

**COST 292: Sayısal ođulortam Verisinin Anlamsal okkipli
Analizi
(Semantic Multimodal Analysis of Digital Media)**

Proje No: 104E019

Dođ.Dr. A. Aydın ALATAN

ARALIK 2008
ANKARA

ÖNSÖZ

19 ortaklı Avrupa Birliđi COST 292 projesi kapsamında gerekleřen bu alıřmanın temel amacı okluortam verilerinden otomatik olarak anlamsal bilgi ıkarmaya yneliktir. COST 292 oluřumuna katılım TBİTAK tarafından desteklenmiř olan bu proje ile mmkn olmuřtur.

İÇİNDEKİLER

ÖNSÖZ.....	2
İÇİNDEKİLER.....	3
TABLO LİSTESİ.....	4
ŞEKİL LİSTESİ.....	5
ÖZET.....	10
ABSTRACT.....	11
1. GİRİŞ.....	12
2. GÖRSEL ÖZNİTELİK ÇIKARMA.....	13
2.1. 2-B ŞEKİL TANIMLAYICILAR.....	13
2.2. 3-B ŞEKİL TANIMLAYICILAR.....	20
2.3. KAMERA HAREKET ANALİZİ.....	38
2.4. İMGE EŞLEME İÇİN İLGİ NOKTASI ÇIKARMA.....	50
3. BÖLÜTLEME.....	57
3.1. İMGE BÖLÜTLEME.....	57
3.2. KATMANSAL HAREKETLİ NESNE BÖLÜTLEMESİ.....	64
3.3. BASKIN KÜMELER İLE VİDEO ÇEKİM SINIRI BÖLÜTLEME.....	75
4. UYGULAMALAR.....	81
4.1. ŞEKİL BİLGİSİ KULLANARAK HEDEF TANIMA.....	81
4.2. GÖRÜNTÜLERDEN YAZI BULMA VE TANIMA.....	91
4.3. ÖRTÜŞEN ARAÇLARIN BÖLÜTLENMESİNDE ÇOKLU KAMERA KULLANIMI.....	121
4.4. TV YAYINLARINDAN LOGO TESPİTİ.....	128
4.5. VİDEO KOPYASI BULMA.....	134
5. SONUÇ VE ÖNERİLER.....	141

TABLO LİSTESİ

Tablo 1: Şekil 22’de verilen video kareleri için kamera hareket tespiti algoritması sonuçları	40
Tablo 2: Her üç yöntemin de öznelik değeri ayrışımı sırasındaki yürütüm zamanları	62
Tablo 3 – Baskın Kümeler Test Sonuçları.....	79
Tablo 4: Çeşitli Öznelik Çıkarma metodları ile sınıflandırıcıların ortak performansları	106
Tablo 5: Haar Dalgacık Dönüşümü seviyeleri ile çeşitli sınıflandırıcıların karşılaştırmalı performansları.	106
Tablo 6: Haar Dalgacık Dönüşümü ile elde edilen öznelik vektörleri ile çeşitli sınıflandırıcıların karşılaştırmalı performansları.....	107
Tablo 7: Sınırlandırma Kutusu Performans Analiz Sonuçları.....	107
Tablo 8: Eşikleme sonuçları	109
Tablo 9: Örnek OCR Sonuçları	113
Tablo 10: İstatistiksel Dil Modeli kullanılarak OCR sonuçlarının iyileştirilmesi.....	116
Tablo 11: İstatistiksel Dil Modeli kullanılarak OCR sonuçlarının iyileştirilmesi.....	117
Tablo 12: OCR, k-ortalamlar ve İDM sonuçları	118
Tablo 13: Opak ve yarı-saydam logo ayrımı..	133

ŞEKİL LİSTESİ

Şekil 1: Değişik m ve n değerleri için taban fonksiyonlarının gerçel (a) ve sanal (b) kısımları. Gösterim amacıyla, aradaki gri değerler, beyaz 1, siyah 0 olacak şekilde oluşturulmuştur.	14
Şekil 2: EÖU görüntüsünün oluşturulmasındaki aşamalar: (a) ikilik görüntü , (b) nesne çeperi ve çeperin dışbükey hale getirilmiş hali, (c) oluşturulan EÖU görüntüsü.....	17
Şekil 3 : Nesnenin en büyük içbükey alanları (a) ve ilgili EÖU tepeleri (b).	17
Şekil 4 : Orijinal eğri (a), döndürülmüş versiyonu (b), ve gürültülü versiyonu (c). İlgili EÖU görüntüleri eğrilerin altlarında verilmiştir.....	18
Şekil 5: 3B arama motorunun temel bileşenleri.....	20
Şekil 6: 3B formatların sınıflandırılması.....	21
Şekil 7: Nokta kümeleri, üçgen ağ, derinlik verisi ve voksel gösterimlerinden birer örnek.....	21
Şekil 8: 3B Tanıma Algoritmalarının Sınıflandırılması.....	25
Şekil 9: Prizmanın Genişletilmiş Gauss Görüntüsü.....	26
Şekil 10: Kabuk ve sector modeli ve kabuk, sektör ve bileşim modellerinin 2B projeksiyonları	27
Şekil 11: a. Radyal (α) ve aksiyal (β) uzaklık b. Spin görüntü oluşturma aşamaları. Şekiller [27]inci referanstan alınmıştır.....	27
Şekil 12: [11] ve [29]'da kullanılan özelliklerin gösterimi	28
Şekil 13: Temel bazı şekillerin şekil indis değerleri, Resim [32]inci referanstan alınmıştır.	29
Şekil 14: Doğru tanımda Hough Dönüşümü.....	29
Şekil 15: Parametrik doğru denklemi.....	29
Şekil 16: Bir modelin Küresel Harmonik betimleyicileri , Resimler sırasıyla [42] ve [43] numaralı referanslardan alınmıştır.....	31
Şekil 17: İlk sıra: Kale verisinin elde edildiği çoklu görüntüler, ikinci sıra: Kale görüntülerinden elde edilen nokta kümesi ve 3 bölümlü k-means sonucu, üçüncü sıra: Cevo resmi ve ve Cevo sahnesinden elde edilmiş nokta kümesi	33
Şekil 18: Hough dönüşümü ile elde edilen en iyi k-düzlem, <i>Kale</i> için $k=2,3$ ve <i>Cevo</i> için $k=1,2$	33
Şekil 19: Ayrık Fourier Dönüşümü ile yapılan çıkarım sonuçları (soldan sağa: Sorgu imgesi ve sırasıyla elde edilen eşleme sonuçları)	34
Şekil 20: Deneyleerde kullanılan veritabanı	34
Şekil 21: a), b) Yaklaşık 1 piksellik sağa yatay dönme (pan right) içeren ardışık video kareleri (100. ve 101. kareler). c), d) Kamera hareketi içermeyen ardışık video kareleri (200. ve 201. kareler). e), f) Kamera hareketi içermeyen fakat bağımsız hareket eden nesne barındıran ardışık video kareleri (32. ve 33. kareler).....	40
Şekil 22: Kamera hareketi modelinde kullanılan dönme, öteleme ve odaklanma parametreleri [18]. ..	41
Şekil 23: Sahne derinliği (solda) ve salt öteleme kamera hareketinden kaynaklanan optik akış (sağda) [13].....	42
Şekil 24: Salt yatay dönme (pan, solda) ve odaklanma (sağda) öteleme kamera hareketlerinden kaynaklanan optik akış (sağda) [13].....	42

Şekil 25: a), b), c) Salt sağa yatay dönme (pan right) içeren ardışık video kareleri (sırası ile 10., 11., ve 12. kareler).	44
Şekil 26: a), b), c) Sola yatay dönme ve pozitif zum karma hareketini içeren ardışık video kareleri (sırası ile 54., 55., ve 56. kareler).	44
Şekil 27: Kamera hareket tespiti ve analizi için geliştirilen yazılımın kullanıcı ara yüzü.	45
Şekil 28: K-Ortalamalar ve Ortalama Kayma algoritmaları aşırı bölütleme karşılaştırmalı sonuçları, a) Orijinal görüntü, b) K-Ortalamalar bölütleme sonucu, yanlış bölütleme sonucu sol kısımda beyaz tabak, arka plan ve el, aynı bölüt içerisinde çıkmıştır, c) Ortalama Kayma bölütleme sonucu.	46
Şekil 29: a), b) Sabit arka plana sahip ardışık video kareleri (sırası ile kare 32 ve 33), c) Aşırı bölütleme sonucu ve KLT algoritması ile hesaplanmış kırmızı hareket yöneyleri, d) Hareket yöneyleri KLT ile hesaplanmış bölütlerin maskesi, beyaz alanlara dahil olan bölütlerde hareket yöneyleri bölüt eşleme ile değil KLT algoritması ile hesaplanmıştır, e) Bölüt hareket yöneyleri (hareket yöneyleri büyüklükleri ile orantılı olarak 0-255 arasında gösterilmiştir), f) RSST ile bölüt birleştirilmesi sonucu.	48
Şekil 30: a), b) Sabit arka plana sahip ardışık trafik video kareleri (sırası ile kare 29 ve 30), c) Aşırı bölütleme sonucu ve KLT algoritması ile hesaplanmış kırmızı hareket yöneyleri, d) Hareket yöneyleri KLT ile hesaplanmış bölütlerin maskesi, beyaz alanlara dahil olan bölütlerde hareket yöneyleri bölüt eşleme ile değil KLT algoritması ile hesaplanmıştır, e) Bölüt hareket yöneyleri (hareket yöneyleri büyüklükleri ile orantılı olarak 0-255 arasında gösterilmiştir), f) RSST ile bölüt birleştirmesi sonucu... ..	49
Şekil 31: LoG Fonksiyonu.....	52
Şekil 32: DoG Uç Noktalarının arandığı komşuluk.....	53
Şekil 33: Oto-korelasyon fonksiyonuna göre farklı sonuç veren değişik tipteki imge noktaları.....	54
Şekil 34: SIFT Tanımlayıcısı.....	55
Şekil 35: Çizgede örnek bir kesik.	59
Şekil 36: (a) Mozaik görüntüsü ve (b) gerçek bölüt görüntüsü, (c) DKGB, (d) önerilen yöntem, (e) Çok ölçekli düzgülü kesik algoritması.	60
Şekil 37: Geliştirilmiş çizge yapısıç	61
Şekil 38: (a) Nesnel görüntüsü ve (b) gerçek bölüt görüntüsü, (c) DKGB, (d) önerilen yöntem, (e) Çok ölçekli düzgülü kesik algoritması.....	62
Şekil 39: (a) İnek görüntüsü ve (b) gerçek bölüt görüntüsü, (c) DKGB, (d) önerilen yöntem, (e) Çok ölçekli düzgülü kesik algoritması.	62
Şekil 40: Bölütleme yöntemlere bağlı olarak elde edilen MSE değerleri.....	62
Şekil 41: Mozaik görüntüsünün bölütlenme sonuçları (a) sınırlandırılmamış bağ ve (b) sınırlandırılmış bağ durumu için.	63
Şekil 42: Wang ve Adelson [5] hareket bölütleme algoritmasının Mobile Calendar videosu için sonuçları. (a)-(b) Orjinal dizinin 3. ve 5. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Orjinal referans çerçeveye ((b)'de gösterilen 5. çerçeve) ait kestirilen hareket katmanları.	65
Şekil 43: Önerilen algoritmanın akış çizeneği [8].	66
Şekil 44: Katman bükme işlemi. (a) Flower Garden dizisinin 324. Çerçevesi. (b) 23. Döngüde kestirilen katmanlar. Katman hareketinin ifadesi için affine model kullanılmıştır. (c) Zamanda geri yönde (4	

çerçeve) bükülen katmanlar. Siyah ve karma renkteki alanlar 320. çerçevedeki sırası ile açılmış (uncovered) ve kapanmış (covered) pikselleri ifade etmektedir. (d) Bükülmüş katmanların, kapanmış alanlardaki görünür katman belirlendikten sonraki hali.	68
Şekil 45: Hareket bölütleme algoritmasının Flower Garden çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 307. ve 309. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü (burada 309, çerçeve).için kestirilen hareket katmanları.....	70
Şekil 46: Hareket bölütleme algoritmasının Flower Garden çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 319. ve 321. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü için kestirilen hareket katmanları.	71
Şekil 47: Hareket bölütleme algoritmasının Mobile Calendar çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 3. ve 5. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü için kestirilen hareket katmanları.....	72
Şekil 48: Hareket bölütleme algoritmasının Mobile Calendar çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 11. ve 13. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü için kestirilen hareket katmanları.	73
Şekil 49 – Aday Çekim Sınırı.....	78
Şekil 50 – Aday Çekim Sınırı İçin Çizge Gösterimi	78
Şekil 51: Çalışmada kullanılan helikopter görüntüleri. Her sınıf için bir örnek verilmiştir.....	82
Şekil 52: Çalışmada kullanılan uçak görüntüleri. Her sınıf için bir örnek verilmiştir.....	84
Şekil 54 : Örüntü Tanıma Genel Şeması.....	93
Şekil 55: Önerilen Çözüm Yöntemi	93
Şekil 56 : Performans analizi için tanımlanan bölgeler : (Mavi çizgi) Yazının gerçek sınırları, (Sarı çizgi) Yazının tespit edilen sınırları, (Kırmızı bölge) Doğru tespit edilen alan, (Mavi bölge) Tespit edilemeyen alan, (Gri bölge) Yanlış tespit edilen alan.....	94
Şekil 57: Ayrık Kosinüs Dönüşümü Öznitelik Maskesi	96
Şekil 58: Örnek İmge ve 20 katsayıdan elde edilen ters dönüşüm.....	96
Şekil 59: HTD Bantları	98
Şekil 59: Ayrıt Haritaları.....	99
Şekil 61: İki sınıf bulunan bir problemde x vektörünün sınıfı bu durumda siyah olarak karar verilir [29].	100
Şekil 62:Voroni bölgeleri [29].....	102
Şekil 63: Destek vektörleri ve optimal ayırıcı hiper düzlemeler [29].....	103
Şekil 64: Sınıflandırma Sonucu	105
Şekil 65: Sobel Ayrıt Haritası.....	105
Şekil 66: Dikey İzdüşüm Profili	105
Şekil 67: Dikey izdüşüm haritası kullanılarak elde edilen sonuç.....	105
Şekil 68 : Tipik tanıma sonuçları	108
Şekil 69: NIST Yazı Tanıma Sisteminin Blok Çizeneği	111
Şekil 70: Örnek Bölütleme Sonucu.....	111
Şekil 71: Örnek Otomatik Yazı Alanlarını Bulma Sonucu	112

Şekil 72: Eşikleme ve NIST OCR Sistemi	113
Şekil 73: Örnek Otomatik Yazı Bulma Sonucu.....	114
Şekil 74: k-ortalamlar sonucunda ortaya çıkan üç küme, üç farklı renk – beyaz, siyah ve mavi – ile işaretlenmiştir.	114
Şekil 75: Üç kümeden siyah ve beyaz renkler ile elde edilen bir sonuç.....	114
Şekil 76: İstatistiksel Dil Modeli ve Viterbi Algoritması	115
Şekil 77: Örnek Sonuç.....	117
Şekil 77: Kuş bakışı görünüş yaratılması	122
Şekil 78: 6 düğümlü ve 7 bağlantılı bir çizge.....	122
Şekil 79: En küçük kapsayan ağaç örneği.....	123
Şekil 80: Kamera-1 ve Kamera-2 görüntüleri	124
Şekil 81: Hareketli cisimlerin bölütleme sonuçları	124
Şekil 82: Kuş bakışı görünüş.....	125
Şekil 83: En küçük kapsayan ağaç.....	125
Şekil 84: Eşik değeri üstündeki bağlantıları kesilmiş en küçük kapsayan ağaç.....	126
Şekil 85: Gruplanmış noktalar	126
Şekil 86: TV kanallarından örnek yarı-saydam logolar.....	128
Şekil 87: Tespit algoritmasına genel bakış.....	129
Şekil 88(a) Videodan elde edilen ayrıtlar ve (b) histogram denkleştirme sonucu ortaya çıkan çevritler. (c) ve (d) eşikleme sonrası önerilen çevrit konaçlarını göstermekte. Beyaz pikseller iç çevriti, siyahlar ise dış çevriti temsil etmekte.	130
Şekil 89: CNN Türk logosu üzerinde (a) tüm evrilme ve (b) kısmi evrilme.....	131
Şekil 90: Biriktirme başarımları sonuçları.....	132
Şekil 91: Pencereleme başarımları.....	132
Şekil 92 Öznitelik vektörlerinin çıkarıldığı, örnek ikinci seviye video elemanları. Bu seviyede Toplam 13 eleman bulunmaktadır.	135
Şekil 93: Referans videolardan veritabanı oluşturulması.....	136
Şekil 94: Ses modülü sorgu süreci.....	137
Şekil 95: COST292.v.metuq1 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile önerilen sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.....	137
Şekil 96: COST292.v.metuq1 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.....	138
Şekil 97: COST292.m.A0metuq1 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.....	138
Şekil 98: COST292.m.A0metuq2 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.....	139

Şekil 99: COST292.m.A0metuq3 ait Ortalama F1 ve Dönüşümler grafiđi. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir..... 139

ÖZET

COST 292 projesi kapsamında gerçekleşen TÜBİTAK projemizin temel amacı çoklu ortam verilerinden otomatik olarak anlamsal bilgi çıkarmaya yöneliktir. Proje kapsamında yapılan çalışmalar, öznitelik çıkarma, bölütleme ve uygulamalar olarak 3 ana başlık altında incelenebilir. Öznitelik çıkarma konusunda 2-B ve 3-B şekil tanımlayıcılar kullanılarak 2-B maskelerin ve 3-B nokta kümelerinin sınıflandırılması incelenmiştir. Her iki boyuttaki öznitelik çıkarma problemi için ilgili literatürden önde gelen yöntemler incelenmiş ve deneyler yardımıyla başarıları incelenmiştir. Kamera hareketlerinin tespiti ve tanımlanması da aynı başlık altında zamansal bir öznitelik olarak analiz edilmiş ve kamera hareketleri başarıyla elde edilmiştir. Özellikle nesne tanımda yaygınlaşan ilgi noktaları çıkarma konusunda öne çıkan yaklaşımlar da bu proje kapsamında incelenmiştir. Bölütleme başlığı altında ise çizge tabanlı özgün bir imge bölütleme algoritması önerilmiştir ve literatürdeki benzerlerine göre başarıyı ortaya konmuştur. Hareketli nesne tespiti problemine yönelik olarak, bölgeler için tanımlı parametrik hareket modeli temelli, ön ve arka planda oluşan açılma ve kapanmaları da dikkate alan yeni bir yaklaşım deneyler ile sınanmıştır. Video kamera çekim sınırlarını bölütlemek için ise baskın kümeler kullanan özgün bir çalışma ortaya konmuştur ve umut verici sonuçlar elde edilmiştir. Uygulamalar başlığı altında öncelikle 2-B şekil bilgisi kullanarak hava taşıtlarını sınıflandırma amacıyla farklı öznitelik ve sınıflandırma deneyler ile sınanmıştır. Görüntüler içine yerleştirilmiş yazıların tespit edilmesi ve sonra tanınması için bir sistem geliştirilmiş ve kapsamlı deneyler ile başarıyı ortaya konmuştur. Diğer bir uygulama olarak, kamera tabanlı güvenlik sistemleri için çoklu kameralar kullanıldığında örtüşmelerin engellenebildiğini gösteren özgün bir yaklaşım önerilmiştir. Son olarak, video parçalarının kopyalarının veri tabanlarından gürbüz bir şekilde bulunabilmesi için özgün bir çokkipli algoritma önerilmiş, TRECVID 2008 yarışmasına COST 292 grubunu temsilen katılıp, başarılı olunmuştur.

ABSTRACT

The purpose of COST 292 project is conducting research on multi-modal analysis on multimedia data for obtaining semantic information. The research efforts on this project can be presented in three main paths, namely, feature extraction, segmentation and applications. Under feature extraction research, 2D and 3D shape descriptors are analyzed from state-of-the-art literature. Camera motion detection and analysis is achieved as a temporal feature for video. Interest point detection problem is also examined as a result of its recent popularity in object recognition research. Under segmentation topic, a novel graph-theoretic image segmentation method is proposed which has advantages over some leading algorithms from the literature. On the other hand, a region-based parametric modeled object segmentation algorithm is proposed that considers occlusions between foreground and background for obtaining a better segmentation result. Graph-theoretic dominant set is applied to video shot boundary detection problem for the segmentation of video into temporal portions. As applications to semantic information retrieval, shape-based aerial vehicle classification is achieved for different descriptors and classifiers. Overlay videotext detection and recognition problem is examined in detail and a remarkable performance is obtained for this problem. For visual surveillance, exploitation of multi-camera system is proposed for occlusion handling problem. As a different application, semi-transparent logo detection is achieved with a promising performance for broadcast video content. Finally, a content-copy detection problem is also examined and a novel multi-modal method is proposed that is found out to be successful under TRECVID 2008 contest.

1. GİRİŞ

Görsel duyunun insanoğluna ait en güçlü ve en karmaşık duyu olması, görsel verilerden otomatik anlamsal bilgiler çıkarma problemini her yönden çekici hale getirmektedir. Görsel veriden çıkarılan temel alt seviye bilgiler ile insanoğluna ait çeşitli teknik problemleri çözmek için gerekli anlamsal üst seviye bilgiler arasında geniş bir uzaklığın bulunması (literatürde “anlamsal uçurum” (*semantic gap*) olarak adlandırılmaktadır), bu boşluğu doldurmak için yoğun araştırmaların yapılmasına neden olmaktadır. Veriden anlamsal bilgi çıkarmayı hedefleyen bu çok geniş problem uzayı içinde bulunan çok farklı araştırma alanlarının bir kısmı bu proje kapsamında irdelenmiş ve deneyler yardımıyla sınanmıştır.

Anlamsal uçurumun kapatılması için gerekli en önemli yaklaşımın, resmin bütününden çok yerel unsurları dikkate almak olduğu her geçen gün daha belirgin olarak ortaya çıkmaktadır. Bu amaçla içeriği anlam yüklü bütünlere bölmeyi hedefleyen bölütleme problemi daha önem kazanmaktadır. Proje kapsamında yapılan çalışmaların önemli bir kısmı bölütleme problemine özgün çözümler bulmaya yönelik yapılmıştır. Bölütlenmiş nesnelere sınıflandırılabilmesi için şekil tanımlayıcıların oldukça önemli olduğu aşikardır. Video verilerinden elde edilen 2-B imgelerin bölütlenmesi sonucunda ortaya çıkan bölgeler için 2-B şekil tanımlayıcıları gerekmektedir. Yakın gelecekte yaygınlaşması beklenen video ve videoda yer alan sahneye ait noktaların kameraya uzaklıklarını veren derinlik haritaları için aynı problem 3-B bir tanımlamayı gerekli hale getirmesi beklenmektedir. Yerel özelliklere erişmenin diğer bir yolu da, bölütleme yapmadan, resimlerde nesnelere üzerinde yer alan çeşitli ilgi noktalarının bulunması ve obje tanıma amaçlı olarak örnek nesnelere benzer noktalarla karşılaştırılmasıdır; projede bu konuyla ilgili olarak çeşitli incelemeler yer almıştır.

