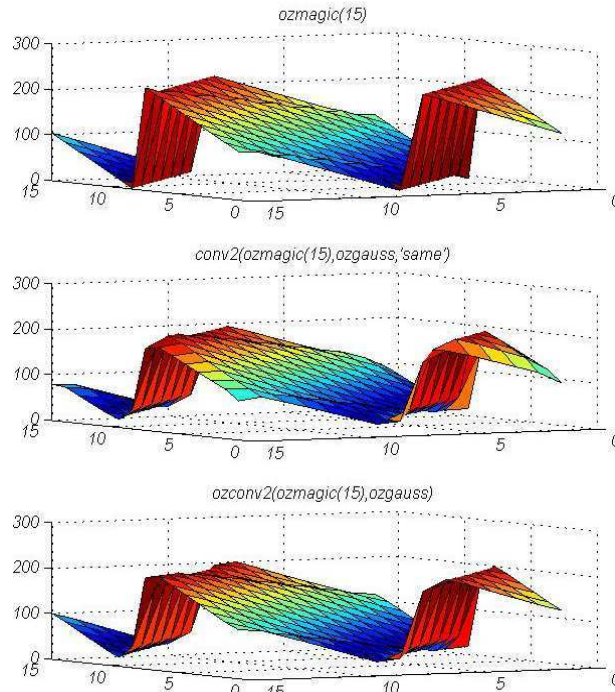
“edge” ve “sym” dolgu yöntemlerinin karşılaştırması

Her ne kadar bizim uygulamamız için “sym” ve “edge” yöntemleri aynı başarıyı verse de daha geniş filtreler kullanan bir uygulamada hangi dolgu yönteminin başarılı olacağı gözlemlenmek istemiştir. Bu nedenle üçüncü karşılaştırmada ölçek uzayının her bir düzlemi (plane) direkt büyük bir Gaussian ile elde edilmiş ve yöntemlerin başarıları karşılaştırılmıştır. Karşılaştırma bir önceki karşılaştırmada olduğu gibi her düzlemin groundtruthu ile arasındaki fark ölçülerek yapılmıştır. “sym” yöntemi (kenarlara yakın piksellerin simetriğinin alınması) bir kere daha, “edge” (kenar piksellerinin tekrarlanması) dahil diğer yöntemlerden başarılı sonuçlar vermiştir. Çizelge 4 karşılaştırma sonucunu özetlemektedir. . “edge” ve “sym” yöntemleri arasındaki farkın görülebilmesi için 3x3 değil, büyük konvolüsyon filtreleri kullanılmıştır. Küçük değerler daha iyidir.

Çizelge 4 Farklı dolgu teknikleri ile elde edilen ölçek uzayı düzlemlerinin (plane) bozulmamış (groundtruth) ölçek uzayı düzlemlerinden piksel başına farkları

	edge	zero	sym	slope
plane_2	0.00063073	0.015182	0.00061138	0.018082
plane_3	0.0011241	0.034549	0.00092942	0.04799
plane_4	0.00213	0.042165	0.0012711	0.069435
lane_5	0.0030425	0.045865	0.0016941	0.082324
plane_6	0.0041532	0.055382	0.0018619	0.10084
plane_7	0.0042484	0.062957	0.0017732	0.17654
plane_8	0.0038747	0.06794	0.0014702	0.22946
plane_9	0.0031023	0.072004	0.0016884	0.27129

Bu sonuçların ışığında Matlab'ın konvolüsyon fonksiyonu “conv2” yerine yerine “sym” yöntemi ile öndolgu kullanan bir “ozconv2” fonksiyonu yazılmıştır. Şekil 16 conv2 ve ozconv2'nin bir imgeye cevabını karşılaştırmaktadır.



Şekil 16 conv2 ve ozconv2 fonksiyonlarının karşılaştırılması.

Şekil 16 de en üstteki grafik orijinal imgenin parlaklık değerlerini, ortadaki grafik Matlab standart conv2 fonksiyonunun cevabını, en alttaki ise “sym” yöntemi ile dolgu kullanan ozconv2 fonksiyonunun cevabını göstermektedir. Kenarlardaki bozulmanın ne kadar azaldığı dikkat çekmektedir.

Fiziksel Olgular Yardımı ile SIFT Noktalarının Hareketlerinin Simülasyonu

Doktora çalışması sırasında imgelerdeki şekillerin yıldız – meteor – yerçekimi – yay gerilimi gibi fiziksel olgulardan yola çıkılarak oluşturulan yöntemler yardımı ile birbirine eşlenmesi yönünde çalışmalar yapılmıştır. Ayrıca eşleme yöntemleri bir şekil sınıflandırma çalışması için de kullanılmıştır.