Proje kapsamında anlamsal bilgiye ulaşma amacıyla çeşitli örnek problemler üzerinde çalışılmıştır. 2-B şekil tanımlayıcıları ile ilgili en önemli uygulama alanlarından birisi taşıtların tanınmasına yöneliktir ve otomatik olarak taşıtların silüet bilgisi üzerinden tanınmasına yöneliktir. Görsel verideki üst seviye anlamsal bilgiye ulaşmanın diğer bir yolu da, veri üzerindeki yazıların bulunup, tanınmasına dayanmaktadır. Yazıyla, üzerinde bulunduğu içerik arasında bir ilişki bulunduğu varsayımı üzerine kurulu bu yaklaşım sonucu, üst seviye bilgilere ulaşmak mümkün olabilir. Son dönemde çok yaygınlaşan görsel veriden güvenlik ve istihbarat amaçlı anlamlar çıkarma problemi için bu proje kapsamında birden çok sayıda kamera kullanımının avantajları incelenmiştir. Özellikle, sahnedeki hareketli nesnelere örtüşmelerini engelleyecek bir yaklaşımla, gerek anormal durumların tespiti, gerekse diğer her çeşit anlamsal analiz için nesne yörüngelerinin daha hatasız çıkarılması hedeflenmiştir. Başka bir uygulama olan (yarı saydam) logo tespiti ile de TV yayınlarının analizinin otomatik olarak yapılabilmesi hedeflenmiştir. Son bir uygulama olarak, belirli bir görsel dizinin bir veritabanı içinde arama problemi ele alınmıştır. Son yıllarda oldukça önem kazanan bu problemde, çeşitli değişikliklere ve işlemlere maruz kalmış bir çoğul ortam verisinin, gürbüz öznitelikler yardımıyla, benzerlerinin veritabanlarından bulunabilmesi hedeflenmektedir.

Proje sonuç raporunda, öncelikle kullanılan görsel öznitelikler açıklandıktan sonra, bölütleme problemine yönelik önerilen farklı algoritmalar açıklanacak, son olarak bu temel unsurları kullanan uygulamalar ve sonuçları sunulacaktır. Konuların farklılıklar dikkate alındığında, ilgili başlıklara ait referanslar, her bir bölümün sonunda yer almaktadır.

2. GÖRSEL ÖZİNİTELİK ÇIKARMA

Bu proje kapsamında görsel öznelik çıkarma başlığı altında 4 ana problem incelenmiştir:

- 2-B Şekil Tanımlayıcılar
- 3-B Şekil Tanımlayıcılar
- Zamansal Öznelik olarak Kamera Hareket Analizi
- İlgili Noktası Bulma

Bundan sonraki bölümlerde sırasıyla bu dört tanımlayıcı incelenmiştir.

2.1. 2-B ŞEKİL TANIMLAYICILAR

2-B şekil tanımlayıcılar bölütlenmiş nesnelere ait maskeleri girdi olarak kullanıp, bunlara ait şekil bilgisini tanımlamaya çalışırlar. Kullanılan nesne tanımlayıcılarını iki gruba ayırmak mümkündür: *Bölge tabanlı yöntemler* ve *çeper (kontur) tabanlı yöntemler*. Bölge tabanlı yöntemlerde nesnelere ait silüetlerinin kapladığı bölge esas alınır. Kontur tabanlı yöntemlerde ise, nesne silüetlerinin sınırlarının şekli esas alınır.

• Bölge Tabanlı Yöntemler

Bu çalışmada, bölge tabanlı iki farklı yöntem uygulanmıştır.

2.1.1.1. Açısız Radyal Dönüşüm (ARD) Betimleyicisi [4]

ARD, MPEG-7 standardında kullanılan iki temel şekil betimleyicisinden birisidir. Bu betimleyici; görüntülerdeki dönme, öteleme ve ölçeklemeye duyarlıdır. ARD, birim çember içinde tanımlı, Zernike Momentlerine [1] benzer, ancak hesaplaması daha hızlı bir şekilde yapılabilen bir dönüşümdür. ARD katsayıları şu şekilde elde edilir:

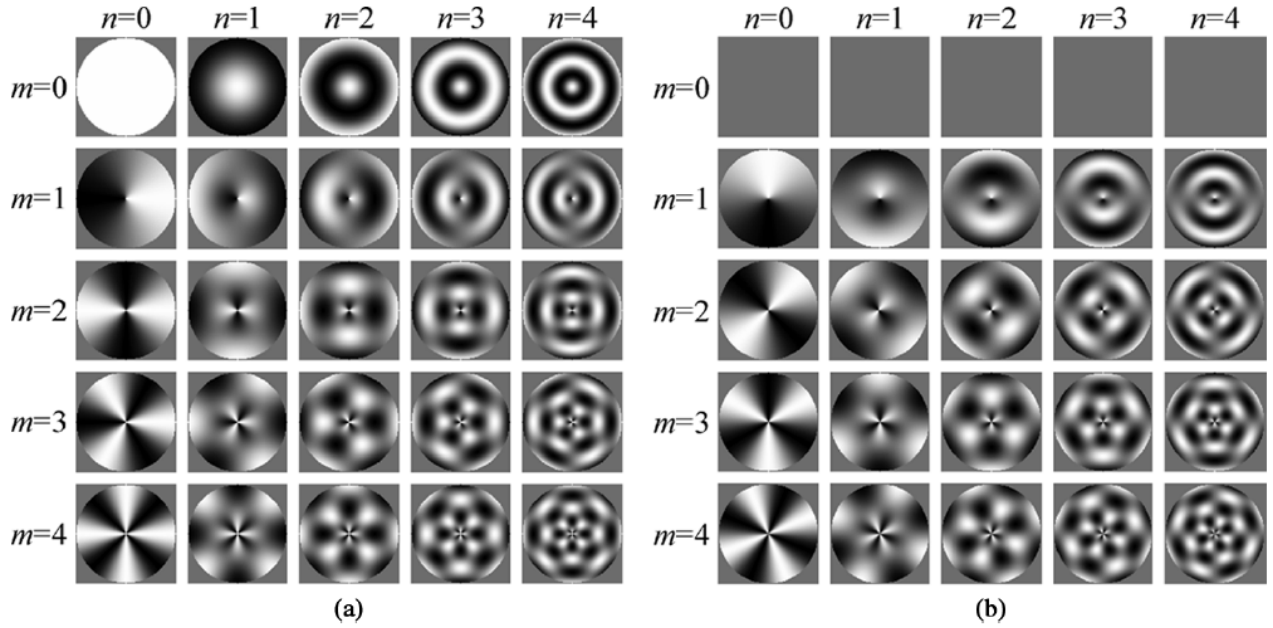
Her bir nesne silüeti için m ve n indisli bir katsayılar kümesi şu şekilde oluşturulur.

$$F_{nm} = \langle V_{nm}(\rho, \theta), f(\rho, \theta) \rangle$$

Burada, $f(\rho, \theta)$ polar koordinatlarda ifade edilmiş hali ile görüntü parlaklık fonksiyonu, $V_{nm}(\rho, \theta)$ ise n ve m derecelerinde, ARD katsayılarının hesaplanmasında kullanılan dönüşüm/tabana fonksiyonlarıdır. Bu fonksiyonlar şu şekilde elde edilir [2].

$$V_{nm}(\rho, \theta) = \frac{1}{2\pi} e^{jm\theta} R_n(\rho)$$
$$R_n(\rho) = \begin{cases} 1, & n = 0 \\ 2 \cos(\pi n \rho), & n \neq 0 \end{cases}$$

Yukarıda anlatılan taban fonksiyonlarının gerçel ve sanal kısımları şekil 1'de gösterilmiştir



Şekil 1: Değişik m ve n değerleri için taban fonksiyonlarının gerçel (a) ve sanal (b) kısımları. Gösterim amacıyla, aradaki gri değerler, beyaz 1, siyah 0 olacak şekilde oluşturulmuştur.

ARD betimleyicisi şu şekilde oluşturulur [3,4] :

i) Bölgesel önışleme: Nesne silüetinin oluşturulması ve aradeğerleme (standard bir büyüklüğe getirme) aşamasıdır. Aradeğerleme, ağırlık merkezi ile ağırlık merkezine en uzak nokta arasındaki uzaklık sabit bir değer olacak şekilde yapılır. Bu sayede, betimleyicinin, ölçekten bağımsızlığı sağlanır.

ii) ARD: Normalize edilmiş görüntü ve ARD taban fonksiyonları kullanılarak ARD katsayıları (yukarıda belirtildiği şekilde) elde edilir. Taban fonksiyonları, teoride polar koordinat sisteminde tanımlanmış olmakla birlikte, pratikte, hesaplama kolaylığı sağladığı için, kartezyen koordinat sisteminde hesaplanır. İzdüşüm alınırken, nesnenin ağırlık merkezi taban fonksiyonlarının orijinine yerleştirilir. Bu sayede, ötelemeye karşı duyarsızlık sağlanmış olur.

iii) Normalizasyon: ARD katsayılarının büyüklükleri $|F_{00}|$ (nesnenin piksel sayısı) ile bölünerek normalize edilir:

$$ArtM[n, m] = \frac{|F_{nm}|}{|F_{00}|}$$

Açıktır ki, büyüklük alınması sebebiyle, $ArtM[n, m]$ dönmelere karşı duyarsızdır (θ 'dan bağımsızdır).

MPEG-7 standardında, 3 seviye radyal ve 12-seviye açısai katsayı ($n < 3$, $m < 12$) kullanılır. $m=0$, $n=0$ katsayısı, normalizasyon sonrasında mutlaka 1 olduğu için, kullanılmaz. Böylece, toplam 35 katsayı kullanılmış olur..

iv) Nicemleme: MPEG-7 standardı, ARD katsayıları için 4-bit nicemleme öngörmektedir. Nicemleme tablosu, bir üssel dağılım modeli esas alınarak oluşturulmuştur.

İki farklı nesnenin ARD katsayılarının benzerliğini ölçmek için, L-1 normu kullanılır.

2.1.1.2 Hu Momentleri [5]:

Ağırlık merkezinin yerinden bağımsız olarak tanımlanan nesne momentleri şu şekilde tanımlanır:

$$\mu_{pq} = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} (x - \hat{x})^p (y - \hat{y})^q f(x, y)$$

Burada, \hat{x} ve \hat{y} nesnenin ağırlık merkezinin koordinatlarını, $f(x,y)$ de görüntünün (x,y) koordinatındaki değerini ifade etmektedir. Açıkta ki, bu momentler nesnelere ötelenmesinden bağımsızdır. Bu momentleri ölçekten de bağımsız hale getirebilmek için aşağıdaki normalizasyon yapılır:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{[(p+q)/2]+1}}$$

Normalize edilmiş momentler kullanılarak, 7 adet Hu katsayısı şu şekilde elde edilir [5]:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{20} + \eta_{12})[(\eta_{03} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ & \quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - \\ & \quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned}$$

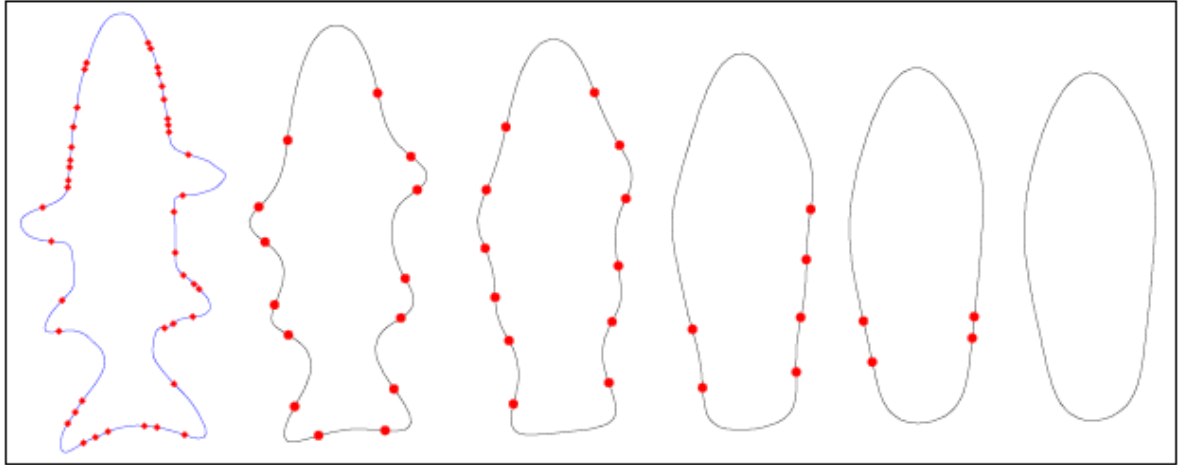
İlk iki katsayı ölçek ve ötelemeden bağımsızlığı, sonraki dört tanesi yansıma ve dönme duyarlılığı sağlar. Yedinci katsayı, işareti ile yansıma ayrıştırılmasını sağlar.

- **Bölge Çeperi Tabanlı Yöntemler**

Bu çalışmada, bölge tabanlı iki farklı yöntem uygulanmıştır.

2.1.2.1 Eğrilik Ölçek Uzayı (EÖU) Betimleyicisi [7]:

EÖU, ikilik görüntülerde, kesişmeyen çeperlerin bükülme noktaları üzerine odaklanmış bir betimleyicidir. Bir çeperin (eğrinin) bükülme noktaları olması için, eğrinin kapsadığı alanın içbükey olması gereklidir. Bir eğrinin bükülme noktaları, eğrilik fonksiyonunun sıfır geçimleri ile tanımlanır. Şekil 2'de, çeşitli eğrilerden oluşan bir örnek verilmektedir. Bu örnekte, bükülme noktaları noktalarla gösterilmiştir. En sağdaki görüntü, convex olması nedeniyle, bükülme noktasına sahip değildir.



Şekil 2 : Çeşitli eğriler için bükülme noktaları [6]. Soldan sağa, aynı eğrinin, Gauss filtrelerinden geçirilmiş halleri verilmiştir. İlgilili filtre standart sapmaları: $\sigma = 1,4,7,10,12,14$.

EÖU betimleyicisi şu şekilde elde edilir:

- İkili görüntünün çeperi morfolojik operasyonlarla bulunur.
- Elde edilen çeperden, rastgele bir noktadan başlanarak ve saat yönünün tersine ilerlenerek N eşit uzaklıklı nokta alınır.
- Alınan noktaların x ve y koordinatları kullanılarak iki vektör, X ve Y elde edilir.
- Ölçek uzayı, X ve Y vektörleri aşamalı olarak, $[0.25 \ 0.5 \ 0.25]$ katsayılı, basit bir alçak geçiren süzgeçten geçirilir. Filtrenin her uygulanmasından sonra, yeni bir ölçek uzayı katmanı elde edilmiş olur. Her yeni katmanda, ilgili görüntü daha fazla az içbükey olur.
- Filtrenin her uygulamasından sonra, yeni çeperin eğrilik fonksiyonu hesaplanır [7]:

$$F(x, y) = \frac{x' y'' - y' x''}{(x'^2 + y'^2)^{3/2}}$$

Burada, (') ve (''), sırasıyla, değişkenlerin parametrik gösterimlerine göre birinci ve ikinci türevlerini ifade etmektedir. Parametrik gösterimdeki parametre, seçilen noktaların sırasından ibarettir. Daha açık bir ifadeyle, eğer örnekleme noktalarının indisleri $i = 1 \dots N$ ise,

$$x' = \frac{x(i+1) - x(i-1)}{2}$$

$$y' = \frac{y(i+1) - y(i-1)}{2}$$

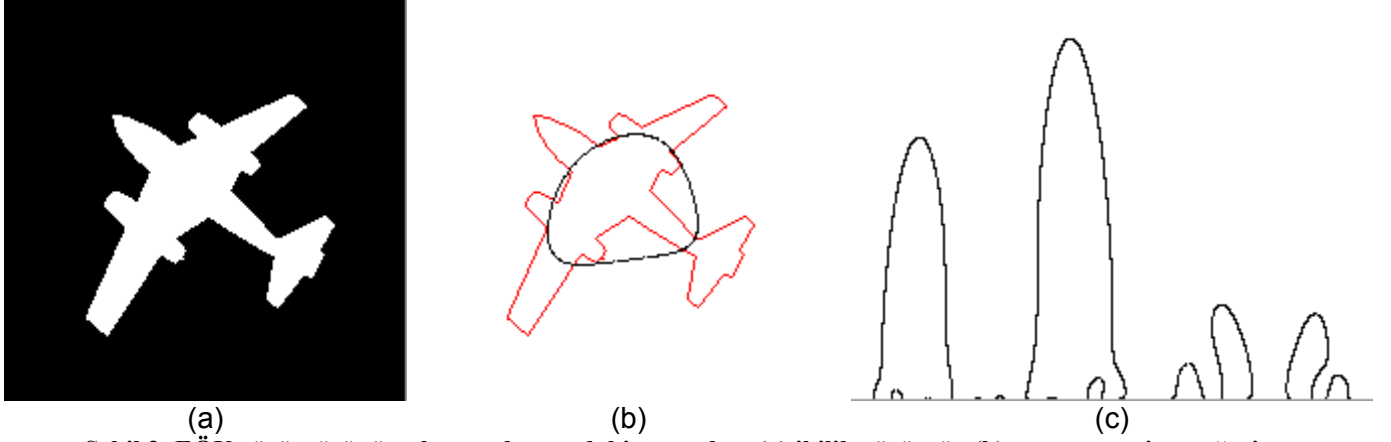
$$x'' = x(i+1) - 2x(i-1) + x(i-1)$$

$$y'' = y(i+1) - 2y(i-1) + y(i-1)$$

Eğer parametre, $[0,1]$ aralığına ölçeklenecek şekilde (i/N şeklinde) normalize edilirse, eğrilik fonksiyonunun paydası 1 olur ve hesaplama kolaylaşır. Detaylar ve çıkarım için bkz [7].

- Her katmanda, eğrilik fonksiyonunun sıfır geçiş pikselleri, o katmanın "katman satırı"nda işaretlenir. Bu işlem, eğrilik fonksiyonunda sıfır geçişi olmayan bir katman bulununcaya kadar sürdürülür.
- Katman satırları kullanılarak, bir EÖU görüntüsü elde edilir. Bu görüntüde, en az bir boş satır bulunur. Başka bir deyişle, (f)'te de belirtildiği gibi, filtreleme işlemi çeper

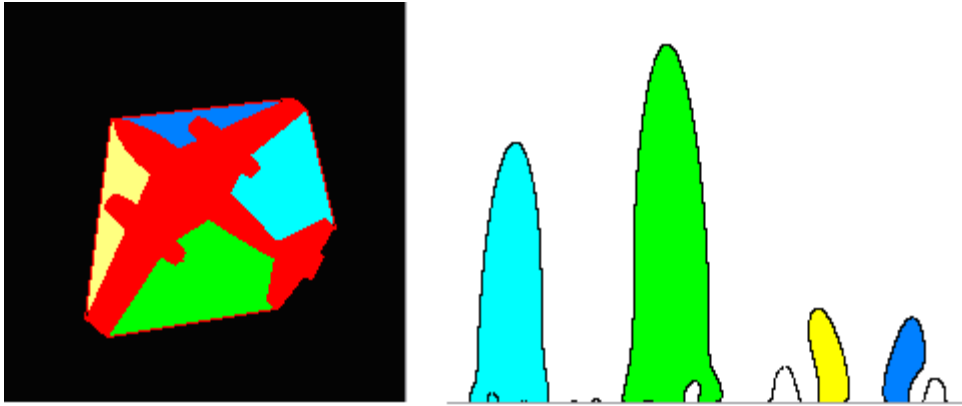
dışbükey hale gelinceye kadar sürdürülür. Şekil 3'te bir ikilik görüntüden EÖÜ görüntüsü elde etmenin önemli aşamalarının sonuçları gösterilmiştir.



Şekil 2: EÖÜ görüntüsünün oluşturulmasındaki aşamalar: (a) ikilik görüntü, (b) nesne çeperi ve çeperin dışbükey hale getirilmiş hali, (c) oluşturulan EÖÜ görüntüsü.

- h. EÖÜ görüntüsünden, en yüksek tepeye göre belirlenen bir eşik değeri kullanılarak, en yüksek tepeler seçilir ve bu tepeler ait EÖÜ koordinatları alınır. Bu koordinatlar, ilgili tepelerin yüksekliğine (y değerlerine) göre sıralanır. x değerleri en yüksek tepeye göre uzaklık şeklinde değiştirildikten ve nicemlendikten sonra aynı sırayla kaydedilir.
- i. Nesnenin dışmerkezlik ve yuvarlaklık değerleri de ayrıca betimleyiciye eklenir.

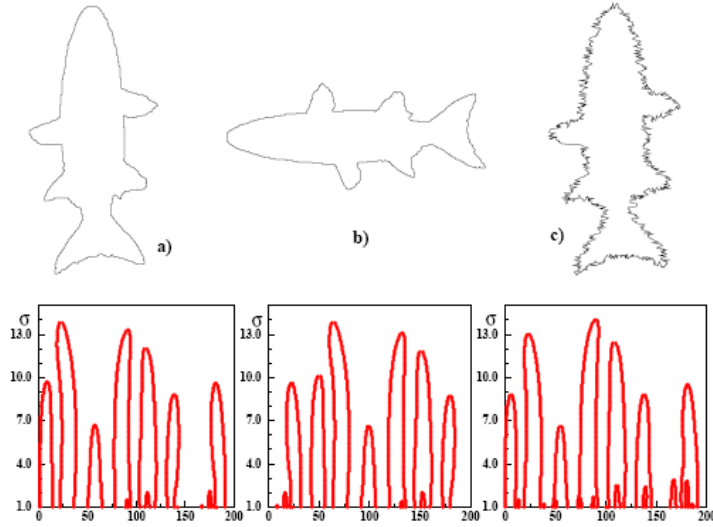
Aşağıdaki örnek (Şekil 4), EÖÜ tepelerinin neye karşılık geldiğinin anlaşılması için verilmiştir. Bu şekilde, nesneyi kapsayan en küçük dışbükey çokgen (yaklaşık olarak) kullanılarak, nesnenin silüetindeki büyük boşluklar ve bunlara karşılık gelen EÖÜ tepeleri farklı renklerde gösterilmiştir.



Şekil 3 : Nesnenin en büyük içbükey alanları (a) ve ilgili EÖÜ tepeleri (b).

EÖÜ betimleyicisi, kolaylıkla ölçekleme ve dönmeye karşı dayanıklı hale getirilebilir. Şekil 5'te görüldüğü gibi, nesnenin döndürülmesi, EÖÜ görüntüsünde basit bir dairesel kaymaya neden olmaktadır. EÖÜ betimleyicisinde x koordinatlarını en yüksek tepeye göre hesaplamak bu nedenle dönmelere karşı dayanıklılık sağlar ve gereklidir. Ölçeklemeye dayanıklılık da her nesne için eşit sayıda çeper noktası alınarak sağlanır.

Gürültüye karşı dayanıklılık da EÖÜ görüntüsünde tepe yüksekliklerini eşikleme ile sağlanır (şekil 5-c).



Şekil 4 : Orijinal eğri (a), döndürülmüş versiyonu (b), ve gürültülü versiyonu (c). İlgili EÖU görüntüleri eğrilerin altlarında verilmiştir.

Ayna görüntülerini de başarıyla tanıyabilmek için, iki görüntünün karşılaştırılması sırasında, EÖU görüntüleri hem düz hem ters sıralama kullanılarak karşılaştırılır bu ikisi arasında en iyi sonuç seçilir.

EÖU görüntüleri dışbükey eğrilerde kullanılamaz olduğundan, betimleyiciye dışmerkezlik ve yuvarlaklık bilgileri de eklenir.

2.1.2.2 Fourier Betimleyicisi

Fourier betimleyicisi, nesnelerin çeperlerinin ağırlık merkezine uzaklığını çeper piksellerinin bir fonksiyonu olarak kullanır ve bu fonksiyonun Fourier dönüşümünü alır. Bu dönüşümün katsayılarını dönmelere karşı dayanıklı hale getirir ve kaydeder. Uygulaması şu şekildedir:

- Nesnenin çeperi ve çeper piksellerinin ağırlık merkezine uzaklık fonksiyonları bulunur. Bu fonksiyon birbirine eşit uzaklıkta N çeper noktasında örneklenir.
- Örneklemeyle elde edilen fonksiyonun Fourier dönüşümü alınır.
- Görüntü gerçel olduğundan katsayıların yarısı yeterlidir. Bu yüzden ilk $N/2$ kullanılır.
- Dönmelere karşı dayanıklılık sağlamak için, katsayıların faz kısımları atılır (büyüklükleri alınır).
- Uzaklık fonksiyonunda ölçekten kaynaklanan farklılıkları engellemek için elde edilen katsayılar sıfırcı katsayıya bölünür.

Fourier betimleyicisi, uygulaması kolay olmakla birlikte, çeperdeki değişimlerin birbirlerine göre konumları hakkında yeterli bilgi sağlayamadığından, genellikle temel değişimlerin belirlenmesinde kullanılmaktadır.

İlgili betimleyicilerle ilgili deneysel sonuçlar ve bu sonuçlara bağlı yapılan çıkarımlar Bölüm 4.1'de sunulmuştur.

- **Referanslar**

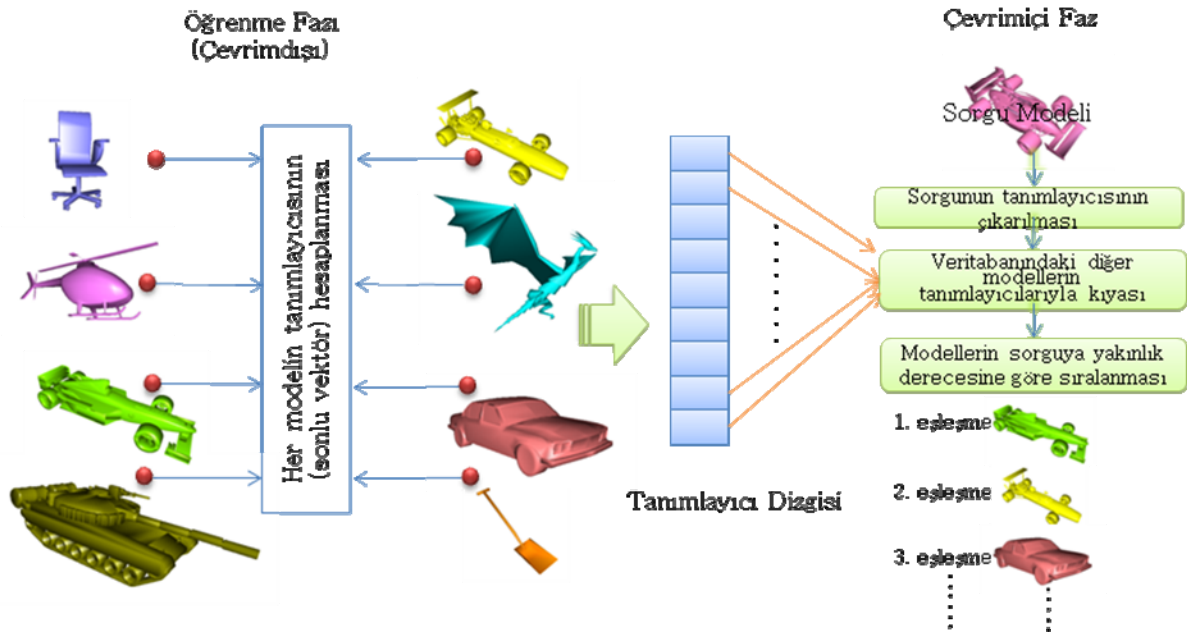
- [1] A.Khotanzad, Y.H.Hong, "Invariant Image Recognition by Zernike Moments", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 5, May 1990
- [2] M.Bober, "MPEG-7 Visual Shape Descriptors, IEEE Transactions", *On Circuits And Systems For Video Technology*, Vol. 11, No. 6, June 2001
- [3] M. Akcay, A. Baskurt, B. Sankur, "Measuring Similarity Between Color Image Regions", *EUSIPCO'2002: European Conf. On Signal Processing*, Toulouse, September 2002
- [4] M.Bober, F.Preteux, W.Y.Kim, "Shape Descriptors", *Introduction to MPEG-7 Multimedia Content Descripton Interface*, Edited by B.S.Manjunath, P.Salembier, T.Sikora, pp.231-260, John Wiley & Sons Ltd., Sussex, England, 2002. ISBN: 0 471 48678 7
- [5] Hu, "M. Visual pattern recognition by moment invariants", *IRE Trans. Information Theory*, Vol. IT-8, Num. 2, 1962.
- [6] F.Mokhtarian, S.Abbasi, J.Kittler, "Robust and Efficient Shape Indexing Through Curvature Scale Space", *Proceedings of British Machine Vision Conference*, pp. 53-62, Edinburgh, U.K., 1996
- [7] F.Mokhtarian, A.K.Mackworth "A Theory of Multiscale Curvature-Based Shape Representation for Planar Curves", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, Vol. 14, No. 8, pp. 789-805, 1992

2.2. 3-B ŞEKİL TANIMLAYICILAR

Üç boyutlu (3B) modellerin alınması, işlenmesi, görüntülenmesi ve kaydedilmesindeki olağanüstü gelişmelerle birlikte 3B modellerin kullanım alanları da yaygınlaşmıştır. Televizyonu, sinemayı, oyunları, mimari, mühendislik ve medikal uygulamaları ve internet dünyasını buna örnek olarak verebiliriz. Haziran 2008 verilerine göre 1.463 milyar kişi internet kullanıcısıdır [1]. Bu durum görsel veri analizini oldukça revaçta bir araştırma konusu haline getirmiştir. Ancak günümüze kadar görsel verilerin alınması, iletimi ve görüntülenmesi konularında daha çok araştırma yapılmış anlamsal bilgi çıkarımı konularına çok da değinilmemiştir. 3B modellerin kullanımının ve bununla birlikte 3B veritabanlarının artışı, 3B verilerinin indekslenmesini, sınıflandırılmasını ve organizasyonunu gerekli kılmıştır.

Şekil 5'de temel bir 3B arama motorunun ana hatları verilmiştir. Sistemin çevrimdışı ve çevrimiçi fazları vardır. Çevrimdışı fazda veritabanında bulunan modellerin betimleyicisi hesaplanır ve kaydedilir, bu aşamanın tek bir sefer çalıştırılması yeterlidir. Çevrimiçi faz ise kullanıcı veritabanında bir model veya modelin benzerlerini aramak istediğinde devreye girer. Bu kısımda arama yapılacak olan modelin betimleyicisi hesaplanır ve veritabanında bulunan modellerin betimleyicileriyle kıyaslanır. Bu kıyas bir uzaklık fonksiyonundan başkası değildir.

Halihazırda bulunan sistemler genellikle bilgisayar grafikleri kullanılarak oluşturulmuş yapay ve poligon ağ modellerini temel alarak tasarlanmıştır. Bu poligon ağ modelleri 3B noktalardan (vertex) ve noktaların birbirleriyle olan bağlantı bilgilerinin verildiği yüzeylerden (face) oluşur. Bununla birlikte bu sistemler sahnede sadece tek bir modelin olduğu varsayımını benimserler. Ancak gerçek hayatta nesnelere sahnede başka nesnelere birlikte yer alırlar ve bazen de modelin yalnızca bir kısmı izleyici tarafından görünür (derinlik verileri ya da 2.5B verisi). Lazer tarayıcılarının, uçuş zamanı kameralarının verileri ve üç boyutlu televizyonda kullanılan veriler 2.5B sahnelerdir ve gerçek uygulama alanları mevcuttur.



Şekil 5: 3B arama motorunun temel bileşenleri

Derinlik verileri (2.5B) üzerinde yapılan çalışmalarda genellikle lazer tarayıcı görüntüleri kullanılmıştır. Bu veriler çok yoğun nokta kümelerine sahiptir; yüzey bilgisi elde etmeye elverişlidir ve modelleri tanımlamak için eğrilik, yüzey normali gibi yüzey bilgileri kullanılmaktadır [2][3][4][5][6]. Ancak 3B Televizyon uygulaması için kullanılan 2.5B verilerin çoklu görüntü sahnelerinden elde edildiği düşünüldüğünde verilerin lazer tarayıcı cihazların verilerine göre daha seyrek ve gürültülü olduğu gözlenecektir. Bu durumda başka yöntemlerin ve tanımlayıcıların bulunması gerekmektedir.

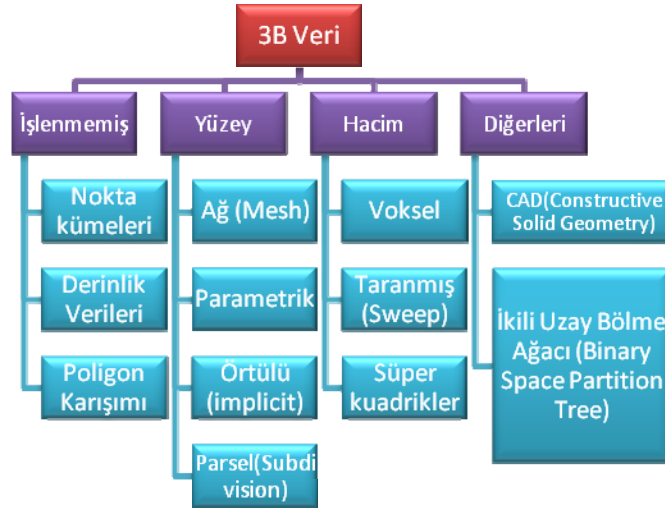
Gerçek hayata ait verilerin yapay modellerden uzak olduğunu kabul etsek dahi tamamen soyutlamak ve 2.5B verileri için işe yarayacakları söylemek yanlış olur. Hatta 3B nesne tanımlama yolunda ortak yaklaşımlarında da bulunabileceği 2B şekil tanımlamada kullanılan yöntemler de gözden geçirilmelidir. 2B şekil tanımlama ve çıkarımı hakkında detaylı bir inceleme için [7]nci referansa bakılabilir.

Gelecek bölümde 3B nesne formatları, veritabanları, çıkarım yöntemleri ve yöntemlerin performans ölçüleri üzerinde durulacaktır.

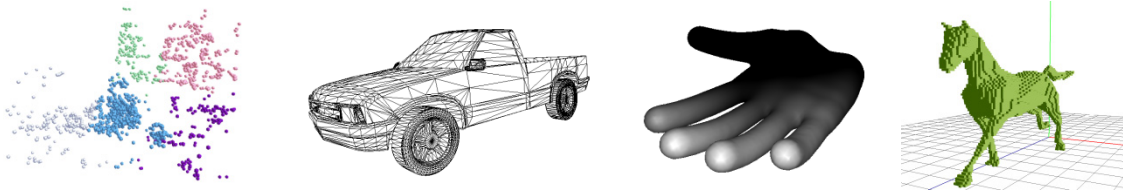
1. İçerik Çıkarımı

a. 3B Model Gösterimi

3B gösterim yöntemleri Şekil 6'de verilmiş olup en çok kullanılan formatlardan örnekler de Şekil 7'de gösterilmiştir.



Şekil 6: 3B formatların sınıflandırılması



Şekil 7: Nokta kümeleri, üçgen ağ, derinlik verisi ve voksel gösterimlerinden birer örnek

3B gösteriminde en basit yöntem nokta kümeleridir. Nokta kümelerini uzayda asılı duran noktaların konumları şeklinde düşünebiliriz. 3B bilgi çıkarımı algoritmalarının nokta kümeleri formatını desteklesi önemlidir. Çünkü diğer gösterim formatları en basit yapı olan nokta kümelerine kolayca dönüştürülebilir. Mesafe algılayıcıları ve bilgisayar görme metodları (çoklu görüntülerden 3B bilgisi çıkarımı gibi) nokta kümesi verileri elde etmekte kullanılır. Ancak derinlik verileri 2B bir görüntünün her pikselinde derinlik bilgisinin olduğu verilerdir.

Noktalar arasında zayıf da olsa bir komşuluk ilişkisi barındırmaktadır (piksel komşuluğu). Gelişigüzel poligonlardan oluşan modellere ise "Poligon Karışımı" (Polygon Soup) denir. İşlenmemiş modeller arasında yapısal olarak en çok bilgiyi içerirler.

Bilgisayar destekli dizayn (CAD) sitemlerinde Yapısal Katı Geometrisi (Constructive Solid Geometry) ve parametrik yüzeyler tercih edilir. Parametrik yüzeyler gerçekçi yüzeyler oluşturmakta kullanılır ve matematiksel formülasyona sahiptir [8]. En popüler parametric gösterim birörnek olmayan rasyonel B-Kama formatıdır (nonuniform rational B-Spline format, NURBS).

Voksel yapılar genellikle medikal uygulamalarda yer alırlar. Voksel basit olarak 2B görüntülerde piksele karşılık gelen hacimsel elemanlardır. Vokseller pozisyon bilgisi taşımaz ancak birden çok bilgiyi de saklayabilir. Medikal rezonans görüntüleme, ultrason görüntüleme ve bilgisayarlı tomografi voksel formatındaki verileri kullanır[1].

Tüm gösterimler arasında en popüler olan ise hiç kuşkusuz poligon ağ formatıdır (polygon mesh). Bu gösterimin de kendi içinde bir çok farklı formatı vardır. VRML, OFF, PLY, PLY ve 3DS bunlardan en popüler olanlarıdır. Bütün hepsinde temel fikir aynıdır; köşeler ve bunların oluşturduğu poligonlar (genellikle üçgen tercih edilir) ve poligonların birbirlerine değen kenarlarının kusursuzca oturmuş bilgileri. Diğer 3B gösterimler hakkında daha ayrıntılı bilgi için [9] ve [8]'e bakılabilir.

b. 3B Bilgi Çıkarımı üzerine Yapılmış Çalışmalar

3B nesne tanımlama üzerine değerli araştırmalar [8] [10], ve tezler [11] [12] [13] [14] bulunmaktadır. İlerleyen bölümlerde yöntemlerin detaylarına değinilecektir. Bir çok 3B bilgi çıkarımı yöntemi mevcuttur ancak bu metodların kesin bir sınıflandırılması henüz yapılmamıştır. Her araştırmacı kendine göre bir sınıflandırmayı tercih etmiştir. Kendisiyle benzer bir sınıflandırmayı tercih ettiğimiz [11]inci kaynakta beş kategori oluşturulmuştur. Histogram tabanlı, dönüşüm tabanlı, çizge tabanlı ve 2B görüntü tabanlı bir sınıflandırmaya gidilmiştir. Iyer [15] ise 3B şekil tanımlama yöntemlerini global özellikli, yapısal özellikli tanımlı, çizge tabanlı ve histogram tabanlı olarak beş sınıfa ayırmıştır.

3B nesne karşılaştırması ve eşleşmesi için benimsenen yöntem, modellerin matematiksel olarak formüle edilip, ki buna modelin betimleyicisi denilmektedir, kıyas edilen modelin betimleyicisi ile veritabanındaki diğer modellerin betimleyicilerini bazı metrikler kullanarak yakınlıklarına bakmaktır. Asıl amaç araştırmayı matematiksel bir temele oturtmak böylece pratik, hızlı, kolay işlenebilen, az yer tutan ve yeni modellerin kolaylıkla eklenebildiği indeksleme ve sınıflandırma sistemleri geliştirmektir.

Amaçlanan iş zor olmasının yanında birçok sorunu da içermektedir. Daha önce de belirtildiği gibi birçok 3B gösterimi ve bunların da birçok formatı mevcuttur. Bu formatlardan bağımsız bir bilgi çıkarımı metodu geliştirmek zaten başlı başına bir araştırma konusudur. Bu sebeple genellikle belli bir format benimsenir ve diğer formatlara ise dönüşüm işlemi uygulanır. Bir başka sorun da yön ve ölçek farklılıklarıdır. Modeller arasında farklı geometric oranlar ve pozlar karşılaştırma aşamasında ciddi problemlere yol açabilir. 3B nesnelere etkin ve verimli bir biçimde betimlemek ise bir başka sorundur. Karşılaştırma aşamasında kullanılacak olan metrik seçimi ise 3B nesne çıkarımı araştırmalarının cevapsız kalmış bir konusudur. Son olarak ise bölütleme ve çakıştırma gibi kimi zaman gerekebilecek işlemleri bu araştırmanın aşılması gereken engelleri olarak sayabiliriz.

2. Performans Değerlendirmesi

Bilgi çıkarım yöntemlerini değerlendirmek için kullanılan çeşitli ölçümler vardır. SHREC isimli 3B nesne çıkarım yarışmasında birazdan anlatılacak olan ölçümlerin herbiri dikkate alınmaktadır. Bunlar Hassasiyet-Anımsama (Precision-Recall), En Yakın Komşu (Nearest Neighbor), Birinci Sıra ve İkinci Sıra (First-tier and Second-tier), E-Ölçüsü, Kümülatif Kazanç (Discounted Cumulative Gain (DCG)), ve Normalize edilmiş Kümülatif Kazanç (Normalized Discounted Cumulative Gain (NDCG)):

- **Hassasiyet-Anımsama (Precision-Recall):**

Bilgi çıkarım algoritmalarının performanslarını ölçmek için kullanılan en yaygın istatistiktir. Hassasiyet, çıkarım sonuçlarında bulunan sorguya benzer (sorguyla aynı sınıftan olan) model sayısının (doğru pozitifler) çıkarılan toplam model sayısına oranıdır. İdeal olanı bu sayının bir olmasıdır, yani tüm çıkarılan modellerin sorguya benzer olmasıdır. Anımsama ise çıkarılan benzer model sayısının veritabanında bulunan sorgu benzeri (doğru pozitifler ve yanlış negatiflerin toplamı) toplam model sayısına oranıdır. İdealde anımsamanın bir olması beklenir yani benzer tüm modellerin ve sadece benzerlerin sorguyu yapana gösterilmesi.

$$\text{Hassasiyet} = \frac{\text{çkarılan benzer model sayısı}}{\text{çkarılan toplam model sayısı}}$$

$$\text{Anımsama} = \frac{\text{çkarılan benzer model sayısı}}{\text{veritabanında bulunan benzer model sayısı}}$$

- **En Yakın Komşu (Nearest Neighbor):**

Çıkarım sonucu sorguya benzer (sorguyla aynı sınıftan olan) modellerin listenin en önünde yer alma yüzdesi olarak ifade edilebilir. İdeal olan %100 olmasıdır.

- **Birinci Sıra ve İkinci Sıra (First-tier and Second-tier):**

Birinci-sıra, çıkarım sonucunun ilk N sırasındaki doğru pozitiflerin yüzdesidir. N ise veritabanında bulunan toplam pozitiflerdir. İkinci-sıra ise doğru pozitiflerin çıkarım listesindeki ilk 2*N sırasındaki yüzdesidir. İdeal olan birinci sıra için %100 ve ikinci sıra için ise %50'dir. Bu anlatımda sorgunun öğrenme grubunda değil de test grubunda olduğu varsayımı benimsenmiştir.

$$\text{Birinci Sıra} = \frac{\text{\#ilk N sırada yer alan benzer model}}{N} \times 100$$

$$\text{İkinci Sıra} = \frac{\text{\#ilk 2N sırada yer alan benzer model}}{N} \times 100$$

- **E-Ölçüsü:**

E ölçümü aşağıdaki gibi ifade edilir:

$$E - \text{Ölçüsü} = \frac{2}{\frac{1}{\text{Hassasiyet}} + \frac{1}{\text{Anımsama}}}$$

İdeal durumda E-Ölçüsü bir olmalıdır (hassasiyet=1, anımsama=1). E-Ölçüsü hesaplanırken çıkarılan toplam model sayısı 32 olarak kabul edilir. Bunun sebebi ise arama yapacak insanların genellikle sınırlı sayıda sonuçla ilgilenmesidir[16].

- **Kümülatif Kazanç (Discounted Cumulative Gain (DCG)):**

Bu ölçüt ilk defa Järvelin, K. ve Kekäläinen, J. [17] tarafından kullanılmıştır. Sonuçların sıralaması ağırlandırılarak hesaplama yapılır. Son gelen sonucun doğru negatif olması ilk gelen sonucun doğru negatif olmasından daha iyidir. Benzer şeyler doğru pozitifler için de söylenebilir. Kümülatif kazanç değerlerini hesaplamak için bir ilişki dizgisi kullanılır (RA). Bu dizginin elemanları şu şekilde belirlenir: indise karşılık gelen sonuç, sorgu ile aynı sınıftan ise bir olarak kabul edilir. İlişki dizgisi (RA) aşağıdaki gibi ifade edilir:

$$RA_i = \begin{cases} 1 & i^{\text{nci}} \text{ pozisyondaki sonuç sorgu ile aynı sınıftansa} \\ 0 & i^{\text{nci}} \text{ pozisyon farklı bir sınıftansa} \end{cases}$$

Kümülatif Kazanç inci derece için aşağıdaki gibi ifade edilir:

$$DCG_i = \begin{cases} RA_i & i = 1 \\ DCG_{i-1} + \frac{RA_i}{\log_2 i} & i > 1 \end{cases}$$

Sorgu için nihai kümülatif kazanç değeri ise şu şekilde hesaplanır:

$$DCG = \frac{DCG_{i_{\text{max}}}}{1 + \sum_{i=1}^N \frac{1}{\log_2 i}}$$

Burada i_{max} maksimum indisi belirtir (veritabanındaki toplam model sayısı) ve N de ilk sırada yer alan benzer sonuç sayısı olarak alınır. Yüksek kümülatif kazanç değerleri algoritmanın daha başarılı olduğu anlamına gelir.

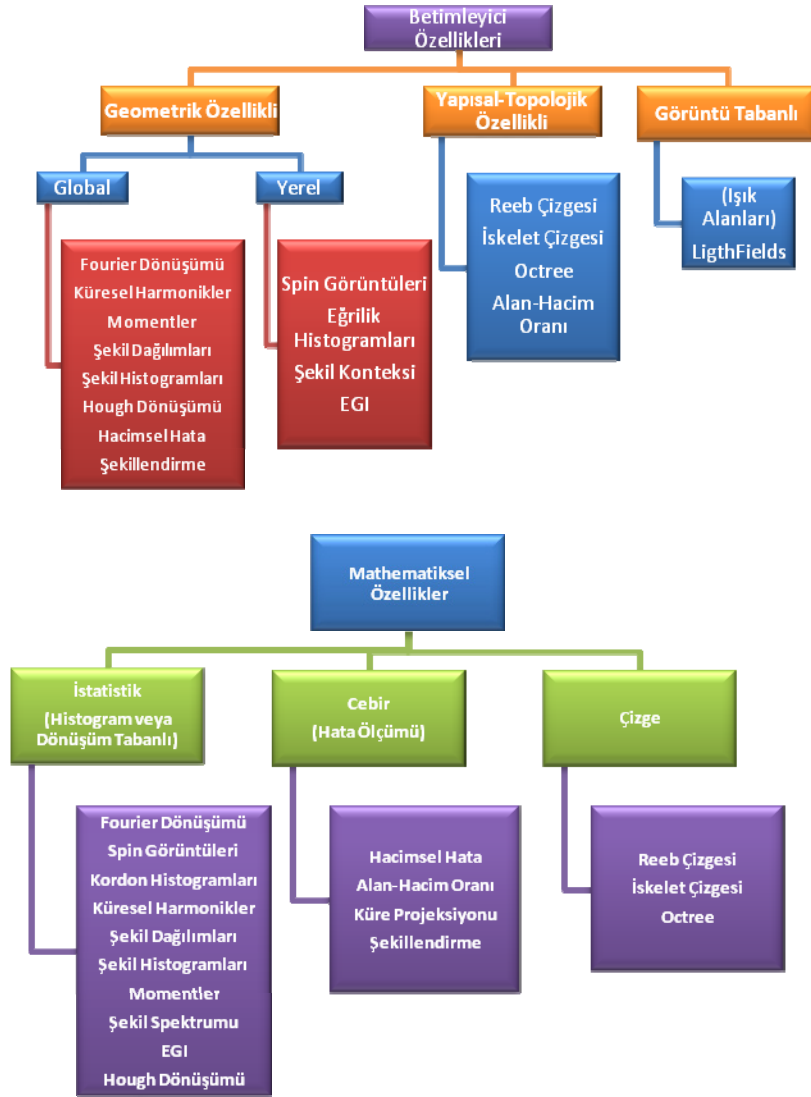
- **Normalize Edilmiş Kümülatif Kazanç (Discounted Cumulative Gain NDCG):**

Bu ölçüt çıkarım algoritmalarını birbiriyle kıyaslamak için kullanılır. Standart kümülatif kazanç ölçütünden farkı ise standard bir veritabanının kullanılmasıdır. Bu sayede algoritmanın başarısı hakkında hızlı bir fikir sahibi olunabilir. Eğer test edilen algoritmanın NDCG değeri pozitif ise ortalamanın üstünde bir başarıya sahip olduğuna NDCG değeri negatifse ortalamanın altında bir başarıya sahip olduğunu söyleyebiliriz. NDCG aşağıdaki gibi formüle edilir

$$NDCG = \frac{DCG_i}{DCG_{\text{avg}}} - 1$$

3. Çalışmalar

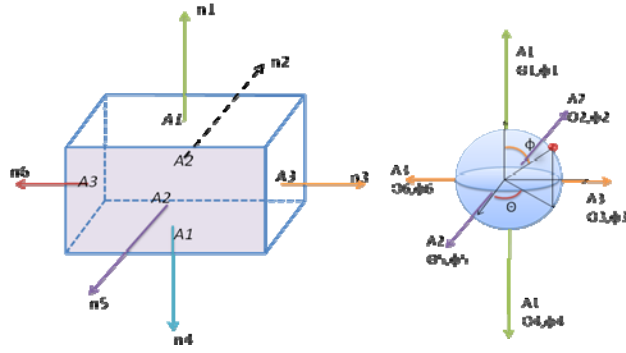
Çıkarım algoritmaları hakkında kısa bir özetin ardından bu bölümde 3B nesne tanımlama ve çıkarımı üzerine yapılan önemli çalışmalar detaylı olarak anlatılacaktır. Bu çalışmaların sınıflandırılması Şekil 8'de verilmiştir. Kaynak [10]'da verilen araştırma bu bölümde sıkça referans alınmıştır. Biz 3B tanımlama algoritmalarını betimleyici özellikleri ve kullandıkları matematiksel yöntemleri dikkate alarak iki farklı kategoride sınıflandırdık. Halihazırda bulunan bir çok çalışma istatistiksel betimleyicili global geometrik tabanlı sınıfa girer.