Test imgesindeki her bir SIFT noktası iki boyutlu hipotetik bir uzayda bir yıldız, eğitim imgesindeki her bir SIFT noktası ise yine iki boyutlu hipotetik bir uzayda bir meteor gösterilsin. Yıldız sistemi ile meteor sistemi aynı iki boyutlu uzayda üst üste başlatılsın. Yıldızlar hareketsiz, meteorlar ise hareketli sayılsın. Meteorlar arasında farazi yayların bulunduğu, yıldızların ise meteorları yerçekimi ile çektiği varsayılsın. Meteorlar yay gerilimleri ve yıldız çekimlerinin birbirini dengeleyeceği bir noktaya hareket edecektir. Sistemin titreşimlerinin azaltılması için uzayda sürtünme kuvvetinin olduğu varsayılabilir.

Yıldız – Meteor Arası Yerçekimi

Bu çalışmada bir meteor ile bir yıldız arasındaki yerçekimi için

$$G = ae^{\frac{u^2}{b^2} - \frac{x^2}{c^2}} + d \quad (8)$$

formülü kullanılmıştır. Burada a , b , c ve d sabitler, u iki nokta arasındaki benzerlik uzaklığı (128 boyutlu SIFT özellikleri arasındaki uzaklık), x ise iki nokta arasındaki iki boyutlu uzamsal uzaklıktır. Gauss fonksiyonuna benzeyen bu fonksiyonun tercih edilme sebebi hiçbir zaman sonsuza ıraksamamasıdır. Hatırlanacağı üzere standart Newton yerçekimi formülü (9) ile verilir:

$$G = \frac{gm_1m_2}{x^2} \quad (9)$$

Bu formülün paydasındaki x terimi (uzamsal uzaklık), meteor yıldızın tam üzerine geldiğinde sıfır olmakta, G yi ise sonsuz yapmaktadır. Simülasyonu süren denklemlerde bir değişken sonsuz değeri aldığı anda simülasyon doğru şekilde ilerlemeyebilmektedir.

Newton yerçekimi formülündeki kütle m yerine ise noktaların benzerliği u (SIFT özellikleri arasındaki uzaklık) kullanılmıştır.

Meteor – Meteor Arası Yay Çekimi

Meteorlar arasında yaylar olduğunu varsayarsak, simülasyon ile yerleri değişmekte olan iki meteor arasında onları doğal uzaklıklarına döndürmeye çalışan bir yay kuvveti (yay gerilimi) oluşacaktır. Bu gerilim standart yay kuvveti formülü ile bulunabilir:

$$P = -k * \Delta x \quad (10)$$

Bu formülde P terimi iki meteor arasındaki gerilimi, k yay sabitini, Δx ise iki meteor arasındaki şimdiki uzaklığın aynı meteorlar arasındaki doğal uzaklıktan farkını göstermektedir.

Meteor Başlangıç Konumları

Bu altı aylık sürenin başında simülasyon üzerinde düşünülmeye başlanmış ve yıldız sistemi arasına normal konfigürasyonunda (aralarındaki uzaklıklar normal uzaklıkta) yerleştirilen meteor sisteminin yıldızların çekimi sonucu hareketlenmesi, ve meteorların kendisine en çok benzeyen yıldızla doğru ilerletilmesi planlanmıştır. Bu doğrultuda sentetik noktalar ile bir çok denemeler ve simülasyonlar yapılmıştır. Aşağıda yapılan simülasyonlardan elde edilen

görüntülerden örnekler verilecektir.

Ancak sınıflama uygulamasına geçildiğinde bu yaklaşımın önemli bir sıkıntısı olduğu görülmüştür: Her bir meteorun yıldız sisteminde başlayacağı noktanın doğru olarak seçilmesi çok zordur. Basitçe meteor sisteminin merkezinin yıldız sisteminin merkezine konması düşünülebilir. Ancak bu yöntemin ne kadar başarılı olacağı soru işaretidir. Ayrıca burada yıldız sisteminin yöneliminin ne olacağı, hangi ölçeğin kullanılması gerektiği gibi sorunlar da çıkmaktadır. Yanlış noktadan başlanan simülasyonun yanlış yerde son bulma olasılığı çok büyüktür.

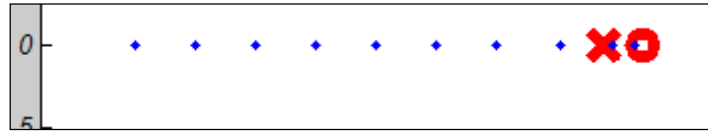
David Lowe (Lowe, 2004) SIFT noktaları ile şekil tanım çalışmasında Hough Transformu ile belli sayıda aday başlama noktası belirlemekte ve daha sonra bunları elemektedir. Bu yaklaşım ileride denenebilir.