Şekil 8: 3B Tanıma Algoritmalarının Sınıflandırılması

Genişletilmiş Gauss Görünteleme (Extended Gaussian Imaging (EGI)):

EGI 1984 yılında Horn [18] tarafından önerilen bir yöntemdir. Sınıflandırmada küresel histogram tabanlı yüzey betimleme algoritmaları arasındadır. Histogram yüzey normallerinin yönlerinin biriktirilmesiyle elde edilir. Genellikle poligon ağ modellerinde kullanılan yöntemde yüzey normal yönlerinin bir kürede düzenli aralıkla ifade edildiğini (θ, ϕ) düşünebiliriz. Radyal bilginin yer almadığı bu gösterim biçimi basit bir dikdörtgenler prizması için Şekil 9'da verilmiştir. Normalin ait olduğu yüzeyin alan bilgisi de eklenerek ağırlıklı bir gösterim elde edilir. Yöntem yer değişime karşı karardır ve basitçe skala farklılıklarına karşı da duyarsız hale getirilebilir. Ancak yöntem dönmeye karşı değişkendir.



Şekil 9: Prizmanın Genişletilmiş Gauss Görüntüsü

Karmaşık Genişletilmiş Gauss Görüntüleme (Complex Extended Gaussian Imaging (CEGI)):

Kang ve Ikeuchi [19] standart EGI algoritmasına radyal bilgiyi de ekleyerek geliştirmişlerdir. Bu durumda histogram kompleks sayılarla $\alpha + j\beta$ şeklinde ifade edilebilir, burada α yüzeyin alanını β da yüzeyin merkeze olan uzaklığını göstermektedir.

Kordon ve Açık Histogramları (Cord and Angle Histograms):

Bu yöntem Paquet [20][21] tarafından önerilmiştir. Kordon modeldeki yüzey noktasını merkeze birleştiren vektör olarak tanımlanır. Poligon ağ modeller için yüzey noktaları üçgenlerin ağırlık merkezi olarak seçilir. Kordonların normalize edilmiş üç özelliğinin histogramları oluşturulur: i) kordon uzunluğu, ii) kordon ile birincil eksenin yaptığı açı, iii) kordon ve ikincil eksenin yaptığı açı. Yöntem gürültüye karşı hassastır ve dönme ve yer değiştirmeye karşı sabittir ancak skala farklılıklarına karşı değişkendir.

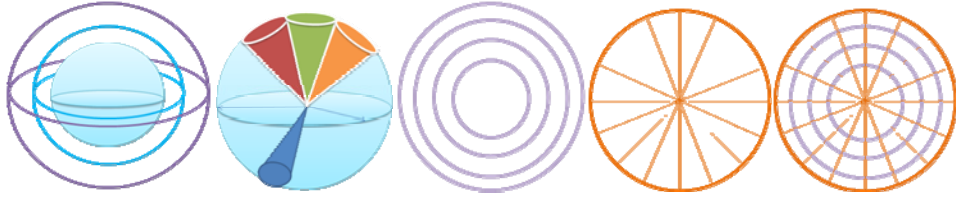
Momentler:

Momentler, özellikle Hu momentleri 2B görüntü tanımada sıkça kullanılmaktadır. Bu momentlerin 3B karşılıkları da mevcuttur, bunlar: geometrik momentler [22], Zernike momentleri [23], küresel momentler [24]. Momentler modele ait noktaların konumlarının dağılımı hakkında bilgi veren scalar büyüklüklere sahiptir. Basit betimleyiciler olmasına karşın skalaya, dönmeye ve yer değiştirmeye karşı değişkendir. Düşük derecedeki momentler geometrinin ana hatları hakkında bilgi içerirler bu yüzden çok fazla ayırıcı değildir, yüksek dereceli momentler ise detay hakkında bilgi verirler. Ancak dereceyi yükseltmek uzay, zaman ve gürültü hassasiyeti açısından fazladan yük getirecektir. Geometrik momentler aşağıdaki gibi formüle edilirler:

$$m_{ijk} = \sum_{p \in M} \sum_{q \in M} \sum_{r \in M} w_p x_p^i y_p^j z_p^k \quad p \text{ Modele (M) ait noktadır}$$

Şekil Histogramları:

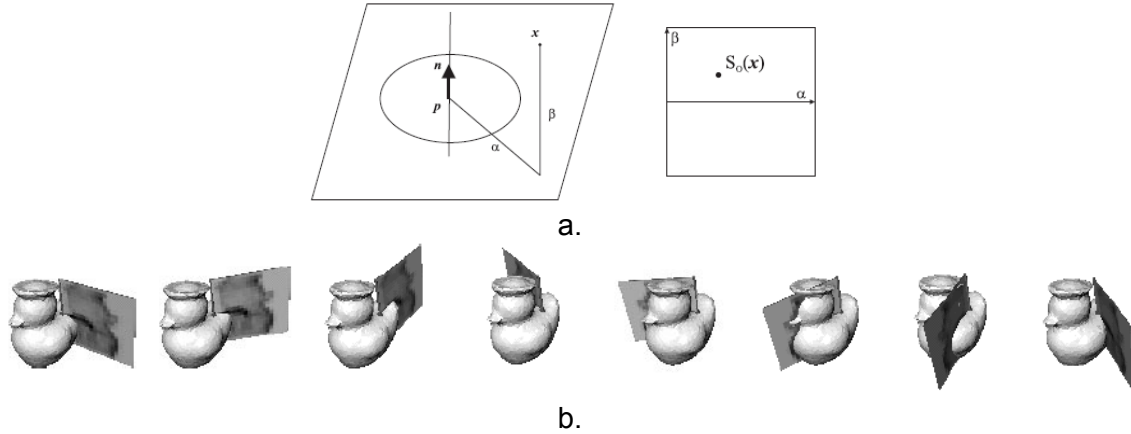
Ankerst ve ark. [25] tarafından kullanılmıştır. Yöntem uzayı konsentrik kürelere (kabuklara), sektör ya da bu ikisinin birleşimine böler. Kabuk modelinde sınır küresi konsentrik kürelere bölünür ve herbir bölme birer akümülatör gibi davranır. Akümülatörler, merkeze olan uzaklığı akümülatör kürenin yarıçapı içinde kalan nokta sayısını tutarlar. Referans [16]'da gösterildiği üzere basit ama zayıf bir betimleyicidir. Bu referansta verilen 12 algoritmanın karşılaştırıldığı Hassasiyet-Anımsama grafiğinin en altında yer alır. Sektör modeli 3B nesnenin yüzey alanının dağılımını küresel açının bir fonsiyonu cinsinden verir. Yine [16]'ıncı referansa göre sektör modeli kabuk modelinden daha iyi sonuçlar vermektedir. En iyi performans ise ikisinin birleşimi olan birleşik modelden gelir.



Şekil 10: Kabuk ve sector modeli ve kabuk, sektör ve bileşim modellerinin 2B projeksiyonları

Spin Görüntüleri:

Spin görüntüleri ilk defa Johnson ve Hebert [26] tarafından kullanılan lokal betimleyicilerdir. Yüzeydeki bir ilgi noktası kullanılarak 2Blu bir histogram elde edilir. Noktadan geçen normal vektörüne ihtiyaç vardır ve bu yüzden nokta kümeleri şeklinde ifade edilen modellerde kullanılamaz. Metodu şu şekilde özetleyebiliriz: nokta seti ve bunlara karşılık gelen normal vektörleri verilsin, $S = \{(p_i, n_i) \mid i = 1, 2, \dots, k\}$, modelin yüzey noktaları ilgi noktasına olan (α_i, β_i) uzaklıklarına göre akümüle edilir, merkezi ilgi noktası olan ve z ekseninde normal vektörünün yönünü gösterecek şekilde silindirik bir koordinat sistemi atanır. Histogramu elde etmek için yarıçap ve yükselti bileşenleri ayrıklaştırılır. Burada β_i , z ekseninde ölçülen uzaklık ve de radyal uzaklıktır (Şekil 11).



Şekil 11: a. Radyal (α) ve aksiyal (β) uzaklık b. Spin görüntü oluşturma aşamaları. Şekiller [27]inci referanstan alınmıştır.

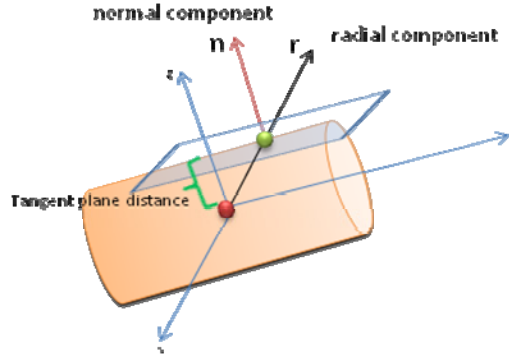
Şekil Dağılımları (Shape Distributions):

Osada ve ark. [28] şekil özelliklerinin dağılımını betimleyici olarak kullanmayı önermişlerdir. Nesnenin farklı özellikleri için histogram hesaplanır ve bazı metrikler (χ^2 , Bhattacharya, Minkowsky LN normları) kullanılarak diğer modellerle karşılaştırma yapılır. Aşağıdaki özellikler dağılımların hesaplanması için seçilmiştir:

- A3: Modelin yüzeyinde rasgele seçilmiş üç nokta arasındaki açı.
- D1: Sabit bir nokta ile yüzey üzerinde rasgele seçilmiş bir nokta arasındaki açı.
- D2: Yüzey üzerinde rasgele seçilmiş iki nokta arasındaki uzaklık.
- D3: Modelin yüzeyinde rasgele seçilmiş üç noktanın oluşturduğu alanın kare kökü
- D4: Modelin yüzeyinde rasgele seçilmiş dört noktanın oluşturduğu hacmin küp kökü

D2 dağılımı kordon histogramlarının benzeridir [21] [20] ancak Akgül ve ark. [11] [29] çalışmalarında yüzeyde gelişigüzel örnek noktalar seçmenin performansı arttırdığından bahsetmişlerdir.

Osada'nın çalışmasının benzeri [28], Akgül [11] tarafından yapılmıştır. Ancak Akgül histogramlar kullanmak yerine olasılık yoğunluk tahminini kernel kullanarak yapmıştır. Üç yerel geometrik özellik kullanılmıştır: radyal özellikler, tanjant düzlemi ve tanjant düzlemi ile radyal vektörün kros çarpımı (Şekil 12)



Şekil 12: [11] ve [29]'da kullanılan özelliklerin gösterimi

Işık Alanları (Light Fields):

Bu yöntem çoklu görüntüleri kullanır ve Cheng ve ark. [30] tarafından önerilmiştir. Veritabanında bulunan tüm modellere belirli açılardan bakılarak 2B silüetleri çıkarılmıştır. Daha sonra, çıkarılan bu silüetler için özellik vektörü hesaplanır ve silüetlerin özellik vektörlerinin kümesi betimleyici olarak kabul edilir. Silüet özellikleri için Zernike momentleri ve Fourier dönüşümü seçilmiştir. Yazarların deney sonuçlarına göre yöntem birçok algoritmanın üstünde bir başarı sağlamıştır.

Reeb Çizgeleri:

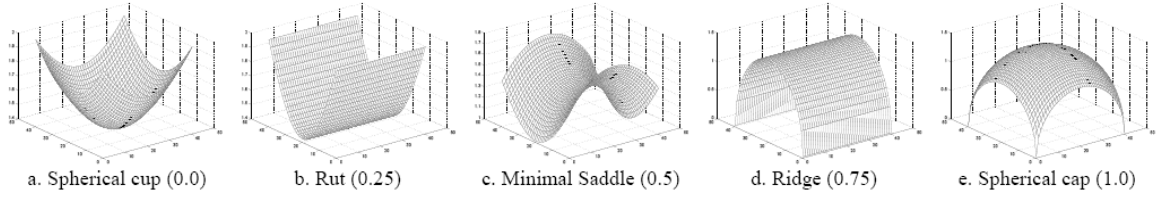
Çizge tabanlı topolojik bir yöntem olan Reeb çizgeleri Hilaga ve ark. [31] tarafından önerilmiştir. Modelin iskelet yapısı çizgesel olarak ifade edilmiş ve karşılaştırma için kullanılmıştır. Temel olarak model, μ foksyonu (noktaya ait eğrilik değeri veya noktanın geodezik uzaklığı) adı verilen bir kritere göre bölümlere ayrılır, ve bölütlerden herbiri çizgedeki bir düğüme karşılık gelir. Düğümler arasındaki bağlantılar ise komşu bölgerin \square fonksiyon ilişkisini gösterir.

Eğrilik Historamları (Curvature histogram (Surface Curvature, shape Spectrum)):

Bu yöntemde yüzey eğriliklerinin histogramından bir betimleyici elde edilir. Algoritma Zaharia ve Preteux [32] tarafından MPEG-7 dahilinde önerilmiştir. Şekil spektrum betimleyicisi ise yüzeye ait şekil indislerinin histogramlarının çıkarılmasıyla elde edilir. Şekil indisi ilk defa Koenderink [33] tarafından iki ana eksenin bir fonksiyonu olarak tanımlanmıştır. Bir kurallı yüzeye ait noktanın normal eğrisinin maksimum ve minimum değerleri κ_1 ve κ_2 ise şekil indisi (SI) şu şekilde ifade edilir:

$$SI = \frac{1}{2} - \left(\frac{2}{\pi} \right) \arctan \left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2} \right)$$

Betimleyici skala, yer değiştirme ve dönme farklılıklarına göre değişmez. Bazı temel şekillerin şekil indisleri Şekil 13'de verilmiştir.

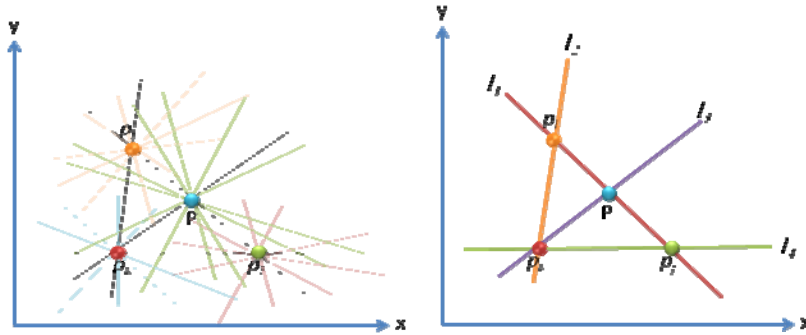


Şekil 13: Temel bazı şekillerin şekil indis değerleri, Resim [32]inci referanstan alınmıştır.

3B Hough Dönüşümü:

Hough dönüşümü görüntü işleme dünyasında sıkça kullanılan ve Hough [34] tarafından önerilmiş bir algoritmadır. 2B görüntülerde doğru, daire, elips ve düzlemleri bulmak için kullanılmıştır. Hough dönüşümü 3B modellerde de tanıma amacıyla ilk defa Zaharia ve Preteux[32] [35] tarafından denenmiştir. Bu yöntemdeki temel varsayım nesnelerin parça parça düzlemsel yapıda olmalarıdır.

3B Hough dönüşümünü anlatmak için doğru tanımda kullanılan uyarlamasından başlamak faydalı olacaktır. Aslında yöntem bir oylama esasına dayanır. Gruptaki her bir nokta, $S = \{P_1, P_2, \dots, P_n\}$, üstünden geçme ihtimali olan doğrular için bir oy kullanır (Şekil 14). Sürekli bir bölge için sonsuz sayıda doğru geçeceğinden uzay ayrıklandırılır.



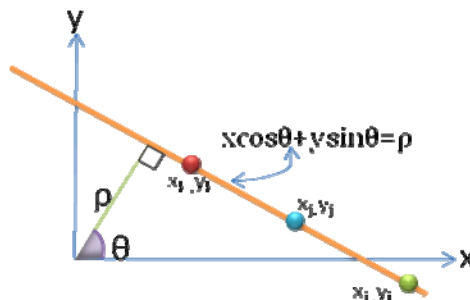
Şekil 14: Doğru tanımda Hough Dönüşümü

Herhangi bir doğru aşağıdaki denklemle ifade edilebilir:

$$y = mx + n$$

Buradaki amaç herbir (x,y) ikilisinin sağladığı (m,n) ikililerini bulmaktır. Fakat sabit bir x değeri için $m = \frac{y-n}{x}$ ya da $n = y - mx$ problemlidir. Bunu aşmak için doğru denklemi parametrik formatta yazılır:

$$x \cos \theta + y \sin \theta = \rho$$



Şekil 15: Parametrik doğru denklemi

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2},$$

$$-\sqrt{|x|_{\max}^2 + |y|_{\max}^2} \leq \rho \leq \sqrt{|x|_{\max}^2 + |y|_{\max}^2}$$

Benzer bir biçimde 3B düzlem denlemi aşağıdaki gibi ifade edilebilir:

$$ax + by + cz = d$$

$$x \cos \theta \cos \varphi + y \sin \theta \cos \varphi + z \sin \varphi = \rho$$

Herbir P=(x, y, z) noktası uzayın ayrıklandırma çözünürlüğüne bağlı olarak değişik (θ, φ, ρ) üçlüsü için oy kullanacaktır. Uzay seyrek aralıklara bölünürse yanlış bilgi kaçınılmaz olur ancak çok sık aralıklara bölmek de zaman ve uzay açısından çok verimsiz olur. Sahnedeki en olası düzlemin en çok oy alan olduğu yani en çok noktaya sahip olduğu varsayımı da yapılır

Ayrık Fourier Dönüşüm Betimleyicisi (Discrete Fourier Transform Descriptor):

Fourier Dönüşümü de 2B içerik tabanlı görüntü çıkarımında kullanılan popüler bir algoritmadır. Vranic ve Saupe [36] bu yöntemi 3B'a adapte etmişlerdir. Basitçe söylemek gerekirse model frekans uzayında ifade edilir. Modelin Fourier dönüşüm katsayıları kullanılarak (yüksek frekanslı bileşenleri gözardı edilerek) vektörel bir betimleyicisi elde edilir. Düşük frekanslı bileşenler modelin temel yapısal özellikleri hakkında bilgi verirken yüksek frekans bileşenleri daha çok modelin ayrıntısı hakkında bilgi taşırlar.

Bu betimleyici skala ve dönme farklılıklarına göre değiştiği için öncelikle bir voksel oluşturmak gereklidir. Ayrıca PCA gibi yöntemler kullanılarak poz normalizasyonu da sağlanmalıdır. Bu başlangıç aşamaları yöntemin dezavantajı gibi gözükse de [36]ncı referansta belirtildiği üzere zaman, uzay ve betimleme gücü açısından oldukça etkili bir yöntemdir. Voksellenmiş ve normalize edilmiş veri (g veriyi ve g_{ijk} tek bir vokseli ifade ederse) için Fourier dönüşümü aşağıdaki denklemlerle ifade edilir:

$$f_{uvw} = \frac{1}{\sqrt{N^3}} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} g_{ijk} e^{-j\frac{2\pi}{N}(iu+fv+kw)}$$

Küresel Harmonikler (Spherical Harmonics):

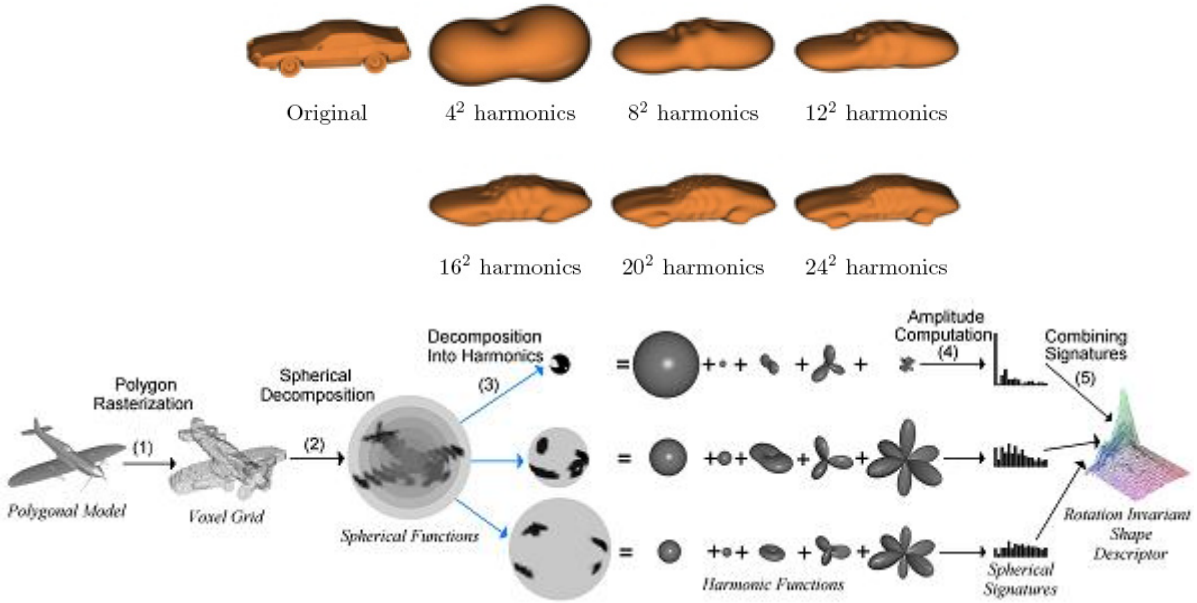
Küresel harmonikler (KH) 3B nesne eşleştirme gibi konularda kullanılan oldukça popüler bir yöntemdir. Kazhdan ve ark. [37] yöntemi "Küresel Harmonic Betimleyiciler" başlığı ile sunsalar da ilk defa kullananlar değillerdir. Küresel harmonikler 3B nesne tanımlamada daha önce Vranic ve ark. [38], Jiao ve ark. [39] tarafından kullanılmıştır. 3B nesne tanımlamada KH kullanan ve bizim ulaşabildiğimiz en eski çalışma Zribi'nin 1996'da [40] yaptığı çalışmadır. Kazhdan ve ark. [37] KH'lerin birçok yöntemden daha başarılı olduğunu deneysel olarak göstermişlerdir. Çalışmalarında EGI, radial dağılım (D2, kordon hist., vb), sektörler, şekil histogramları ve voksel betimleyicileri kullanmışlardır.