Bu raporun ileriki bölümlerinde anlatılacak tanıma/sınıflama çalışmamızda ise meteorlar direkt kendileri ile eşleşen yıldızların üzerinde başlatılır ve yay kuvvetleri altında başlangıç noktalarından uzaklaşmalarına izin verilir. Bu noktada farklı kriterler ayırt edici / uzaklık ölçüsü olarak kullanılabilir. Şimdilik yaylarda biriken ortalama gerilim kullanılmaktadır. Ancak örneğin simülasyon sonucunda meteorların yer değiştirme miktarı da uygun bir kriter olabilir. En uygun kriter sonraki dönemde araştırılabilir.

Bu çalışmada şimdilik meteorların başlama noktası olarak direkt kendileri ile eşlenen yıldızlar seçilmiştir.

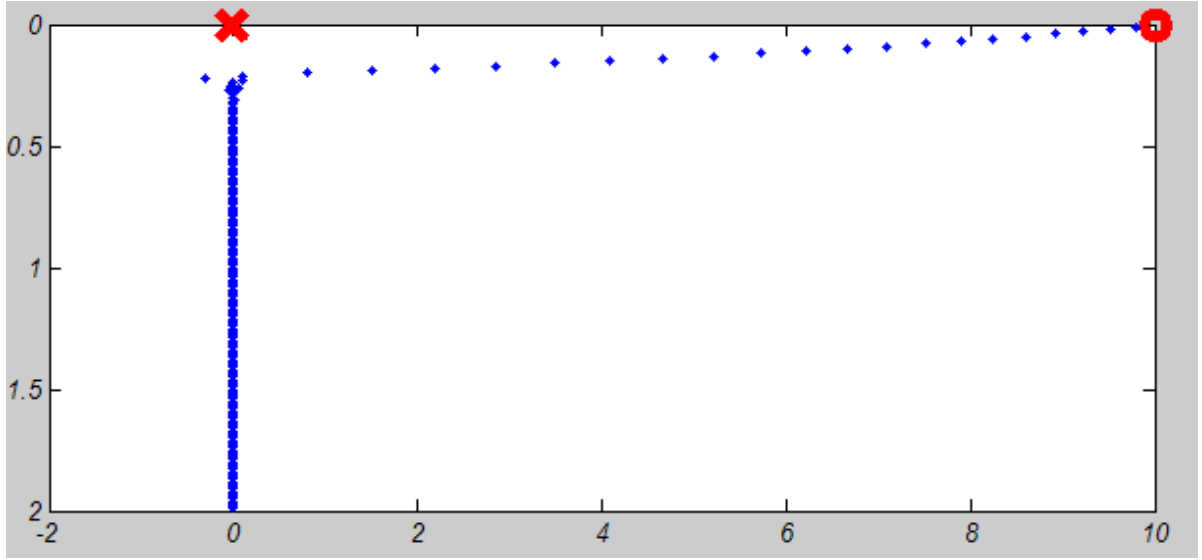
Örnek Simülasyon Çıktıları

Şekil 17 de bir meteorun hareketi görülmektedir. Çarpı yıldızı, yuvarlak meteor başlangıç noktasını, noktalar ise meteorun her iterasyondaki yerini göstermektedir. Meteor yaydan fırlayan ok gibi davranmaktadır.



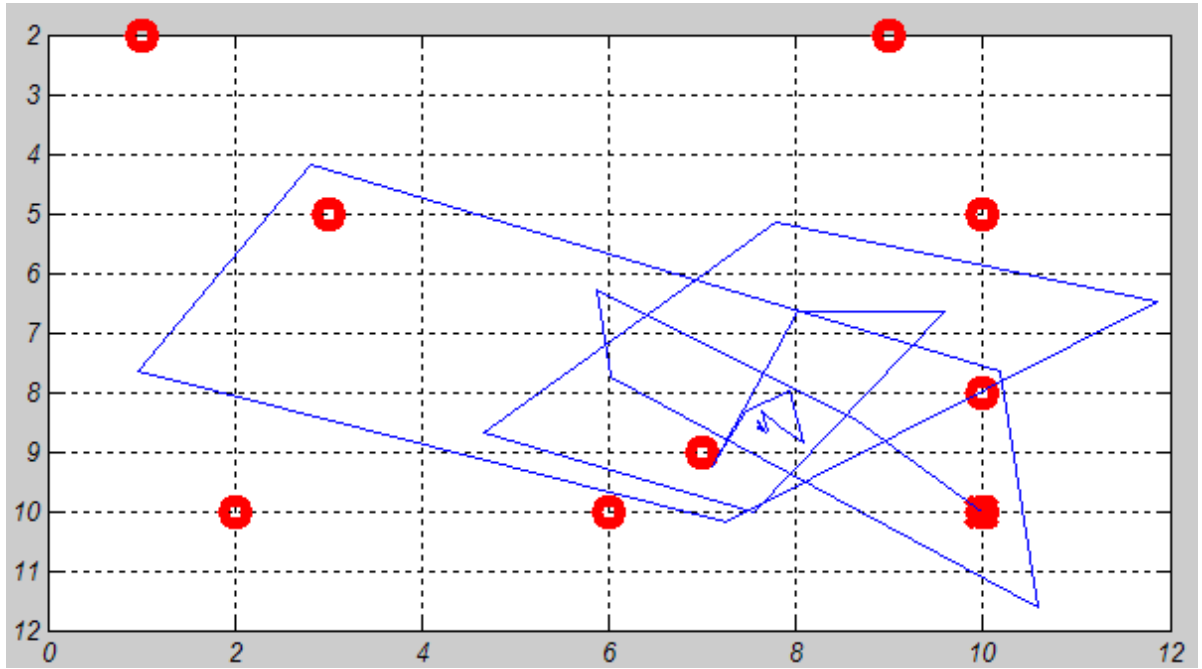
Şekil 18 Bir meteorun hareketi

Şekil 19 bir meteorun zamana göre hareketini göstermektedir. Çarpı yıldızı, yuvarlak meteor başlangıç noktasını, noktalar ise meteorun her iterasyondaki yerini göstermektedir. Meteor aslında tek boyutta hareket etmektedir. Dikey eksen zamanı göstermektedir. Önceki simülasyonda olmayan sürtünme kuvveti eklenmiştir. Bu yay – ok etkisini azaltmıştır.



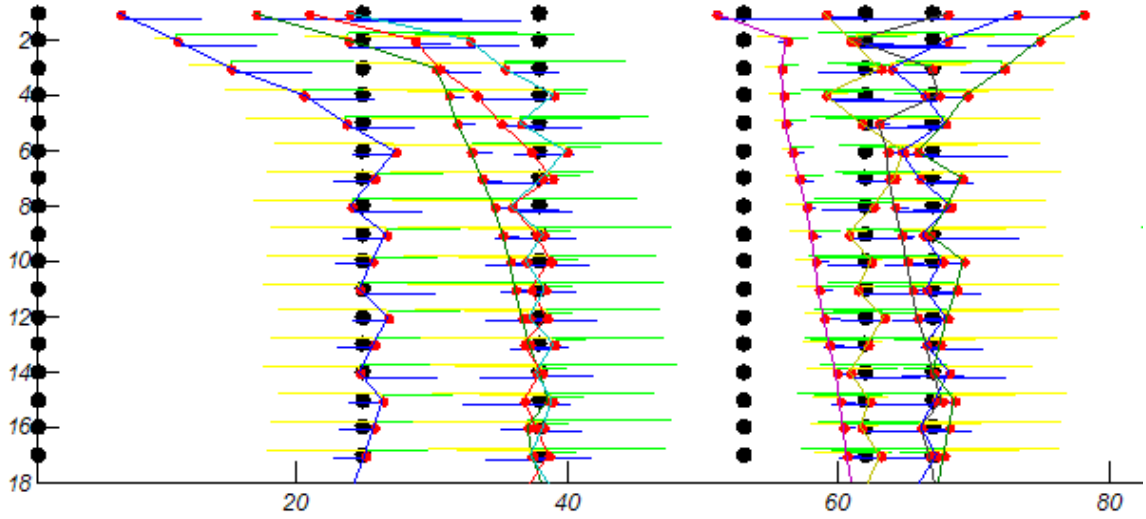
Şekil 20 Zamana göre meteorun hareketi

Şekil 21 da iki boyutlu meteor hareketine izin veren simülasyonlardan birinin çıktısı görülmektedir. Bu çıktıda yuvarlaklar yıldızları, çarpı meteor başlangıcını, mavi yol ise meteorun izlediği yolu göstermektedir. Bu örnekte meteor yıldız kümesinin merkezine yakınsamıştır. Bu çıktı simülasyonunun kümeleme amaçlı da kullanılabileceğini göstermektedir.