KH'lerin matematiksel anlamı bu kısımda anlatılacaktır. S2, R3'deki birim küre olsun, R3'deki bir V vektörü θ , $0 \leq \theta \leq \pi$, ve φ , $0 \leq \varphi \leq 2\pi$, olmak üzere $V=(\cos\varphi\sin\theta, \sin\varphi\sin\theta, \cos\theta)$ şeklinde ifade edilebilir. Böylelikle Hilbert Uzayında S2' de tanımlı karelerinin integrali alınabilir fonksiyonlar için iç çarpımı (inner product) aşağıdaki gibi ifade edilir [41]:

$$(f, g) = \int_0^\pi \left[\int_0^{2\pi} f(\theta, \phi) \overline{g(\theta, \phi)} d\phi \right] \sin \theta d\theta$$

Böylelikle KH'ler S2 için ortonormal baz oluştururlar. Sonuç olarak $f(\theta, \phi) = \sum \alpha_{lm} Y_l^m(\theta, \phi)$ olan herhangi bir fonksiyon aşağıdaki gibi yazılabilir:

Burada $\alpha_{lm} = \langle f, Y_l^m \rangle$, ve Y_l^m , l derecesinden m sınıfından bir Legendre fonksiyonu ve k_{lm} de normalizasyon sabitidir[41]. Daha ayrıntılı bilgi için [41]inci referansa bakılabilir. Küçük rütbeli frekans bileşenlerinin fazı ihmal edilip sadece büyüklükleri kullanılırsa ($|\alpha_{lm} Y_l^m(\theta, \phi)|$, L_2 normu) modelin vektörel betimleyicisi elde edilir. Kısaca küresel harmonikler Fourier Dönüşüm betimleyicilerinin daha genel bir uyarlamasıdır.



Şekil 16: Bir modelin Küresel Harmonik betimleyicileri , Resimler sırasıyla [42] ve [43] numaralı referanslardan alınmıştır

Burada 3B nesne tanımadaki kullanılan bütün yöntemler anlatılmasa da içlerinden en önemli olanları seçilerek sunulmuştur. Aralarında özellikle kusursuz modeller üzerinde yüksek başarı sağlayanlar olsa da bütün ihtiyaçlara cevap verebilecek evrensel bir betimleyici henüz keşfedilmemiştir.

4. 3B Nesne Tabanları ve Bilimsel Yarışmalar

3B tanıma ve çıkarımı konusunda çalışanlar için veritabanı çözümü bazı üniversite araştırma gruplarından ve ticari kaynaklardan gelmektedir. Princeton Şekil Çıkarım ve Analiz grubu herkese açık bir şekilde 1184 üçgen ağ formatında model sağlamaktadır [16]. Bu veritabanı birçok araştırmacı tarafından referans olarak alınmaktadır. Benzer bir biçimde Ulusal Tayvan Üniversitesi bilgisayar bölümü 10910 modelden oluşan .obj formatında bir veritabanına erişimi olanaklı kılmaktadır [44].

“National Design Repository”, Drexel üniversitesinde bulunan “Geometric and Intelligent Computing” laboratuvarı [45] ise 3B CAD ve katı modellere sahiptir. “3D CAD Browser” [46] adlı site ise 7600’ün üzerinde 3B model sağlamaktadır.

“AIM@SHAPE” [47], McGill üniversitesinde geliştirilmiş 3D Shape Benchmark [48], “Stanford Computer Graphics Laboratory” veritabanı arşivleri[49], ticari bir kuruluş olan Turbo Squid [50], 3D Cafe [51], ve 3dxtras [52] ise diğer 3B model sağlayıcıları arasında sayılabilirler.

Ayrıca her yıl 3B model çıkarımı üzerine AIM@SHAPE tarafından yarışmalar düzenlenmektedir.

5. Deneyler

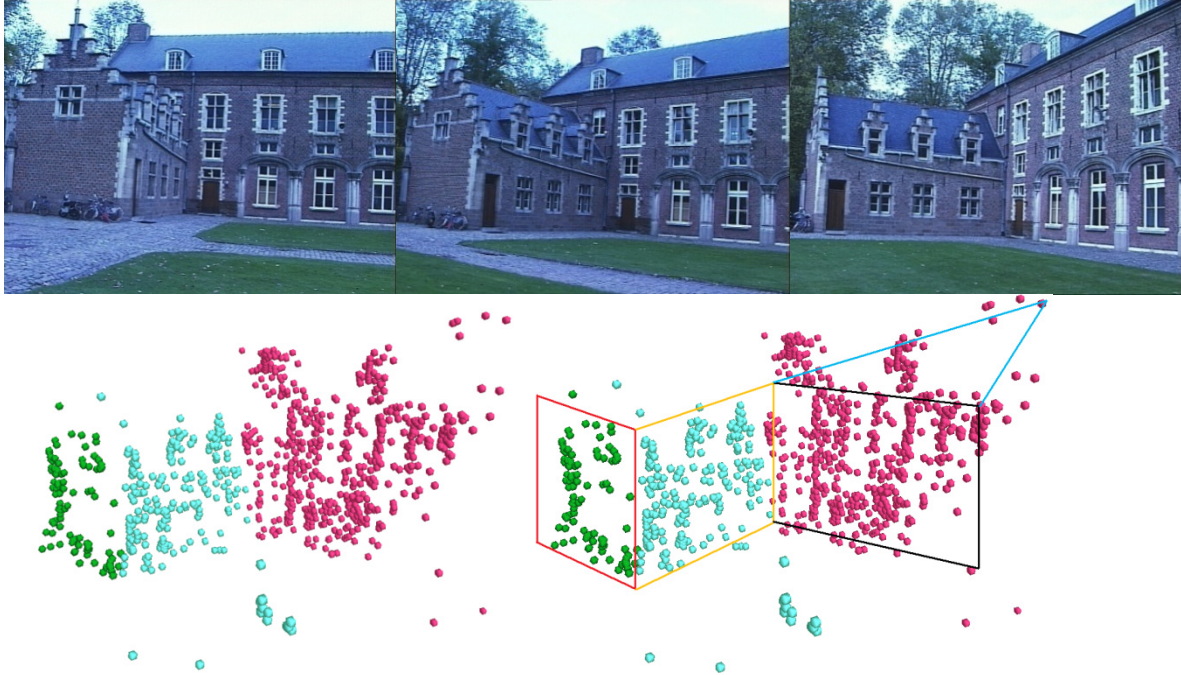
Daha önceki bölümlerde de anlatıldığı gibi kusursuz 3B model tanıma konusunun birçok zorlukları vardır. Ancak gerçek uygulamalarda rastladığımız veriler bu şekilde olmayıp, aşağıdaki özelliklerden en az birini barındırmaktadır:

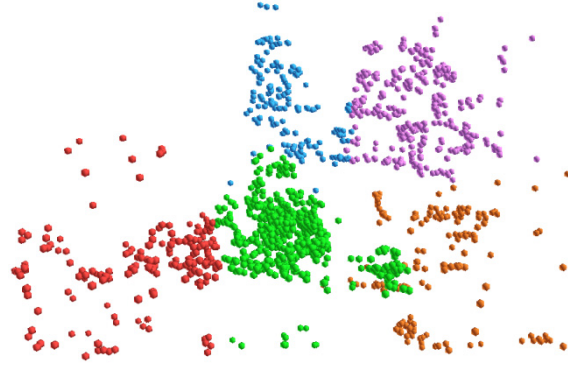
- Eksik model bilgisi (2.5B)
- Eksik yüzek bilgisi (nokta kümeleri)
- Gürültü
- Çok az bilgi (seyrek veri)
- Karışıklık
- Çakışma

Burada PSB veritabanından elde edilen hem kusursuz 3B modelleri (Şekil 20) hem de çoklu görüntülerden çıkarılmış 2.5B veriler üzerinde çalışılmıştır. Kusursuz modeller üzerinde Ayırık Fourier Dönüşümü, 2.5B nokta kümelerinde ise 3B Hough dönüşümü incelenmiştir. Sahne bölütlemesi için K-ortalamlar algoritması kullanılmıştır.

K-ortalamlar Algoritması :

Bu yöntem 3B nokta kümelerinden $F = \{P_1, P_2, \dots, P_n\}$ noktaların uzamsal benzerlikleri kullanılarak bölütleme yapar. İki sahne (kale, cevo) için Şekil 17’de sonuçlar verilmiştir. Nokta kümesinin elde edildiği çoklu görüntüler de Şekil 17’de gösterilmiştir.

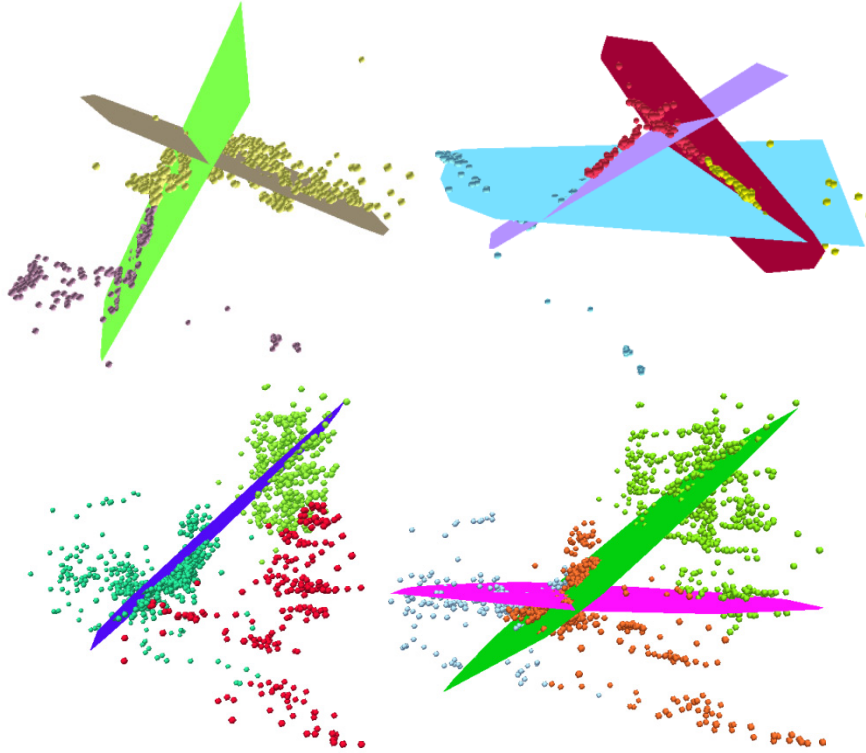




Şekil 17: İlk sıra: Kale verisinin elde edildiği çoklu görüntüler, ikinci sıra: Kale görüntülerinden elde edilen nokta kümesi ve 3 bölümlü k-means sonucu, üçüncü sıra: *Cevo* resmi ve *Cevo* sahnesinden elde edilmiş nokta kümesi

Hough Dönüşümü:

Hough dönüşümü zaman ve yer açısından karmaşık bir algoritmadır. Parametre belirlenmesi ve yerel maksimum belirleme diğer zorluklardır. Şekil 18'de *Cevo* ve *Kale* verileri için Hough dönüşüm sonucu elde edilen düzlemler gösterilmektedir.



Şekil 18: Hough dönüşümü ile elde edilen en iyi k-düzlem, *Kale* için $k=2,3$ ve *Cevo* için $k=1,2$

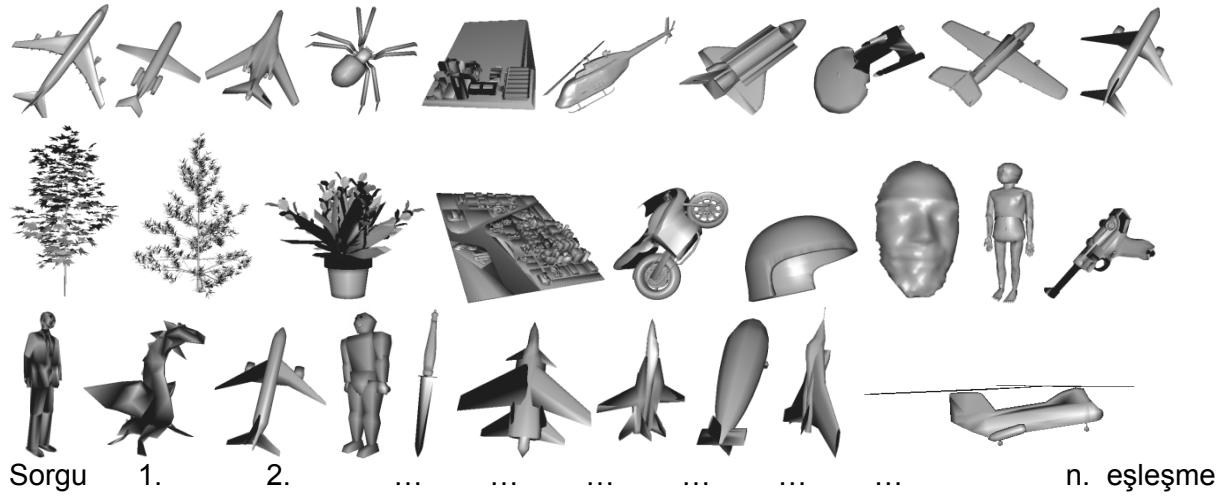
Ayrık Fourier Dönüşümü (Discrete Fourier Transformation (DFT)):

Klasik 3B Ayrık Fourier Dönüşümünde yüzey düzenli aralıklarla örneklendirilir ve dönüşüm bu veri üzerinde uygulanır. Ancak algoritmanın seyrek nokta kümeleri üzerindeki performansını ölçmek amacıyla bu çalışmada üçgen ağ modellerinin sadece köşe bilgileri kullanılmış, yüzey bilgisi ve yüzeyden örneklenen noktalar kullanılmamıştır. Modeller PSB

veritabanından elde edilmiş küçük bir gruba aittir. Vokselleme işlemi klasik metotta olduğu gibi gerçekleştirilmiştir. Ancak [36]nci referansta vksel değerleri modelin vksel içine düşen yüzey alanının toplam yüzey alanına oranını göstermektedir. Nokta kümeleri kullanıldığında ise vksel değerleri sadece bir ve sıfırlardan oluşmaktadır. Hatırlatmak gerekirse vkselleştirilmiş g verisi üzerinde dönüşüm aşağıdaki formülle ifade edilir:

$$f_{uvw} = \frac{1}{\sqrt{N^3}} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} g_{ijk} e^{-\frac{2\pi i}{N}(iu+ju+kw)}$$

Bu çalışmada $N=32$ olarak alınmıştır modeller ise $64 \times 64 \times 64$ 'lük gridlere bölünmüştür. Düşük rütbeli frekans bileşenlerini için f_{uvw} katsayılarının mutlak değerleri u, v, w parametreleri $-3 \leq u, v, w \leq 3$ aralığında alınarak elde edilmiştir. $\|f_{-u-u-w}\|, \|f_{000}\|, \dots, \|f_{uvw}\|$ vektörleri 11 normu kullanılarak karşılaştırılmıştır. Şekil 19'de Şekil 20'da verilen veritabanı ve DFT kullanılarak elde edilen sonuçlar verilmiştir.



Şekil 19: Ayrık Fourier Dönüşümü ile yapılan çıkarım sonuçları (soldan sağa: Sorgu imgesi ve sırasıyla elde edilen eşleme sonuçları)



Şekil 20: Deneylerde kullanılan veritabanı

6. Sonular

Tipik bir sahneden elde edilmiř 3B veri kumesinde nesnelere b6lütme en temel problem olarak ortaya çıkmaktadır. Bu amalar sahnede yer alan düzlemleri ayırıtırmaya yönelik alıřmalar yapılmıř ve 3B nokta kümesi iinde düzlemsel duran noktalar Hough algoritması yardımıyla ayırıtırılabilmiřtir. Ancak, sahnedeki düzlemlerin sayısı hala önemli bir parametre olarak durmaktadır. K-ortalamlar yönteminde ise uzamsal özellikler dikkate alınarak, nesnelere ayırıtırılmasına alıřılmıřtır. Deneysel sonuçlara göre her iki yöntemin de seyrek 3B sahne noktalarını kümeleme konusunda ok başarılı olmadığı gözlenmiřtir. 3B modeller kullanarak yapılan deneylerde ise 3B ayırık Kosinüs dönüşümü temelli yaklařımla yapılan deneylerde bazı sınıflar iin yüksek başarımlar, diđer sınıflar iin daha düşük sonuçlar elde edilmiřtir.

7. Referanslar

- [1]. Public. Wikipedia the free encyclopedia. Wikipedia. [Online] [Cited: 5 11, 2008.] <http://en.wikipedia.org>.
- [2]. 3D object recognition from range images using pyramid matching. Xinju Li, Igor Guskov. s.l. : Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.
- [3]. Recognizing Objects in Range Data Using Regional Point Descriptors. Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. s.l. : Proceedings of European Conference on Computer Vision (ECCV), Prague, Czech Republic, 2004.
- [4]. A new paradigm for recognizing 3d object shapes from range data. Salvador Ruiz-Correa, Linda Shapiro, and Marina Miela. s.l. : ICCV, Oct 2003.
- [5]. 3D free-form object recognition in range images using local surface patches. Hui Chen, Bir Bhanu. Pattern Recognition Letters, s.l. : Elsevier, 2007.
- [6]. Automated retrieval of 3D CAD model objects in construction range images. F. Bosche, C.T. Haas. Automation in Construction, s.l. : Elsevier, 2007. 17 (2008) 499–512.
- [7]. Dengsheng Zhang, Guojun Lu. Review of shape representation and description techniques. s.l. : Pattern Recognition Letters Elsevier.
- [8]. A Survey Of Free-Form Object Representation and Recognition Techniques. Flynn, Richard J. Campbell and Patrick J. Computer Vision and Image Understanding, s.l. : Academic Press, 2001.
- [9]. Funkhouser, Tom. [Online] [Cited: 5 9, 2008.] <http://www.cs.princeton.edu/courses/archive/spring06/cos426/index.html>.
- [10]. A survey of content based 3D shape retrieval methods. Veltkamp, J. W. H. Tangelder and R. C. Genoa, Italy : Proc. of the Shape Modeling International 2004 (SMI'04), 2004.
- [11]. Density Based Shape Descriptors and Similarity Learning for 3D Object Retrieval. Akgül, Ceyhan Burak. s.l. : PhD Thesis, 2007.
- [12]. Shilane, Philip Nathan. Shape Distinction For 3D Object Retrieval. s.l. : PhD Thesis, 2008.
- [13]. Goodall, Simon. s.l. : PhD Thesis, 2007.
- [14]. Kazhdan, Michael M. Shape Representations and Algorithms For 3D Model Retrieval. s.l. : PhD Thesis, 2004.
- [15]. Threedimensional shape searching: state-of-the-art review and future trends. N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. s.l. : Computer-Aided Design, April 2005. 37(5):509–530.
- [16]. The Princeton Shape Benchmark. Philip Shilane, Patrick Min, Michael Kazhdan, Thomas Funkhouser. s.l. : Shape Modeling International, Genoa, Italy, 2004.
- [17]. IR evaluation methods for retrieving highly relevant documents. Järvelin, K. & Kekäläinen, J. Athens Greece : Proceedings of the 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000.
- [18]. Extended gaussian images. Horn, B. s.l. : In Proceedings of the IEEE., 1984, Cilt 72, pages 1656–1678.

- [19]. The complex EGI: A new representation for 3D pose determination. Ikeuchi, S. B. Kang and K. s.l. : IEEE Trans. Pattern Analysis and Machine Intelligence, 1993. 15(7):707–721.
- [20]. Nefertiti: a query by content system for three-dimensional model and image databases management. Eric Paquet, Marc Rioux. Washington, DC, USA : Proc. of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, 1997.
- [21]. Description of shape information for 2D and 3D objects. E. Paquet, A. Murching, T. Naveen, A. Tabatabai, and M. Rioux. s.l. : Signal Processing:Image Communication, 2000. 16:103–122.
- [22]. The MPEG-7 standard and the content-based management of three-dimensional data: A case study. . Rioux, Eric Paquet and Marc. Florence, Italy : In IEEE International Conference on Multimedia Computing and Systems, 1999.
- [23]. 3D zernike descriptors for content based shape retrieval. Klein, M. Novotni and R. New York, NY, USA : Proc. of the 8th ACM symposium on Solid modeling and applications., 2003.
- [24]. 3D model retrieval with spherical harmonics and moments. In B. Radig and S. Florczyk, editors, Mustererkennung 2001 (DAGM 2001),. Saupe D., V. Vranic D. Munich, Germany : DAGM 2001, 2001.
- [25]. Nearest neighbor classification in 3D protein databases. Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. Heidelberg, Germany : In 7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99), 1999.
- [26]. Using spin images for efficient object recognition in cluttered 3D scenes. JOHNSON, A. E. AND HEBERT. s.l. : IEEE Trans. Pattern Anal. Mach. Intell., 1999.
- [27]. Spin-images: A representation for 3-D surface matching. Ph.D. thesis. Jonhson, A. E. s.l. : Robotics Institute, Carnegie Mellon University, 1997.
- [28]. Shape Distributions. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. s.l. : ACM Transactions on Graphics, October 2002. 21(4):807{832.
- [29]. Density-Based 3D Shape Descriptors. Akgül C.B., Sankur B., Yemez Y., and Schmitt F. s.l. : EURASIP Journal on Advances in Signal Processing, 2007. 32503.
- [30]. On Visual Similarity Based 3D Model Retrieval. Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen and Ming Ouhyoung. s.l. : EUROGRAPHICS , 2003. Volume 22 (2003), Number 3.
- [31]. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, Tosiyasu L. Kunii. Los Angeles, CA, USA : ACM SIGGRAPH 2001, 2001. 1-58113-374-X/01/08.
- [32]. Three-dimensional shape-based retrieval within the MPEG-7 framework. Preteux, T. Zaharia and F. San Jose CA : In Proceedings of the SPIE Conference 4304 on Nonlinear Image Processing and Pattern Analysis XII, 2001.
- [33]. Koenderink, J. Solid shape. s.l. : The MIT Press, Cambridge, Massachusetts, 1990.
- [34]. Hough, P. Methods and means for recognizing complex patterns. In US Patent, 1962.
- [35]. Shape-based retrieval of 3D mesh models. Preteux, T. Zaharia and F. Lausanne, Switzerland : Proc. of the IEEE International Conference on Multimedia and Expo, ICME 2002, 2002.
- [36]. 3D shape descriptor based on 3D Fourier transform. Saupe, D. V. Vranic and D. Budapest, Hungary : In Proc. of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS'01), September 2001.
- [37]. Rotation invariant spherical harmonic representation of 3D shape descriptors. M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Air-ela-Ville, Switzerland : In Proc. of the 2003 Eurographics, ACM SIGGRAPH symposium on Geometry Processing (SGP'03), . 2003.
- [38]. Tools for 3D-object retrieval: Karhunenloeve transform and spherical harmonics. Dejan V. Vranic, Dietmar Saupe, and J. Richter. France : Proceedings of the IEEE 2001 Workshop Multimedia Signal Processing, 2001.
- [39]. Invariant Descriptor of Fourier-Mellin for Indexing by Contents a Database of Three Diemsional Images. s.l. : IEEE, International Conference on Signal Processing, ICSP02, 2002.

- [40]. Set of Invariant Features for Three-Dimensional Gray-Level Objects by Harmonic Analysis. Mourad Zribi, Hubert Fonga and Faouzi Ghorbel. s.l. : Proceedings of ICPR '96 IEEE, 1996. 1015-4651/96.
- [41]. FFTs for the 2-sphere, Improvements and variations. HEALY, D. M., ROCKMORE, D. N., KOSTELEK, P. J., AND MOORE. 2003 : J. Fourier Analy. Appl.
- [42]. Tools for 3D-object retrieval: Karhunen-Loeve Transform and spherical harmonics. D. V. Vranic, D. Saupe, and J. Richter. Cannes, France : In: Proceedings of the IEEE 2001 Workshop Multimedia Signal Processing, October 2001.
- [43]. A search engine for 3D models. T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin and D. Jacobs. 2003 : ACM Transactions on Graphics. 22(1):83–105.
- [44]. On Visual Similarity Based 3D Model Retrieval. Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen and Ming Ouhyoung. s.l. : Computer Graphics Forum (EUROGRAPHICS'03), 2003, Cilt 22.
- [45]. designrepository.org. [Online] [Cited: 5 12, 2008.] <http://www.designrepository.org/>.
- [46]. 3D CAD Browser 3D Models. [Online] [Cited: 5 12, 2008.] <http://www.3dcadbrowser.com/>.
- [47]. Aim@Shape Project - Shape Repository. [Çevrimiçi] [Alıntı Tarihi: 13 5 2008.] <http://shapes.aim-at-shape.net/>.
- [48]. McGill 3D Shape Benchmark. [Çevrimiçi] [Alıntı Tarihi: 13 5 2008.] <http://www.cim.mcgill.ca/~shape/benchMark/>.
- [49]. Data Archives. [Online] [Cited: 5 12, 2008.] <http://graphics.stanford.edu/data/>.
- [50]. 3D Models, Plugins, Textures, and more at Turbo Squid. [Online] [Cited: 5 12, 2008.] <http://www.turbosquid.com/>.
- [51]. 3D Cafe. [Online] [Cited: 5 12, 2008.] <http://www.3dcafe.com/>.
- [52]. 3DXtras.com - Download Absolutely Free 3D Models. [Online] [Cited: 5 12, 2008.] <http://www.3dxtras.com/>.

2.3. KAMERA HAREKET ANALİZİ

Kamera hareket analizi bir çok problemde anlamsal bilgiye ulaşmak için kullanılabilir. Kamera hareketinin tespiti, nesne bölütleme öncesinde önemli bir ön adım olmakla birlikte, kamera hareketlerinin belirlenmesi de, videoyu anlamlı parçalara bölme ve videoda yer alan önemli bölümlerin bulunması amacıyla kullanılabilir. Bu kısımda önce kamera hareketinin tespiti için öneriler yapıldıktan sonra, daha sonra kamera hareketlerinin sınıflandırılması gerçekleştirilecektir.

2.3.1 Kamera Hareket Tespiti

İlgili literatür incelendiğinde, daha önce bu konuda yapılan çalışmaların birçoğunun kamera hareket tespiti ile birlikte hareket parametrelerinin nicel çıkarımına yönelik olduğu görülebilir [1]-[13]. Bu çalışmalardan bir kısmı ardışık görüntüler arasındaki optik akışın hesaplanmasını gerektirir [4],[6],[12],[13]. Optik akışın çıkarımı işleminin hesaplama karmaşıklığı yüksek olduğu için diğer bir yaklaşım, hareket yöneylerinin sıkıştırılmış MPEG verilerinden çıkarımına yönelmiştir [2],[3],[5],[8],[10]. Bununla birlikte, kodlanmış verideki hareket vektörleri, görüntülerde hareketli nesne, bölgesel sabit yeğlilik ve bölgesel eş doku içeren alanlarda hatalı sonuçlar vermesi, kodlama işleminde hatalar (niceleme hatası gibi) oluşması ve en genel durumu (MPEG kodlanmamış bütün veriler) içermemesi nedeni ile bazı çalışmalarda tercih edilmemiştir. Bu iki grubun (kodlanmış MPEG verisi ve optik akış) dışında hız ve performans açısından kabul edilebilir sonuçlar vermesi nedeni ile bazı çalışmalarda nadir öznitelik noktaları (ve öznitelik eşleştirme) kullanılmıştır [1],[7],[9],[11].

Bu çalışmada, en genel durumu da içermesi nedeni ile hareket yöneylerinin çıkarımında yukarıda anlatılan son yöntem, nadir öznitelik noktalarının eşleştirilmesi yaklaşımı tercih edilmiştir. Bununla birlikte, [14]'deki gibi kamera hareket parametrelerinin nitel çıkarımı gerekmediğinden, görüntüler arasındaki hareketin ifadesinde istatistik ve parametrik modeller yerine hızlı çalışacak eşikleme yöntemi kullanılmıştır. Aynı şekilde, kamera hareketinin bulunmadığı durumlarda, bağımsız hareket eden nesnelerin varlığı [6]'da verilen ve bağımsız hareket eden nesnelerin konumlarını bölgesel olarak belirleyen karmaşık yöntemin aksine eşikleme ile belirlenmektedir.

Kamera Hareket Tespitinin ayrı bir kalemde ele alınmasının sebebi, kamera hareket analizi işlemlerinin göreceli olarak yüksek karmaşıklığa sahip olmasıdır. Bu nedenle öncelikli olarak kamera hareketinin varlığının tespiti, toplam algoritma sürecinde iyileştirme sağlayacaktır. Bu doğrultuda, kamera hareketinin olmadığı durumlarda, hareket analizi işlemleri çalıştırılmayacak ve toplam zamansal karmaşıklık düşürülecektir.

Geliştirilen algoritmadaki ana fikir görüntü içerisindeki görsel olarak belirgin alanlardaki hareketi inceleyip, kamera hareketinin varlığına karar verme üzerine kuruludur. Bununla beraber, kamera hareketinin bulunmadığı durumlarda bağımsız hareket eden nesnelerin varlığı da belirlenmektedir. Yapılan testlerde önerilen algoritmanın gerçek zamandan daha hızlı çalıştığı görülmüştür. Ardışık video kare çiftlerine uygulanan kamera hareketi tespit algoritması adımları aşağıda verilmiştir:

- i. İlk görüntüde (ardışık video karelerinden zamansal ekseninde önce gelen kare) her piksel için Harris ([15]) tepkisi hesaplanır.
- ii. Birinci görüntü [6]'da verilen yaklaşıma benzer bir biçimde bloklara (tipik olarak 8x8) ayrılır ve her blok içindeki Harris tepki değerleri toplanır.

- iii. Bir bloğun içindeki Harris tepki değeri toplamı belirli bir eşik değerinden (T_s) büyük ise bu blok belirgin alan (takip edilebilirlikleri yüksek alanlar) olarak kabul edilir ve kamera hareket tespiti işleminde değerlendirmeye katılır (Bu özelliğe sahip blokların sayısı N_v ile ifade edilsin).
- iv. Harris tepki değeri toplamı T_s eşik değerinin üzerinde olan her blok içindeki en keskin (Harris tepki değeri en yüksek) köşe pikseli bulunur.
- v. Elde edilen köşe noktalarının bir sonraki video karesindeki yerleri piramitsel KLT ([16]) algoritması ile hesaplanır, bu şekilde belirgin alanlardaki (Harris tepki değeri toplamı T_s eşik değerinin üzerinde olan bloklar) hareket yöneyleri elde edilmiş olur (Büyüklüğü sıfırdan farklı hareket yöneylerinin sayısı N_m ile ifade edilsin).
- vi. N_m / N_v oranı (görüntüde takip edilebilir belirgin alanlardaki hareket oranı) belirli bir eşik değerinin (T_m) üzerinde ise ardışık görüntülerde kamera hareketinin var olduğu kabul edilir. Aksi takdirde, kamera hareketsiz kabul edilir. Birinci görüntüde T_s eşik değerini aşan bloklar mevcut değilse ($N_v = 0$), kamera hareketinin saptanması için görüntüde yeterli bilgi olmadığı kabul edilir. Bu durumda, videoda bir önceki görüntü çiftinden elde edilen kamera hareketi sonucu kullanılır.
- vii. Kamera hareketinin var olmadığı durumda, görüntü içerisinde bağımsız hareket eden nesnelere varlığı N_m sayısının belirli bir eşik değerini (T_i) aşması ile kontrol edilir.

2.3.1.1 Deneysel Sonuçlar

Yukarıda adımları verilen kamera hareket tespiti algoritması iki ayrı veriye uygulanmıştır. Örnek olarak, Şekil 21'de iki farklı durumu (bağımsız hareket eden nesne içermeyen hareketli ve hareketsiz kamera durumları) ifade edecek şekilde ardışık video kareleri verilmiştir. Şekil 21 a) ve b)'de verilen görüntüler yaklaşık olarak 1 piksellik sağa yatay dönme hareketi içermekteyken, c) ve d)'de verilen görüntüler hareket içermemektedir. Buna göre, geliştirilen algoritma a) ve b) arasında kamera hareketi saptamış, c) ve d) arasında ise kamera hareketi tespit etmemiştir. Son olarak, e) ve f)'de bağımsız hareket eden nesne içeren hareketsiz kamera durumuna örnek ardışık video kareleri için algoritma, görüntülerde kamera hareketi olmadığını fakat bağımsız hareket eden nesne bulunduğunu tespit etmiştir. Sonuçlar, Tablo 1'de özetlenmiştir.



(a)



(b)



(c)



(d)



(e)



(f)

Şekil 21: a), b) Yaklaşık 1 piksellik sağa yatay dönme (pan right) içeren ardışık video kareleri (100. ve 101. kareler). c), d) Kamera hareketi içermeyen ardışık video kareleri (200. ve 201. kareler). e), f) Kamera hareketi içermeyen fakat bağımsız hareket eden nesne barındıran ardışık video kareleri (32. ve 33. kareler).

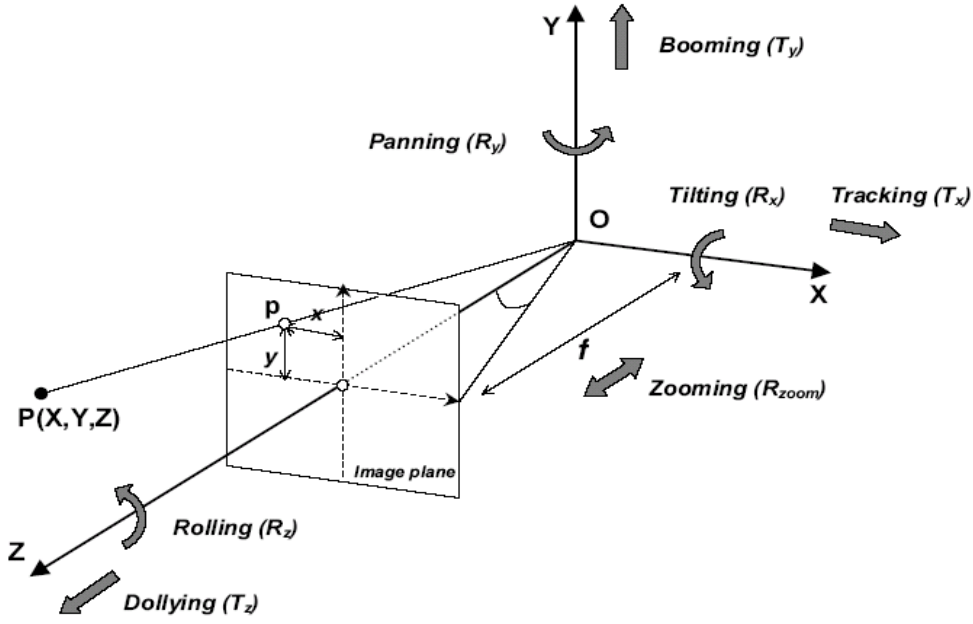
Tablo 1: Şekil 21’de verilen video kareleri için kamera hareket tespiti algoritması sonuçları

	Kamera Hareketi	Bağımsız Hareket Eden Nesne
Şekil 21a - b	Mevcut	Mevcut Değil
Şekil 21 c - d	Mevcut Değil	Mevcut Değil
Şekil 21 e - f	Mevcut Değil	Mevcut

2.3.2 Kamera Hareket Analizi

Kamera hareketi analizinde kullanılan 3-B hareket modeli 3 dönme (R_x , R_y , R_z), 3 öteleme (T_x , T_y , T_z) ve 1 odaklanma (R_{zoom}) olmak üzere toplam 7 parametre ile ifade edilmektedir [17]. Kamera odak noktası, kullanılan kartezyen koordinat sisteminin merkezinde konumlandırılarak elde edilen bu parametreler Şekil 22’de gösterilmiştir.

Bağımsız hareket eden nesnelerin bulunmadığı, aynı sahneye ait iki görüntüde kamera hareketinin görüntü düzlemleri üzerinde yarattığı hareket yöneyleri bu 7 parametre cinsinden (1)’de verilmiştir [17]. Bu denklemlerde, u_x ve u_y görüntülerde sırası ile x ve y eksenini yöndeki hareketi, Z görüntünün (x,y) konumundaki sahne derinliğini ve f de kameranın odak uzaklığını vermektedir.



Şekil 22: Kamera hareketi modelinde kullanılan dönme, öteleme ve odaklanma parametreleri [18].

$$u_x = -\frac{f}{Z} \cdot (T_x - xT_z) + \frac{x \cdot y}{f} \cdot R_x - f \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_y + y \cdot R_z + f \cdot \tan^{-1}\left(\frac{x}{f}\right) \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_{zoom} \quad (1)$$

$$u_y = -\frac{f}{Z} \cdot (T_y - yT_z) - \frac{x \cdot y}{f} \cdot R_y + f \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_x - x \cdot R_z + f \cdot \tan^{-1}\left(\frac{y}{f}\right) \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_{zoom}$$

Kullanılan yöntemde, Z-ekseni yönündeki kamera hareketi (*dolly*, T_z), odaklanma parametresine (R_{zoom}) bağlı olduğu için dikkate alınmamıştır ($T_z = 0$) [13]. Dolayısı ile Z-ekseni yönündeki kamera hareketleri, kamera odaklanma değişimi olarak değerlendirilmiştir. Buna göre, (1)'deki yöneylerin dönme ve öteleme bileşenleri tekrar (2)'de verilmiştir.

$$\begin{aligned} u_x^{\text{öteleme}} &= -\frac{f}{Z} \cdot T_x \\ u_y^{\text{öteleme}} &= -\frac{f}{Z} \cdot T_y \end{aligned} \quad (2)$$

$$u_x^{\text{dönme}} = \frac{x \cdot y}{f} \cdot R_x - f \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_y + y \cdot R_z + f \cdot \tan^{-1}\left(\frac{x}{f}\right) \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_{zoom}$$

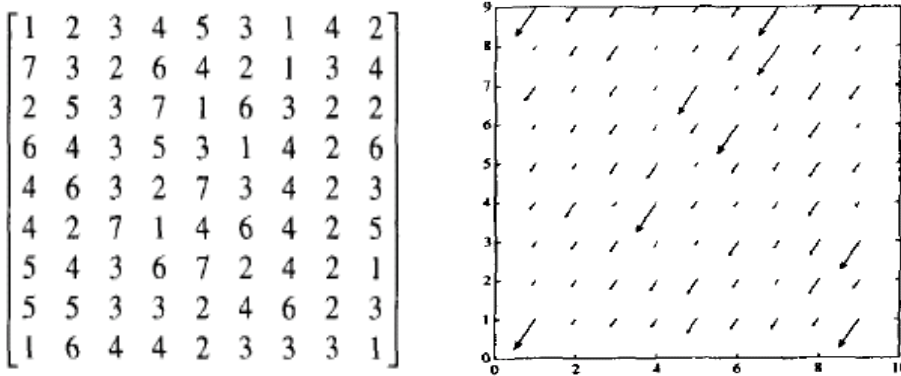
$$u_y^{\text{dönme}} = \frac{x \cdot y}{f} \cdot R_y + f \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_x - x \cdot R_z + f \cdot \tan^{-1}\left(\frac{y}{f}\right) \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_{zoom}$$

Kamera hareketine ait 6 parametrenin bulunmasında görüntüler arasındaki optik akış kullanılmaktadır. Algoritmanın ana fikri şu şekilde özetlenebilir: dönme hareketi ve odaklanmadan kaynaklanan optik akış bileşenleri toplam optik akıştan çıkartıldığında geriye kalan akış yöneyleri sadece öteleme hareketinden kaynaklanır ve bu yöneyler birbirine paraleldir.

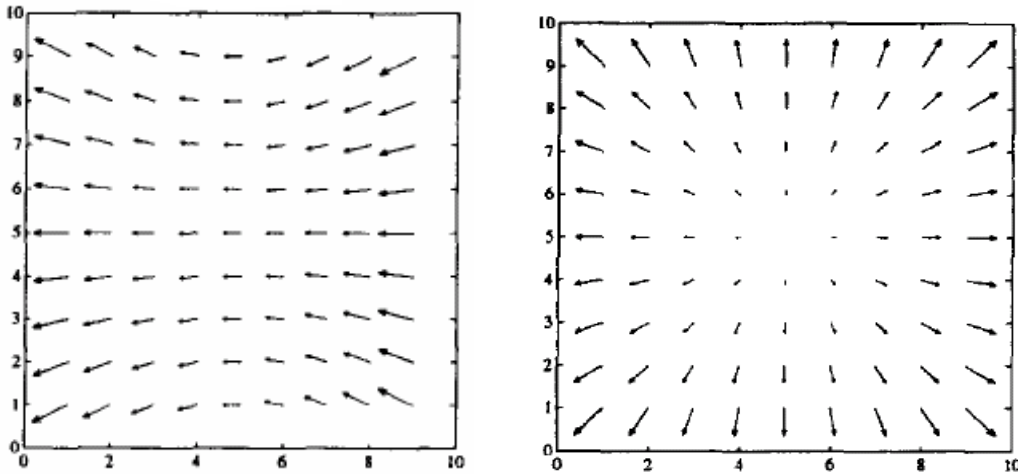
Buna fikre ek olarak, dönme ve zum hareketinden kaynaklanan akış yöneyleri sahne derinliğinden bağımsız iken, öteleme hareketinden kaynaklananlar ise sahne derinliğine bağlıdır (Şekil 23 ve Şekil 24). Bu fikri kullanarak, dönme (Rx, Ry, Rz) ve zum (Rzoom) parametreleri için çıkarma işleminden sonra paralel yöneyler kalacak şekilde en uygun değerler hesaplanabilir. Bu amaç doğrultusunda, çalışmalarımızda döngülü bir yöntem olan *simplex* eniyileme algoritmasını kullandık. Kullanılan ceza fonksiyonu aşağıda verilmiştir [13].

$$C = \sum \|u_i^{\text{öteleme}}\|^2 * \theta_i \quad (3)$$

(3)'de verilen eşitlikte $u_i^{\text{öteleme}}$, dönme ve zum ile ilgili bileşenin çıkarımından sonra kalan optik akış yöneylerini ve θ_i bu yöneylerin açisal yönlerini ifade etmektedir.



Şekil 23: Sahne derinliği (solda) ve salt öteleme kamera hareketinden kaynaklanan optik akış (sağda) [13].



Şekil 24: Salt yatay dönme (pan, solda) ve odaklanma (sağda) öteleme kamera hareketlerinden kaynaklanan optik akış (sağda) [13]

(2)'de verilen optik akış dönme ve odaklanma bileşenlerinin eniyilemesinde izlenen üç farklı yöntem mevcuttur. Bunlardan ilki, kamera görüş alanının (FOV) bilindiği varsayımı üzerine kuruludur (bu durumda x/f ve y/f de bilinmektedir) ve çok hassas sonuçlar (özellikle görüş alanının geniş olduğu durumlarda) sunmaktadır. Bu varsayım en genel durumda (yayınsal verilerde) geçerli olmadığı için bu yöntem tercih edilmemiştir. İkinci yaklaşım, dar görüş alanı varsayımına dayanmaktadır ($x/f \ll 1$, $y/f \ll 1$). Bu durumda (2)'deki optik akış dönme bileşenleri (4)'de verilen hali alır. Bu yöntem, geniş görüş alanına sahip görüntülerde iyi sonuç vermediği için tercih edilmemiştir.

$$\begin{aligned} u_x^{\text{dönme}} &= -f.R_y + y.R_z + x.R_{\text{zoom}} \\ u_y^{\text{dönme}} &= f.R_x - x.R_z + y.R_{\text{zoom}} \end{aligned} \quad (4)$$

Üçüncü ve son yöntemde, dönme ve odaklanma parametrelerine ek olarak kamera odak uzaklığı da bilinmeyen olarak kabul edilip optimizasyon işlemine dahil edilmektedir. Kullandığımız algoritmanın gerçekleştiriminde en genel durumu dâhil etmesi nedeni ile bu yöntem tercih edilmiştir.

2.3.2 Deneysel Sonuçlar

Şekil 25'de, Kamera Hareket Analizinden geçirilen ve salt sağa yatay dönme içeren ardışık video kareleri verilmiştir. Analiz ile elde edilen sonuçlar $[Tx Ty Rx Ry Rz Rzoom]$ formatında sırası ile şu şekildedir (pozitif yatay dönme değeri sağa doğru dönüş anlamındadır):

- a)-b) arasında $[0.056 \ 0.015 \ 0.077 \ 0.903 \ 0.000 \ -0.066]$,
- b)-c) arasında $[-0.080 \ -0.028 \ -0.069 \ 1.005 \ 0.000 \ 0.102]$,
- c)-d) arasında $[0.022 \ 0.005 \ 0.007 \ 0.921 \ 0.000 \ 0.015]$.

Şekil 26'de sola yatay dönme ve pozitif odaklama karma hareketini içeren ardışık video kareleri verilmiştir. Kamera hareket analizi ile elde edilen sonuçlar sırası ile şu şekildedir:

- a)-b) arasında $[0.0053 \ 0.000 \ 0.417 \ -2.819 \ -0.001 \ 1.144]$,
- b)-c) arasında $[-0.086 \ -0.004 \ 0.275 \ -3.113 \ 0.001 \ 0.701]$,
- c)-d) arasında $[-0.307 \ 0.0525 \ 0.386 \ -3.457 \ -0.000 \ 1.252]$.

Şekil 27'de kamera hareket tespiti ve analizi için geliştirilen yazılımın kullanıcı ara yüzü verilmiştir.



(a)

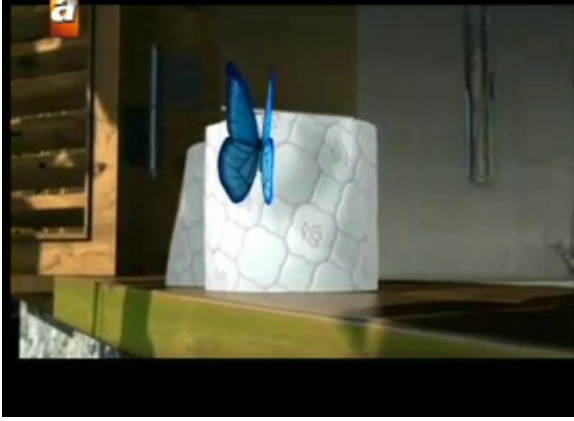


(b)



(c)

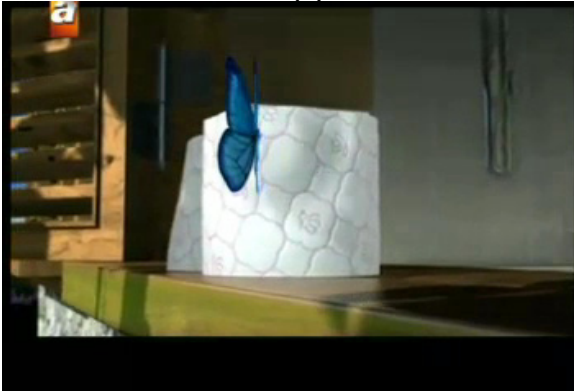
Şekil 25: a), b), c) Salt sağa yatay dönme (pan right) içeren ardışık video kareleri (sırası ile 10., 11., ve 12. kareler).