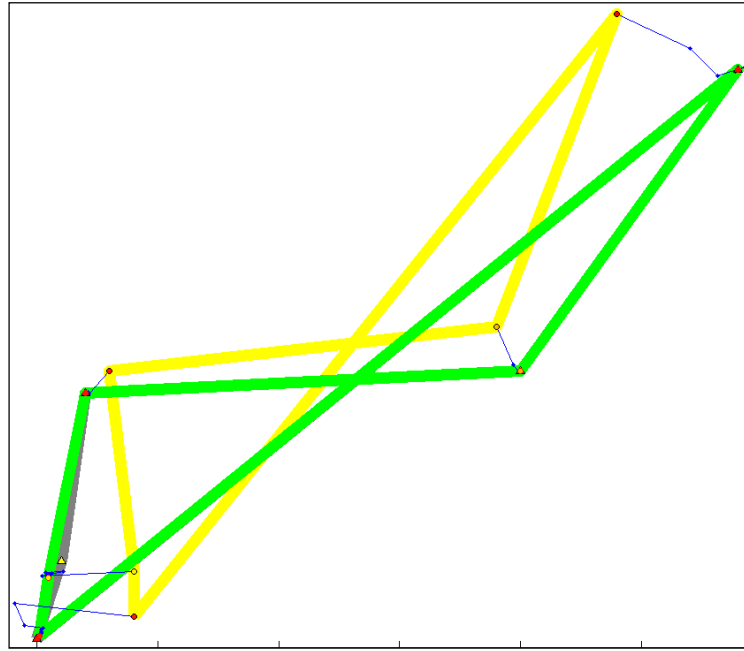
Şekil 22 İki boyutlu meteor hareketine gösteren bir simülasyon çıktısı

Şekil 23 Yay kuvvetinin sisteme etkisini göstermektedir. Dikey eksen zamanı (iterasyonu) göstermektedir. Siyah noktalar yıldızları, hareketli küçük noktalar meteorları simgelemektedir. Meteorların hareketlerinin birbirine karışmaması için her bir meteorun zaman ekseninde yolunu belirten çizgiler konulmuştur.



Şekil 24 Yay kuvvetinin sisteme eklendiği ilk simülasyonlardan birinin çıktısı

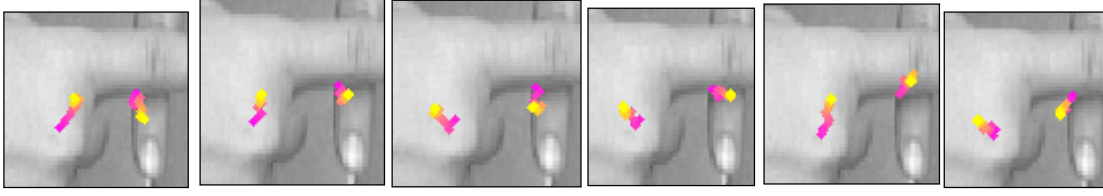
Şekil 25 Sentetik verilerle yapılan iki boyutlu bir simülasyon çıktısını göstermektedir. Üçgenler yıldızları, yuvarlaklar meteorları, ince çizgiler meteorların izlediği yörüngeleri göstermektedir. Aradaki kalın çizgiler hipotetik nesnenin kenarlarını betimlemektedir. Bu simülasyonda meteorların yakınsayabilmesi için kendisine etkiyen kuvvet en yakınındaki yıldız uzaklığı ile ağırlanmıştır. Dolayısı ile meteor bir yıldızla yaklaştıkça daha az yer değiştirmektedir.



Şekil 26 Sentetik verilerle yapılan iki boyutlu simülasyon çıktısı

Şekil 27 bir el resmi üzerindeki yıldızlara konulan meteorların aynı başlangıç noktasından başlayarak gerçekleştirdikleri altı simülasyon çıktısını göstermektedir. Meteor ve yıldızlar gösterilmemiş, sadece meteorların yörüngeleri gösterilmiştir. Yörüngeler kırmızı ile başlamakta sarı ile son bulmaktadır. Bu simülasyonlarda her iterasyonda cisme sadece rastgele bir başka meteorla arasındaki yay gerilimi etki ettirilmiştir. Ayrıca içinde bulunulan iterasyonda yeri değişen meteorlar da rastgele seçime girebilmektedir. Rastgelelik nedeni ile her simülasyon sonucu biraz farklı noktalarda oluşmakta ancak yine de benzer noktalarda

oluşmaktadır. Bu yöntemle 1-her iterasyon daha az zamanda tamamlanabilmektedir, çünkü her iterasyonda her meteor için sadece bir yay kuvvetine bakılmaktadır. 2- meteorlar doğru yerlere daha hızlı yayılmaktadır, çünkü yeni yeri hesap edilen meteorlar da o iterasyon içerisindeki diğer meteorların yerlerini etkileyebilmektedir. 3-meteor sistemi kayması daha az olmaktadır.