(a)

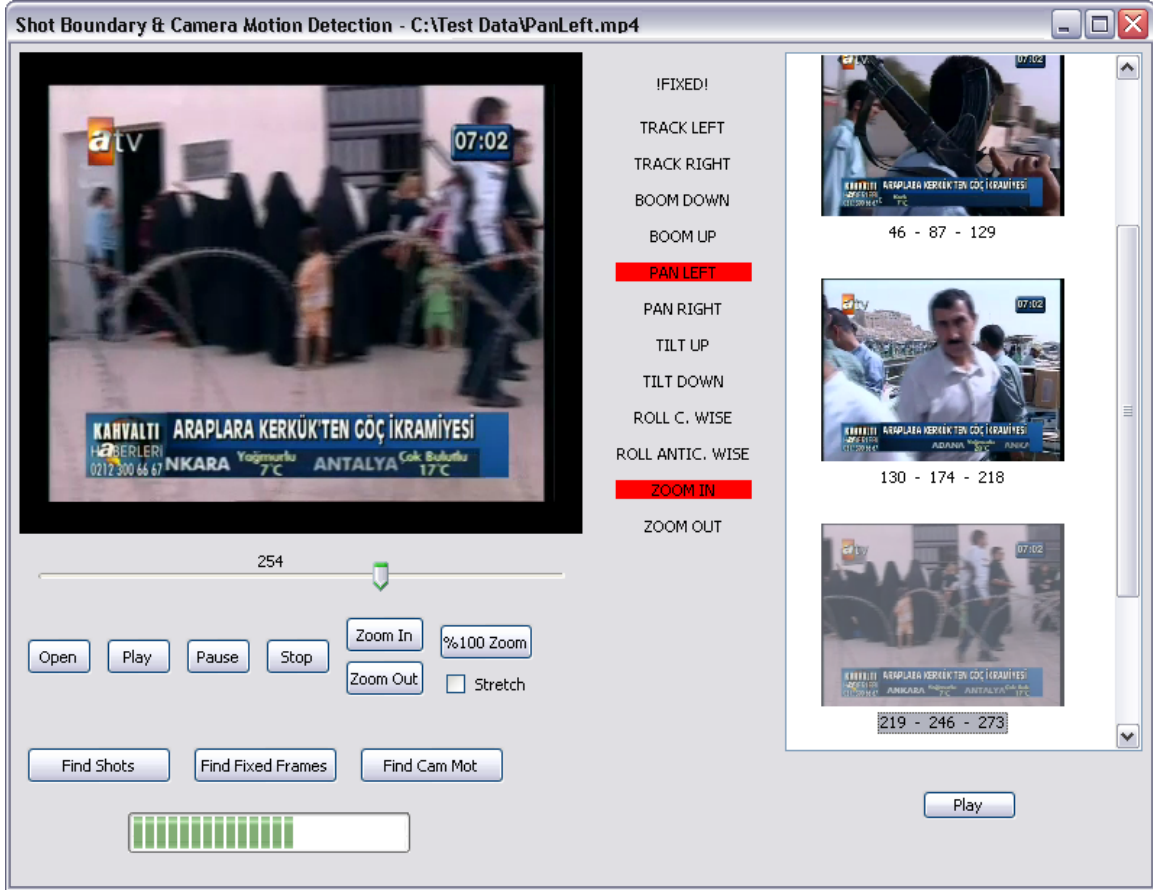


(b)



(c)

Şekil 26: a), b), c) Sola yatay dönme ve pozitif züm karma hareketini içeren ardışık video kareleri (sırası ile 54., 55., ve 56. kareler).



Şekil 27: Kamera hareket tespiti ve analizi için geliştirilen yazılımın kullanıcı ara yüzü.

2.3.3 Sabit Kamera Çekimleri için Hareketli Nesne Bölütlemesi

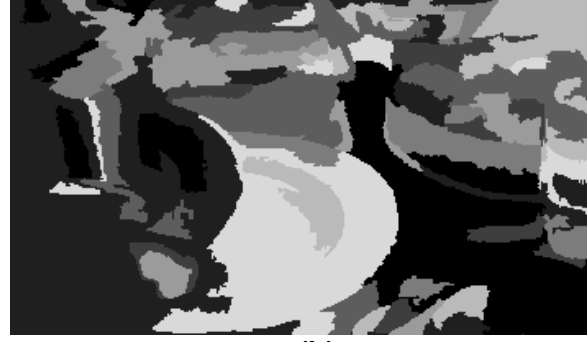
Kamera hareket tespiti ile elde edilen sabit görüntülerdeki hareketli nesnelerin bölütlenmesinde bölgesel tabanlı (piksel tabanlına aksine) bir yaklaşım izlenmiştir. Buradaki amaç, benzer özellikler gösteren (renk değerleri) bölgelerin aynı nesnelere ait olduğu varsayımı altında bu bölgelerin hareket özelliklerine göre bölütlenmesidir. Buna göre, geliştirilen algoritma üç ana adımda özetlenebilir:

- i. Görüntünün bölgesel renk değerlerine göre aşırı bölütlenmesi.
- ii. Bölütlerin optik akış yöneylerinin hesaplanması (ardışık görüntülerdeki bölütler arası hareketin ifadesinde ötelemenin yeterli olduğu varsayılmıştır).
- iii. Aşırı bölütlenmiş alanların hareket verilerine göre birleştirilmesi.

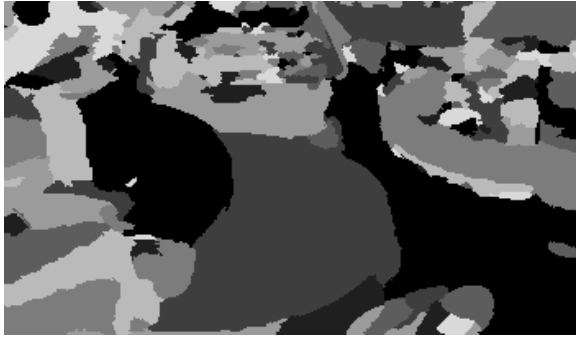
İlk aşama olan aşırı bölütleme işlemi için histogram tabanlı *K-Ortalamalar (K-Means)* ve piksel tabanlı *Ortalama Kayma (Mean Shift)* [22] algoritmaları uygulanmıştır. Yapılan çalışmada, aşırı bölütleme işleminde *K-Ortalamalar* algoritmasının daha hızlı olduğu gözlemlenmiş fakat *Ortalama Kayma* algoritmasının daha tutarlı ve nesne sınırlarını iyi belirleyen sonuçlar verdiği görülmüştür (Şekil 28). Bu nedenle, uygulamada *Ortalama Kayma* algoritması tercih edilmiştir.



(a)



(b)



(c)

Şekil 28: K-Ortalama ve Ortalama Kayma algoritmaları aşırı bölütleme karşılaştırmalı sonuçları, a) Orijinal görüntü, b) K-Ortalama bölütleme sonucu, yanlış bölütleme sonucu sol kısımda beyaz tabak, arka plan ve el, aynı bölüt içerisinde çıkmıştır, c) Ortalama Kayma bölütleme sonucu.

İkinci adım olan bölüt hareket yöneylerinin hesabında iki kademeli bir yöntem izlenmiştir. Birinci kademe, köşe noktaları bol bulunan bölütlerde piramitsel KLT izleme ([16]) ile bölüt hareket yöneyinin bulunmasını içermektedir. Buna göre, bir bölütteki köşe noktası sayısı belirli bir eşğin üzerinde ise bu köşe noktaları KLT izleme ile takip edilmektedir. Elde edilen hareket yöneylerinin ortanca değeri o bölütün hareket yöneyi olarak atanır. Ortalama yerine ortanca değer kullanılması sebebi, aşırı bölütleme ve KLT izlemede oluşacak hataları en aza indirmektir. İkinci kademe, köşe noktası sayısı az olan (belirgin olmayan, doku özelliği zayıf) bölütlerde, bölüt hareketinin bölge tabanlı bölüt eşleme [21] ile hesaplanmasını içermektedir. İkinci kademe olan bölüt eşleme işlemi yüksek zamansal karmaşıklığa sahip olduğu için birinci kademe ile bazı bölütlerdeki hareket yöneyleri çıkartılmaktadır. Bölüt eşlemede ortalama mutlak fark (MAD) benzeşim ölçütü kullanılmıştır.

$$MAD(u, v, i) = \frac{1}{N_i} \sum_{x, y \in S_i} |I_1(x, y) - I_2(x + u, y + v)| \quad (5)$$

(5)'de S_i i bölütünü, N_i i bölütü içerisindeki piksel sayısını ifade etmektedir. Buna göre, bölütlerin hareket yöneyleri MAD ölçütünün en küçültmesi ile sağlanır:

$$(u'_i, v'_i) = \arg \min_{u, v} (MAD(u, v, i)) \quad (6)$$

Son adımda, aşırı bölütlenmiş alanların hareket verilerini dikkate alacak şekilde çizge tabanlı ve bölgesel bir bölütleme algoritması olan özyinelemeli en kısa ağaç (RSST) ([19], [20]) algoritması ile birleştirilmesi gerçekleştirilmiştir. Çizgesel boyutta, bölütler çizge boğumları (node), bölütler arasındaki komşuluk da çizge bağları (link) ile ifade edilmektedir. Boğum ağırlık yöneyi (V_i) bölütün hareket yöneyi olarak tayin edilmiştir. Bağ ağırlık değeri (7)'de verildiği gibi hesaplanmıştır [20].

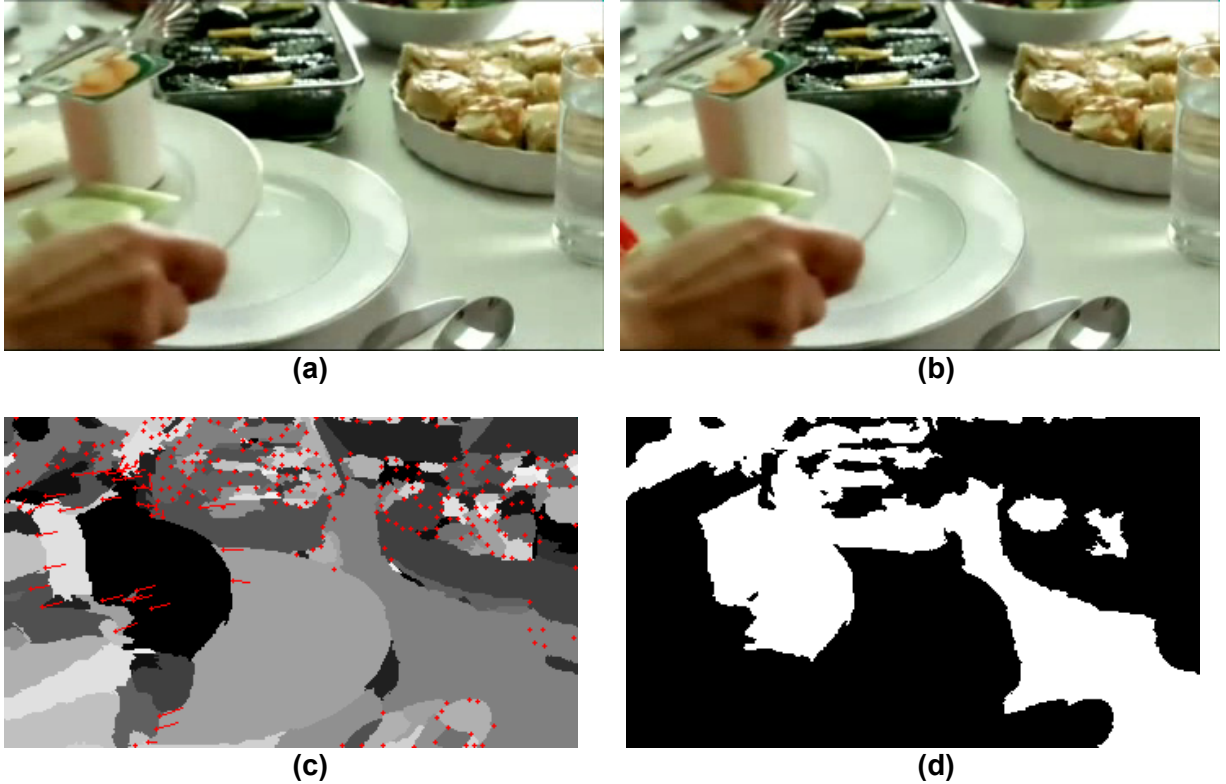
$$LW_i = \left| \bar{V}_i - \bar{V}_j \right| \frac{N_i N_j}{N_i + N_j} \quad (7)$$

Çizge oluşturulduktan sonra çizge bağları, bağ ağırlık değerlerine göre büyükten küçüğe sıralanır ve en düşük bağ ağırlık değerine sahip olan boğumlar (bölütler) birleştirilir. Bölütlerin birleştirilmesinde izlenen çizge boğum ve bağ ağırlık güncellemeleri aşağıda verilmiştir.

$$\bar{V}_k = \frac{\bar{V}_i * N_i + \bar{V}_j * N_j}{N_i + N_j} \quad N_k = N_i + N_j \quad (8)$$

2.3.3.1 Deneysel Sonuçlar

Hareketli nesne bölütlemesinde anlatılan algoritma adımlarının ardışık video karelerine uygulanmasından elde edilen sonuçlar Şekil 29 ve Şekil 30'da verilmiştir. Şekil 30'da hareketli bölge içerisindeki bazı bölütlerin bölüt eşleme sonucunda hatalı hareket yöneyleri çıkardığı görülmektedir. Bunun ana sebebi, hatalı aşırı bölütleme (nesne gölgesi, ön-arka plan renk benzerliği v.b. nedenler ile) sonucunda hareketli nesnelerin bazı parçalarının arka plana ait bölütlerin içerisinde bulunmasıdır. Bu parçalar hareket içerdiği için arka plan bölütleri için sıfırdan farklı hareket yöneyleri elde edilmektedir.



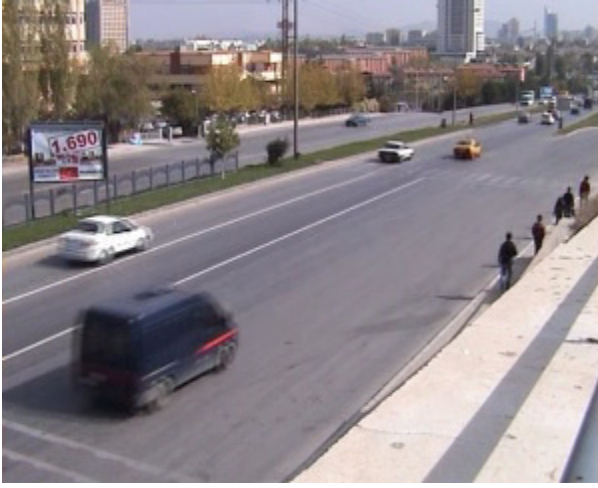


(e)

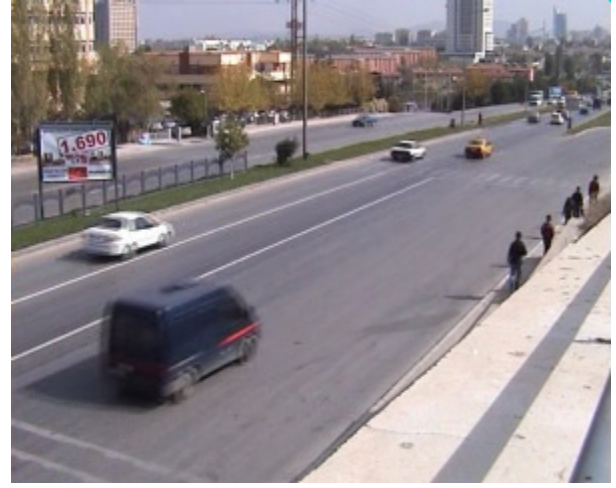


(f)

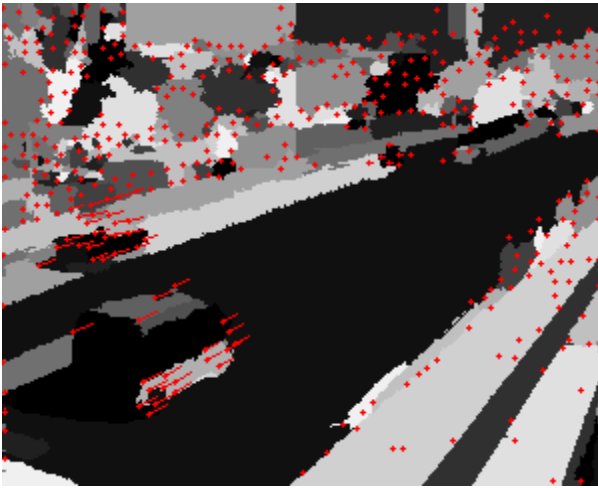
Şekil 29: a), b) Sabit arka plana sahip ardışık video kareleri (sırası ile kare 32 ve 33), c) Aşırı bölütlenme sonucu ve KLT algoritması ile hesaplanmış kırmızı hareket yöneyleri, d) Hareket yöneyleri KLT ile hesaplanmış bölütlerin maskesi, beyaz alanlara dahil olan bölütlerde hareket yöneyleri bölüt eşleme ile değil KLT algoritması ile hesaplanmıştır, e) Bölüt hareket yöneyleri (hareket yöneyleri büyüklükleri ile orantılı olarak 0-255 arasında gösterilmiştir), f) RSST ile bölüt birleştirilmesi sonucu.



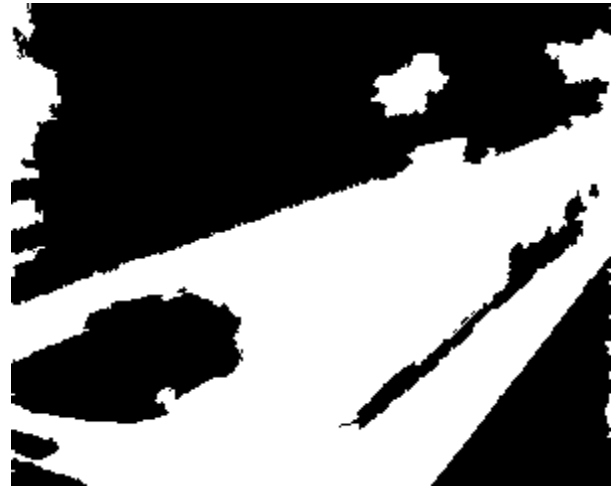
(a)



(b)



(c)



(d)



Şekil 30: a), b) Sabit arka plana sahip ardışık trafik video kareleri (sırası ile kare 29 ve 30), c) Aşırı bölütleme sonucu ve KLT algoritması ile hesaplanmış kırmızı hareket yöneyleri, d) Hareket yöneyleri KLT ile hesaplanmış bölütlerin maskesi, beyaz alanlara dahil olan bölütlerde hareket yöneyleri bölüt eşleme ile değil KLT algoritması ile hesaplanmıştır, e) Bölüt hareket yöneyleri (hareket yöneyleri büyüklükleri ile orantılı olarak 0-255 arasında gösterilmiştir), f) RSST ile bölüt birleştirilmesi sonucu.

2.3.4 Referanslar

- [1] Lertrusdachakul T., Aoki, T., Yasuda, H., "Camera Motion Characterization through Image Feature Analysis", Proc. of the Sixth International Conference on Computational Intelligence and Multimedia Applications, pages 186-190, Aug. 2005
- [2] J.-G. Kim, H.S. Chang, J. Kim, and H.-M. Kim. "Threshold-based camera motion characterization of MPEG video", ETRI Journal, 26(3), pages 269-272, 2004
- [3] R. Wang, and T. Huang, "Fast camera motion analysis in MPEG domain", International Conference on Image Processing, vol. 3, pages 691-694, 1999
- [4] Jinzenji K., Ishibashi S., Kotera H., "Algorithm for automatically producing layered sprites by detecting camera movement", International Conference on Image Processing, vol.1, pages 767 -770, 1997
- [5] Jong-Il Park, Inoue S., Iwadate Y., "Estimating Camera Parameters From Motion Vectors of Digital Video", IEEE Workshop Multimedia Signal Processing, pages 105-110, 1998
- [6] Denzler J., Schless V., Paulus D., Niemann H., "Statistical approach to classification of flow patterns for motion detection", International Conference on Image Processing, vol. 1 pages 517 -520, 1996
- [7] Naito, M. Matsumoto, K. Hoashi, K. Sugaya, F., "Camera Motion Detection using Video Mosaicing", pages 1741-1744, July 2006
- [8] Rong Jin, Yanjun Qi, Alexander Hauptmann, "A Probabilistic Model for Camera Zoom Detection", The Sixteenth Conference on the International Association for Pattern Recognition", vol. 3, pages 859- 862, 2002
- [9] Werner Bailer, Peter Schallauer, Georg Thallinger, "Joanneum Research at TRECVID 2005 – Camera Motion Detection", Institute of Information Systems and Information Management, 2005
- [10] P. Krämer, J. Benois-Pineau, "Camera Motion Detection in the Rough Indexing Paradigm", TREC Video 2005, On-line proceedings, USA
- [11] J. Park, N. Yagi, K. Enami, and E. Kiyoharu, "Estimation of Camera Parameters from image sequence for model-based video coding", IEEE trans. Circuits systems Video Tech. 4(3), pages 288-296, 1994

- [12] P. Bouthemy, M.Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization", IEEE Trans. Circuits Syst. Video Technology, no. 7, vol. 9, pages 1030-1044, Oct. 1999
- [13] M.V. Srinivasan, S. Venkatesh, R. Hosie, "Qualitative estimation of camera motion parameters from video sequences", Pattern Recognition, vol. 30, no. 4, pages 593-606, 1997
- [14] Hyeokman Kim, Tae-Hoon Kwon, Woonkyung M. Kim, Byung-Do Rhee, Samuel M. Song, "Fast algorithm for detection of camera motion", Proceedings of SPIE, Real-Time Imaging III, vol. 3303, pages 78-87, 1998
- [15] C. Harris and M.J. Stephens, "A combined corner and edge detector", Alvey Vision Conference, pages 147-152, 1988
- [16] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker", Microprocessor Research Lab, Intel Corporation
- [17] H. C. Longuet-Higgins, F.R.S., K. Prazdny, "The Interpretation of a Moving Retinal Image", Proceedings of the Royal Society of London, vol. 208 B, pages 385-397, July 1980
- [18] MPEG7 Standard: ISO/IEC/JTC1/ SC29/WG11, "Coding of Moving Pictures and Audio, Camera Motion Descriptor"
- [19] O.J. Morris, M.J. Lee and A. G. Constantinides, "Graph theory for image analysis: an approach base on the shortest spanning tree", Proc. Inst. Elect. Eng., vol. 133, pages 146-152, April 1986
- [20] Ozan Ersoy, "Image Segmentation with Improved Region Modelling" Master of Science Thesis, METU 2004
- [21] Michael Bleyer et.al, "Color-segmentation based coputation of dense optical flow with application to video object segmentation" Österreichische Artificial Intelligence Journal, 24(1), pages 11-15, 2005
- [22] D. Comaniciu, P. Meer, "Mean Shift Analysis and Applications," IEEE Int'l Conf. Comp. Vis., Kerkyra, Greece, pages 1197-1203, 1999.

2.4. İMGE EŞLEME İÇİN İLĞİ NOKTASI ÇIKARMA

İmgelerin anlamsal kategorizasyonu için literatürde global ve lokal olmak üzere iki ayrı yaklaşım uygulanmaktadır. Global yaklaşım imgeden bir bütün olarak alt seviye tanımlayıcılar çıkarılması ve bunlara göre sınıflandırma yapılmasını içerir. Her ne kadar bazı kategoriler bu analiz yöntemiyle ulaşılabilir olsa da (örn. Gökyüzü, orman, deniz, kalabalık v.s.), bu kategoriler için bile imge kalabalık ve heterojen bir sahne içerdiğinde analiz başarısı çok düşmektedir.

Lokal özellikleri kullanan diğer yaklaşımda ise bu tür olası problemleri engellemek ve anlamsal analiz işlemini obje bazlı kategoriler için genişletmek mümkündür. Bu çalışmada lokal imge analizi kullanarak anlamsal sınıflandırma yapabilmek için gerekli alt seviye analiz araçları ve bunların çıktılarının üst seviyede anlamsal sınıflandırma amacıyla nasıl kullanılabileceği anlatılmaktadır.

İmge analizi için lokal özellikleri kullanan ilk çalışmalar Moravec [1] ve Harris [2] İlgili Noktası Detektörleridir. Çok-ölçekli detektörler konusunda ilk örnek ise Blostein ve Ahuja [3] tarafından verilmiştir. Bu çalışmada Laplace filtresinin maksimum verdiği noktalarda bulunan olgulara çok-ölçekli öbekler denilmiştir. Daha sonraları bu çalışma Lindeberg [4] tarafından genişletilmiş ve otomatik ölçek seçimi çatısı altında

öbek kavramı tekrar tanımlanmıştır. Yine öbek detektörü olarak önerilen Hessian-bazlı başka bir yöntem de mevcuttur [5]. Son olarak ilgin dönüşümlere duyarlı bölge detektörleri de yine Lindeberg'in çalışmasında ortaya koyduğu prensiplerden yola çıkarak geliştirilmiştir [6].