Şekil 28 Bir el resmi üzerindeki yıldızlara konulan meteorların aynı başlangıç noktasından başlayarak gerçekleştirdikleri altı simülasyon çıktısı

Fiziksel Olgular Yardımı ile Şekil Tanıma Çalışması

Bu şekil tanıma çalışmasında eldeki Türk İşaret Dili Alfabesi videolarından (Altun ve Albayrak, 2006) anahtar kareler çıkarılmış, bu anahtar kareler üzerinden SIFT noktaları çıkarılmış, ve SIFT noktalarının oluşturduğu yıldız – meteor – yay sistemleri yardımı ile anahtar kareler sınıflandırılmıştır.

Şimdilik yüzde 25-35 arası tanıma hatasına inilebilmektedir.

Sınıflandırma Yöntemi

Sınıflandırma için 1NN sınıflandırma yöntemi kullanılmıştır. Uzaklık ölçüsü ileride anlatılacaktır.

Veritabanı, Eğitim ve Test Setleri

Üzerinde çalışılan veritabanında dört farklı işaretçi tarafından icra edilmiş Türk İşaret Dili harfleri videoları vardır. Toplamda $4 \times 29 = 116$ harf videosu bulunmaktadır. Örnek sayısının azlığından dolayı test ve eğitim setleri *k-fold cross validation* yöntemi ile elde edilmiştir. Bunun için sırası ile her bir işaretçinin videoları test seti olarak seçilmiş, geri kalan $3 \times 29 = 87$ video ise eğitim seti olarak kullanılmıştır. Bu yaklaşımda test ve eğitim setleri her zaman için farklı işaretçiler ile oluşturulmaktadır. Bu da geliştirilen yöntemin işaretçiden bağımsız çalışabildiğini göstermektedir.

Anahtar Karelerin Çıkarılması

Türk İşaret Dili Alfabesi harflerinin tanınması için her bir videodan çıkartılan anahtar kareler yeterlidir. Altun ve Albayrak 2006 (Altun ve Albayrak, 2006)daki gibi bu çalışmada da bu anahtar kare bir sonrakine en çok benzeyen video karesi olarak seçilmiştir.

SIFT Noktaların Çıkarılması

Anahtar karelerin üzerinden SIFT noktaları (Lowe, 2004) SIFT yönteminin geliştiricisi David Lowe'nin kendi web sayfasında yayınlanan Matlab kodu çalıştırılarak elde edilmiştir.

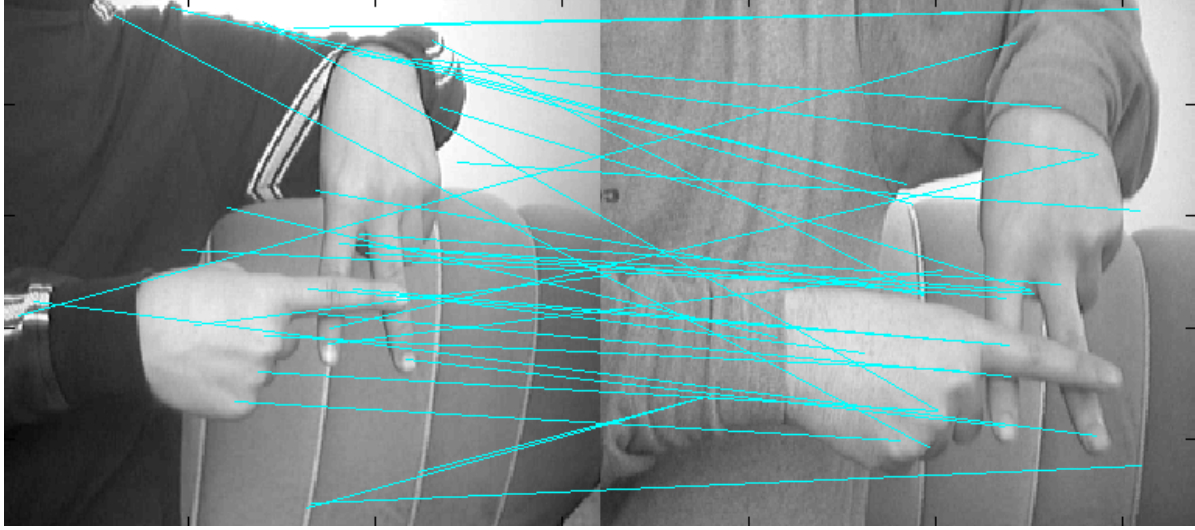
Eğitim İmgelerinin Noktalarının Sadece Geçerli Alan İçine Sınırlandırılması

Eldeki imge veritabanında yer alan işaretler oldukça dikkatli bir şekilde oluşturulduğu için bunlar içindeki el bölgeleri ten renginden yararlanılarak kolaylıkla bulunabilmektedir. Bu çalışmada da Altun ve Albayrak 2006 (Altun ve Albayrak, 2006)da kullanılan yöntemle YCrCb uzayındaki renk değerlerinden faydalanan ten bölgeleri bulunmuş ve eğitim noktaları bu bölgeler içindekilerle sınırlandırılmıştır. Test imgelerinde noktalara böyle bir

sınırlama getirilmemiştir.

Test ve Eğitim İmgelerinde Birbiri İle Eşleşen SIFT Noktalarının Bulunması

Lowe'nin yayınladığı kod büyük oranda bu bölümde de kullanılmıştır. Bu kodda eşleşecek imgelerden birindeki SIFT noktaları sırası ile ziyaret edilir ve diğer imgedeki tüm noktalar eşleme adayı olarak kabul edilir. En yakın adayın benzerlik uzaklığı d_1 ve en yakın ikinci adayın benzerlik uzaklığı d_2 olsun. $d_1 < 0.80d_2$ ise d_1 uzaklığındaki en yakın aday eşlenmiş olur.



Şekil 29 A harfini gösteren iki anahtar kare üzerinde birbiri ile eşleşen SIFT noktaları.

Yukarıdaki yöntemde birinci imgedeki noktalardan bazıları ikinci imgedeki aynı nokta ile eşlenebilmektedir. Bu nedenle bu çalışmada eşlenen noktalar ikinci kere ziyaret edilmekte ve sayını noktaya eşlenen noktalardan biri hariç hepsi elenmektedir. Böylece iki imge arasındaki nokta eşlemesinin bire-bir hale gelmesi sağlanmaktadır.

Az Sayıda Nokta Eşlenmiş İse İmgelerin Kullanılmaması

Aynı sınıftan olmayan el işaretlerini içeren imgeler arasında eşlenen nokta sayısı az olmaktadır. Bu nedenle belirli sayıdan az sayıda nokta eşlemesi gerçekleşmiş ise bu imgelerin aynı sınıftan olmadıklarına karar verilmektedir.

Meteor Sisteminin Doğal Uzaklıklarının Çıkarılması

Eğitim imgesindeki noktaların, başka bir deyişle meteor sistemindeki meteorların, kendi aralarındaki uzaklıklar çıkarılarak doğal uzaklık olarak kaydedilir.

Meteorların Eşleştikleri Yıldızların Üzerine Konulmaları

Her bir meteor diğer imgede eşleştiği yıldızın üzerine konulur. Eğer iki imge benzer el işaretlerini barındırıyorsa meteorlar arasındaki uzaklıklar doğal uzaklıklara yakın olacak, aksi halde ise yeni uzaklıklar çok farklı olacaktır.

İki Meteor Arası Yay Geriliminin Hesaplanması

Meteorlar arasında yaylar olduğunu varsayarsak, farklı el işareti içeren bir imgeden çıkarılan noktalar(yıldızlar) üzerine konulan meteorlar arasındaki yaylar üzerinde çok miktarda yay gerilimi birikecektir. Bu gerilim (11) ile bulunabilir:

$$P = -k * \Delta x \quad (11)$$

Bu formülde P terimi iki meteor arasındaki gerilimi, k yay sabitini, Δx ise iki meteor arasındaki şimdiki uzaklığın aynı meteorlar arasındaki doğal uzaklıktan farkını göstermektedir.

Sistemdeki Ortalama Yay Geriliminin Tahmin Edilmesi

Sistemdeki ortalama yay geriliminin *tam olarak hesaplanması* için her bir meteora etkiyen toplam yay gerilimini bulmak, daha sonra bunların ortalamasını almak gerekir. Herhangi bir meteora etkiyen toplam yay gerilimini bulmak için ise o meteorun tüm diğer meteorlar ile yeni uzaklığını hesaplamak eski uzaklıktan farkını bulmak ve ona etkiyen toplam yay gerilimini bulmak gerekecektir. Bu yaklaşım çok fazla sayıda hesaplama gerektirir.

Bu çalışmada sistemdeki tüm meteorlar ziyaret edilir, her biri için rastgele bir başka meteor seçilir ve önce ikisi arasındaki eski – yeni uzaklık arasındaki fark, sonra da gerilim hesap edilir. Ortalama yay gerilimi eldeki gerilimlerin ortalaması ile tahmin edilir. Bu yolla sistemdeki ortalama yay gerilimi çok daha az işlemci gücü kullanarak hızlı bir şekilde *tahmin edilmektedir*.

Bu sayede örneğin 1000 noktalı bir sistemde hesap süresi 1000 de bire düşecektir.

Sadece bir farklı meteor yerine n farklı meteorun yay gerilimi de kullanılabilir.

Bu çalışmanın önemli bir katkısı bu hızlandırmadır.

Simülasyonsuz Sınıflama

İmgeler arası uzaklık ölçüsü olarak ortalama yay gerilimi kullanılarak yapılan 1NN sınıflaması sonucu hata %25 ile % 35 arasında değişmektedir. Daima farklı noktalar ile yay gerilimi hesaplandığından her bir çalıştırmada hata farklı değer alabilmektedir.

Simülasyonlu Sınıflama

Yıldız – meteor benzetmesinden yola çıkarak ve sisteme yıldızlar ve meteorlar arasında yer çekimi eklenerek meteorların hareketleri simüle edilebilir. Simülasyon sonucunda sisteminin üzerindeki yay gerilimini biraz azaltması sağlanabilir. Bu şekilde yapılan denemelerde şimdilik simülasyonsuz sınıflamadan daha kötü sonuçlar alınmıştır. Ancak henüz simülasyon parametreleri kalibre edilememiştir. Simülasyon parametrelerini iyi kalibre ederek, veya ortalama yay gerilimi haricinde bir uzaklık ölçüsü kullanarak (örneğin ortalama yay gerilimi + ortalama yer çekimi) simülasyon ile daha iyi sonuçlara ulaşılması mümkün olabilir.

Sonuç

SIFT noktalarının hareketlerinin, yıldız – meteor yerçekimi, meteorlar arası yay çekimi gibi fiziksel olgular yardımı ile simüle edilmesi ve bu hareketler yardımı ile şekiller arasında sınıflama için kullanılabilecek bir uzaklık ölçüsünün geliştirilmesi üzerine çalışılmıştır. Simülasyonun daha az iterasyonla yakınsaması ve her iterasyonun daha az işlemci gücü ile (daha hızlı) başarılması için uzaklık ölçüsü tam olarak hesaplanmak yerine tahmin edilmektedir.

Yapılan sınıflandırma deneyleri ümit vericidir.

Gelecek dönemde sınıflandırma deneylerinde başarısız sonuç veren örnekler incelenerek daha iyi bir uzaklık ölçüsü bulunması ve simülasyon ile daha başarılı sonuçlar elde edilmesi planlanmaktadır.

ÖZGEÇMİŞ

Doğum tarihi	15.08.1974	
Doğum yeri	Kahramanmaraş	
Lise	1991-1992	Kahramanmaraş Lisesi
Lisans	1993-1997	Orta Doğu Teknik Üniversitesi Fen Edebiyat Fakültesi Fizik Bölümü
Yüksek Lisans	1999-2000	University Of Chicago Computer Science Department
Çalıştığı kurum	2001-Devam ediyor	YTÜ Elektrik – Elektronik Fakültesi Bilgisayar Mühendisliği Bölümü Araştırma Görevlisi

Yayınlar

Altun, O. and Albayrak, S., (2007), "Hızlı Hizalama, Kalın Kenar Bulma ve Cezalandırılmalı Korelasyon ile Türk İşaret Dili Alfabeti Tanıma", Sinyal İşleme ve İletişim Uygulamaları Kurultayı SİU 2007, Eskişehir.

Altun, O., Albayrak, S., Ekinci, A. and Bükün, B., (2006), "Turkish Fingerspelling Recognition System Using Axis of Least Inertia Based Fast Alignment", AI 2006: Advances in Artificial Intelligence, Sattar ve Kang, Springer Berlin / Heidelberg, 4304: 473-481.

Altun, O. and Albayrak, S., (2006), "Increasing the Effect of Fingers in Fingerspelling Hand Shapes by Thick Edge Detection and Correlation with Penalization", Advances in Image and Video Technology, Chang ve Lie, Springer Berlin / Heidelberg, 4319: 1133-1141.

Altun, O., Dursunoglu, N. and Amasyali, M. F., (2006), "Clustering Application Benchmark", Workload Characterization, 2006 IEEE International Symposium on.

Bilgin, G. and Altun, O., "Cardiac problem diagnosis using SPECT images with statistical neural networks", Artificial Intelligence and Soft Computing, IASTED/ACTA Press.

Bilgin, G. and Altun, O., (2006), "Cardiac Problem Diagnosis with Statistical Neural Networks and Performance Evaluation by ROC Analysis", Applied Electronics, 2006. AE 2006. International Conference on.