İmge analizinde önemli diğer bir konu ise, yukarıda bahsedilen yöntemlerle bulunan ilgi noktalarına bağlı bölgelerin nasıl tanımlanacağıdır. Global yaklaşımda yapılanın tersine burada her bir bölgeden ayrı ayrı alt seviye tanımlayıcılar çıkarılmaktadır. Literatürde çok farklı alt seviye tanımlayıcılar önerilmiştir. Bunlardan en basit ve en yaygın kullanılanı normalize edilmiş karşılıklı ilinti'dir. Bu aslında iki bölgeyi herhangi bir başka tanımlayıcı çıkarmadan, piksel değerleri üzerinden karşılaştırmak için kullanılan bir yöntemdir. Bu yöntem piksel değerlerini direkt olarak kullandığından en küçük değişimler durumunda bile aynı kaynaktan gelen iki bölgeyi eşleyemeyebilir. İşte bu yüzden, çeşitli değişimlere duyarlı birçok tanımlayıcı bu alanda kullanılmıştır. Bunlardan en çok bilinenler görsel özellikleri bölgenin alt bölgelerinde sayılması şeklinde özetlenebilecek olan SIFT [7] ve Şekil Bağlamı [8] tanımlayıcılarıdır. Direkt piksel parlaklığı değerlerini kullanarak histogram oluşturan tanımlayıcılara bir örnek ise 2-D Fırıl İmgesi [9] tanımlayıcısıdır. Bu tanımlayıcı aslında üç boyut verisini tanımlamak üzere önerilen Fırıl İmgesi [10] tanımlayıcısından esinlenilerek iki boyutta gerçekleştirilmiştir. Bunun dışında bölgeyi türevler cinsinden tanımlayan Yönlendirilebilir Filtreler [11], SIFT'in dönmeye karşı dayanıklı versiyonu olan RIFT [9], Gabor Filtrelerine Dalgacıklar gibi çeşitli değişimlere dayalı daha birçok alt seviye tanımlayıcı literatürde mevcuttur.

İmge analizinde son aşama ise bu bölgesel tanımlayıcılarla eşlemesi yapılabilen bölgeleri kullanarak, kategoriler arası sınıflandırmayı sağlayan bir çatının kurulmasıdır. Bunun için de Özellik Torbası [12] isimli en basit olanlardan İşaret Kümesi [13] gibi karmaşık olanlara kadar uzanan çalışmalar bulunmaktadır.

2.4.1 İlgi Noktası Detektörleri

2.4.1.1. Gauss Laplace (LoG):

Bu detektörün en bilinen kullanımı Lindeberg'in Ölçek-Uzay Teorisi [14] içindeki kullanımudur. Bu teori içinde "öbek" normalize edilmiş LoG'un maksimumu olarak tanımlanır. LoG ve Normalize LoG aşağıdaki şekilde verilmiştir.

$$LoG = |G_{xx} + G_{yy}| \longrightarrow ScaleNormalizedLoG = \sigma^2 |G_{xx} + G_{yy}| \quad (1)$$

LoG'un bu teori'de ölçek detektörü olarak seçilmesini anlamak için önce Gauss fonksiyonunun Ölçek-Uzay tanımındaki çekirdek olarak seçilmesini anlamak gerekir. Bu ise, bir ölçek seçici'ye neden ihtiyacımız olduğunun açıklanmasını gerektirir. Bu açıklamayı hemen yapabilmek mümkündür. Objeler farklı ölçeklerde gözlemlenirler. Dolayısıyla ölçeklerarası bir objeyi aramak için ya objenin (ya da parçalarının) ölçeğini objektif olarak seçecek bir detektöre, ya da detaylı analizlerin tümünü her bir olası ölçekte yapmaya ihtiyacımız vardır. İlk seçenek Ölçek-Uzay tanımı sayesinde ilk defa sağlam bir temele oturacak şekilde gerçekleşmiştir.

Ölçek-Uzay Teorisinin kalbinde Gauss Fonksiyonu ve uygulamalı matematik literatüründe önemli yeri olan ve ısı yayılımı ile ilgili olan *Yayınım Denklemi* vardır. Bu denklemde belli bir sıcaklığa sahip bir noktadan belli uzaklıktaki bir noktada belli bir süre sonra sıcaklığın ne olacağı modellenmektedir. İmge analizi alanına geçebilmek için ise bu denklemdeki sıcaklık yerine ölçek, iki nokta arasındaki mesafe ise iki boyutlu imge uzayında koordinatlar ile yer değiştirmesi gerekir. Bu denklem aşağıda verilmektedir:

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

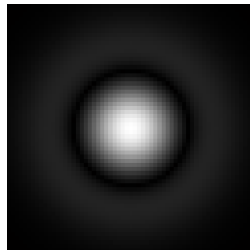
İmgelerin değişik ölçekteki hallerini elde etmek için ise bir düzleştirici fonksiyon kullanmak gerekir. Bu fonksiyonun sahip olması gereken birçok özellik yanında en önemlisi nedensellik ilkesini bozmamasıdır. Bu alanda Nedensellik İlkesi “imgenin ifade edildiği ölçek arttırıldıkça yeni görsel yapıların oluşmaması” anlamına gelmektedir. (2) numaralı denklemi tek boyutta ifade edersek, ölçekleri ifade eden çekirdek fonksiyon için nedensellik ilkesini nasıl garanti altına aldığını görebiliriz.

$$\partial_t L = \frac{1}{2} \partial_{xx} L \quad (3)$$

Yukarıdaki denklem ortaya iki şart koyar:

1. Bir fonksiyon için ikinci türev pozitifse, bir minimum yakınında bulunduğu anlamına gelir. Bu da Ölçek-Uzay’da ölçek parametresi arttıkça minimum’un değerinin artacağı anlamına gelir.
2. Benzer şekilde, eğer ikinci türev negatif ise, bir maksimum noktası yakınında bulunduğu anlamına gelir. Bu durumda da Ölçek-Uzay’da ölçek parametresi arttıkça maksimum değeri düşer.

Bu kuramsal temellere dayanan LoG detektörü imgenin farklı ölçekteki hallerine uygulanır ve bu ölçeklerin herbirinden lokal maksimumlar bulunur. Bu lokal maksimumlar o ölçeklerdeki öbek benzeri yapıların merkez noktalarını belirler. Bu bölgelerin büyüklüğü ise buldukları ölçeğe göre belirlenir. Bu detector öbek benzeri yapılar için şekilsel olarak bir eşleştirilmiş filtreyi andırır. Aynı zamanda insanın görsel sisteminin de ölçek seçim kısmının güzel bir modeli olduğu belirtilen bu filtrenin görüntüsü aşağıdaki şekildedir.



Şekil 31: LoG Fonksiyonu

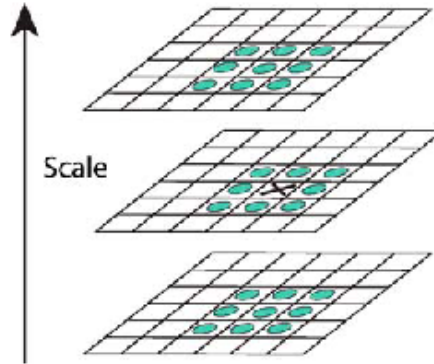
2.4.1.2. Gauss Farkı (DoG)

Bu detektör Normalize LoG detektörünün çok yakın bir kopyasıdır. Isı Yayılmı Denkleminin çözümü olan Gauss fonksiyonu sayesinde ölçeğe göre türev alma işlemini (2), aşağıdaki şekilde sonlu fark ile yaklaşımlamak mümkündür.

$$\frac{\partial G}{\partial \sigma} = (2\sigma) \frac{\partial G}{\partial t} \longrightarrow \partial_{\sigma} G = (2\sigma) \frac{1}{2} \nabla^2 G = \sigma \cdot \nabla^2 G \quad (4)$$

$$\sigma \cdot \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \longrightarrow G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G \quad (5)$$

Buradaki denklemde k ardışık iki ölçekte sabit olduğu sürece DoG yaklaşımı LoG'a uygundur. DoG ile oluşturulan Ölçek-Uzay'da ise ilgi noktalarının yerini bulmak için aşağıdaki şekilde gösterilen komşuluk içinde uç değerler aranır.



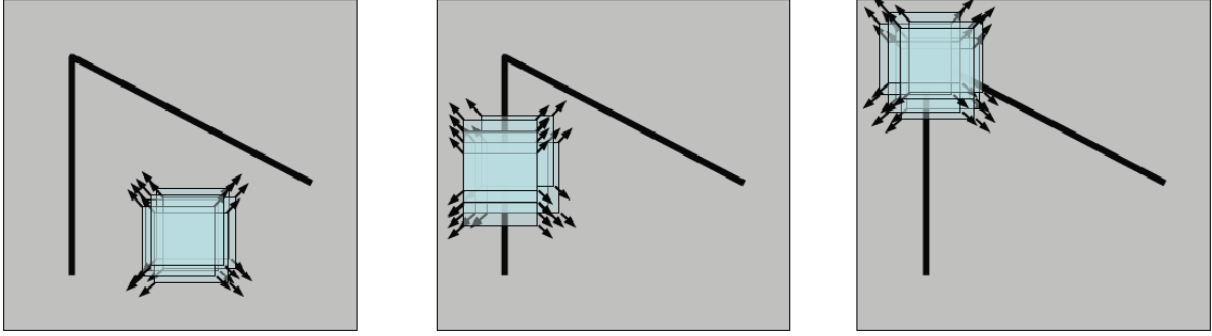
Şekil 32: DoG Uç Noktalarının arandığı komşuluk.

2.4.1.3. Harris-Laplace

Harris köşe detektörü [2] isminden de anlaşılacağı gibi köşe noktalarının bulunmasında kullanılan bir detektördür. Bu detektörün mantığı ikinci moment matrisi ya da diğer adıyla özilinti matrisi üzerine kuruludur. Özilinti ismi ise aynı isimli fonksiyonun tanımından gelmektedir. Bu fonksiyon sinyal üzerinde belli bir merkez etrafında çeşitli yönlerde hareket edildiğinde merkezle olan ilintinin ne şekilde değiştiğini ölçer. Bu fonksiyonun ayrık bir versiyonu Moravec tarafından tanımlanmıştır[1].

$$c(x, y) = \sum_w [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \quad (6)$$

Bu fonksiyonun anlamını ve altında yatan mantığı anlamak için aşağıdaki imge üzerinde verdiği tepkileri incelemek gereklidir.



Şekil 33: Oto-korelasyon fonksiyonuna göre farklı sonuç veren değişik tipteki imge noktaları.

(6)'ıncı denklem bazı düzenlemeler ve sadeleştirmeler sonrası şu şekilde gelir:

$$c(x, y) = [\Delta x \quad \Delta y] M(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (7)$$

Bu denklemdeki M matrisi oto-korelasyon ya da ikinci moment matrisidir. Bu matrisin eigen değerleri bize yukarıdaki resimdeki nokta tiplerinden hangisinde bulunduğunu verir. Eğer eigen değerlerinin ikisi de küçükse bu en soldaki nokta tipinde bir noktada olduğumuz anlamına gelir. Bir eigen değeri yüksek diğeri düşük ise ortadaki türde, iki eigen değeri de yüksekse en sağdaki türde bir köşe noktasında bulunduğu düşünülebilir. Bu son tipteki köşe noktaları Harris Detektörünün bulmak istediği noktalar olup aşağıdaki fonksiyonla bir noktanın ne kadar köşe olduğuna bakılabilir.

$$köşelik = \det(M) - \alpha \cdot \text{trace}^2 M \quad (8)$$

Bu detektör her bir ölçekte ayrı ayrı uygulanıp ilgi noktaları bulunur. Sonra bulunan bu Harris noktalarının ölçeği LoG ile tekrar rafine edilerek Harris-Laplas ya da çok ölçekli Harris ilgi noktaları elde edilmiş olur [15].

2.4.1.4. Hessian-Laplace

Hessian Matrisi temel alınarak geliştirilen [15] bu detektör aslında LoG'un biraz daha seçici versiyonu olarak da düşünülebilir. Bu matris ve determinant'ı aşağıda verilmiştir:

$$Hessian = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \quad (9)$$

$$Det = I_{xx}I_{yy} - I_{xy}^2 \quad (10)$$

Yukarıdaki Hessian Determinant'ın LoG'dan daha seçici olduğu nokta uzatılmış bölgelerdir. Bu bölgelerde ikinci türevlerden biri çok küçük olur ve bunların çarpımına bakan Hessian Determinant bundan çok etkilenir. Oysa LoG için ölçüt ikinci türevlerin toplamıdır ve ikinci türevlerden biri düşük değerlere sahip olsa da LoG fonksiyonu yüksek sonuç vermeye devam edebilir. Bu detektörün çok ölçekli halinde ise Hessian

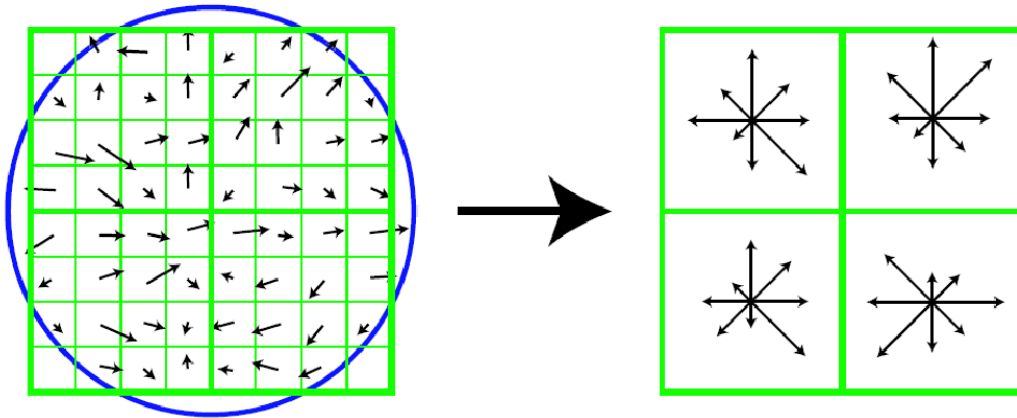
herbir ölçekte ayrı ayrı uygulanır ve sonra bulunan ilgi noktaları LoG'a göre ölçeklendirilir. Bu işlem Hessian durumunda aslında aynı anda Hessian'ın hem determinant'ının hem de trace'inin lokal uç noktasında bulunan ilgi noktalarının seçilmesi anlamına gelir, çünkü LoG ile Hessian trace aynı tanıma sahiptir.

2.4.2. İlgin Bağımsız İlgi Noktası Uyumlandırması

Bu işlem ilgi noktalarının imge içindeki koordinatları ve ölçeklerinin yanısıra, buldukları bölgenin şeklinin de değişmesi durumunda kullanılmak üzere geliştirilmiştir. Bu işleme konu olan ve bölgenin şeklinin daireden elips'e ne kadar kaydığını gösteren ölçüt daha önce karşılaştığımız ikinci moment matrisidir. Bu uyumlandırma sonucu orjinal imgedeki eliptik bölge önce normalize edilir ve bir çembere dönüştürülür. Sonra alt seviye tanımlayıcılar çıkartılır ve eşleştirilir.

2.4.3. Ölçek Bağımsız Öznitelik Dönüşümü (SIFT) Bölgesel Tanımlayıcısı

İlgi noktalarını birbiriyle eşleştirmek üzere alt seviyede tanımlamak için Ölçek Bağımsız Öznitelik Dönüşümü (SIFT) [7] kullanılmıştır. Bu tanımlayıcı 4x4'lük bir nicemlenmiş ızgara üzerinde 8 nicemlenmiş yönde hesaplanan gradyan histogramından oluşur. Bu histogram bölgenin dönmesine karşı dayanıklı değildir ama karşılaştırma sırasında normalize edilebildiği için aynı bölgenin farklı şekilde döndürülmüş versiyonlarını eşleştirmek mümkündür.



Şekil 34: SIFT Tanımlayıcısı

2.4.5. Referanslar:

- [1] H. Moravec, "Towards Automatic Visual Obstacle Avoidance," *Int'l Joint Conference on Artificial Intelligence*, page 584, 1977.
- [2] C Harris, M Stephens, "A combined corner and edge detector," *Alvey Vision Conference*, 1988.
- [3] D. Blostein and N. Ahuja, "A multiscale region detector," *Computer Vision, Graphics and Image Processing*, Vol 45, pp. 22-41, 1989.

- [4] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *Int'l Journal of Computer Vision*, Vol. 30(2), pp. 77-116, 1998.
- [5] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors," *Int'l J. Computer Vision*, Vol. 1, pp. 63-86, 2004.
- [6] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," *Proc. ECCV*, Vol. 235, p. 128, 2002.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. Computer Vision*, Volume 60, Number 2, pp. 91-110, Nov. 2004.
- [8] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *Trans. On Pattern Analysis and Machine Intelligence*, Vol. 2, pp. 509-522, Apr. 2002.
- [9] Svetlana Lazebnik, Cordelia Schmid, Jean Ponce, "A Sparse Texture Representation Using Affine-Invariant Regions," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, pp.319, Volume 2, 2003
- [10] A. E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, pp.433-449, 1999.
- [11] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 27, pp. 1615-1630, Oct. 2005.
- [12] L. Fei-Fei, R. Fergus, A. Torralba, "Recognizing and Learning Object Categories," *ICCV Course*, Oct. 2005.
- [13] P. Moreels and P. Perona, "Common-Frame Model for Object Recognition," in *Proc. NIPS*, 2004.
- [14] Tony Lindeberg, "Scale-Space Theory in Computer Vision," Dordrecht: Kluwer Academic Publishers, 1994.
- [15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A Comparison of Affine Region Detectors," *Int'l J. Computer Vision*, Vol. 65, pp. 43-72, Jan. 2005.
- [16] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Proc. CVPR 2006*, Vol. 2, pp. 2169-2178.

3. BÖLÜTLEME

Bölütleme problemi anlamsal bilgilere ulaşmadaki en önemli ara basamak olarak sayılabilir. Bir veriyi uzamda veya zamanda homojen parçalara ayırmakla anlamsal bilgiyi çıkarabilmek daha mümkün hale gelmiş olmaktadır. Bu problemdeki temel zorluk, bölütlemenin de kimi zaman anlamsal üst seviye bilgilere ihtiyaç duyması (örn. sahnedeki nesne sayısı) ve buna bağlı algoritmaların performanslarının istenilen düzeye ulaşmamasıdır. Bölütlemenin başarıyla yapılması durumunda, anlamsal bilgilere ulaşmak, verinin homojen hale getirilip daha modellenabilir olması nedeniyle, mümkün olabilmektedir.

Bu proje kapsamında 3 farklı bölütleme problemine özgün çözümler sunulmuştur. Bu problemlerden ilki imge bölütlemeye yöneliktir. Bu problem için önerilen yaklaşım, çizge kuramsal popüler bir algoritmanın işlemsel olarak oldukça hızlandırılmasına rağmen bölütleme performansının çok az etkilenmesinin sağlanmasıdır. Hareketli nesne bölütleme problemi için literatürdeki tarihsel gelişim takip edilerek, rasgele şekilli (göreceli küçük) bölgeler için parametrik hareket modellerinin bulunması ve benzer harekete sahip komşu bölgelerin biraraya getirilmesi yaklaşımı bensenmiştir. Ancak, bu yaklaşım harekete bağlı açılma ve kapanmaları da dikkate alan bir formülasyon ile zenginleştirilmiştir. Video çekimlerini kamera sınırlarını bölme amacıyla yapılan çalışmalar ise çizge kuramında yer alan domine setler kavramı üzerine oturtulmuş ve ortaya özgün bir çalışma konmuştur.

3.1. ÇİZGE KURAMSAL İMGE BÖLÜTLEME

Bilgisayarla görü alanındaki en zorlu problemlerden biri, görüntü anlamanın en önemli araçlarından biri olan görüntü bölütlemesidir. Her ne kadar birçok çözüm yolu önerilse de, sahip olduğu karmaşık ve öznel değer kümesine bağlı olarak problem henüz tam olarak çözülmemiştir. Literatürde, araştırmancının büyük bir kısmı yerel özelliklere dayalı çözümler üzerinde odaklanmıştır [1][1][2]. Genel olarak çok etkin olmalarına rağmen, bölütleme sırasında global özellikler kaybolabilmektedir. Diğer taraftan, son on yılda, çizge kesiklerini global eniyileme yöntemli olarak kullanan çizge tabanlı bölütleme yöntemleri popülerlik kazanmıştır [3]-[9].

Çizge tabanlı yaklaşım, bölgesel yöntemlerin tersine tepeden aşağı bölütleme düzeni ortaya koymaktadır [3]. Görüntü, "büyük bir resim" olarak düşünülmekte ve boğumları (V) pikseller olan, bağları (E) ise piksellerin parlaklık benzerliğine bağlı olarak oluşturulan bir ağırlıklandırılmış çizge (G) üzerine oturtulur. Çizge üzerinden bölütleme, kesik değerlerinin çeşitli özel matrislerin özdeğer vektörlerinin ayrışımı ile eniyilenmesi ve bu işlemin döngüsel bir şekilde çizgenin parçalanması için kullanılması sonucunda sağlanır, [3][4][6][8]. [9]'da ise farklı olarak bölütleme içiçe kesikler sayesinde elde edilir. K -yol kesik şekline dayalı olarak elde edilen farklı bir stokastik bölütleme yöntemi de [7]'de verilmektedir.

Global yöntemlerde, gruplama insan algısına benzer bir biçimde (bütünden ayrıntıya doğru) yapılmaktadır. Her ne kadar global yöntemler böyle gerçekçi bir varsayımı kullansalar da, yürütümleri görüntünün boyutu arttıkça zaman alıcı olabilir. Hesaplama problemi genellikle görüntülerin indirgenmesi [3] ile ya da çoklu ölçek yaklaşımı [6] kullanılarak çözülmeye çalışılmıştır. Önerilen yöntem, düzgeli kesik görüntü bölütleme yönteminin [3] işlem zamanını azaltmak ve görsel ayrıntıları korurken performansını arttırmak amacına dayalıdır. Bu amaçla, çizge yapısındaki boğumlar piksel yerine aşırı bölütlenmiş bölgeler tarafından

oluşturulur. Böylece, çizge karmaşıklığı da azaltılmış olur. Fakat, bölgelerle oluşturulan çizge düzensiz bir boğum dağılımının oluşmasına sebep olur. Düzensiz boğum dağılımı ise, daha çok sayıda bağa sahip olan boğumların birleştirilmesine yönelik bir yanlılığa yol açar. Bu çalışmada aynı zamanda bölütleme kalitesini düşüren bu yanlılığın ortadan kaldırılması için bir çözüm önerisi sunulmuştur. Bildirinin 2'inci bölümünde düzgeli kesik görüntü bölütleme yöntemi gözden geçirilmektedir. Önerilen algoritmanın ayrıntıları 3'üncü bölümde anlatılmaktadır. Deneysel sonuçların 4'üncü bölümde sunulmasından sonra, son kısım özetleyici görüşlere ayrılmıştır.

3.1.1. Düzgeli Kesik Yöntemine Genel Bakış

Düzgeli kesik görüntü bölütleme yöntemi (DKGB) [3], tüm görüntüden başlayarak aşağıya doğru parçalama ile global bir çizge tabanlı yöntemdir. DKGB'de iki ana aşama vardır, *çizge oluşumu* ve *döngüsel bölme*. Düzgeli kesik yönteminin tepeden aşağı özelliği, görüntünün pikseller arası ilişkilerin tutulduğu bir çizgeye dönüştürülmesi ile sağlanır. Çizge, yönsüz ve ağırlıklandırılmış bir çizge olup boğumlar piksellere karşılık gelmekte, boğumlar arası bağ enerjileri Denklem (1)'de verilen ceza fonksiyonu ile hesaplanmaktadır.

$$w_{i,j} = \begin{cases} e^{-|I(i)-I(j)|^2/\sigma_I - |X(i)-X(j)|^2/\sigma_X} & |X(i) - X(j)|^2 < R \\ 0 & \text{diğer yerlerde} \end{cases} \quad (1)$$

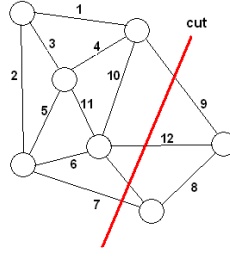
Denklem (1)'de, I görüntüye, $I(i)$ i nci piksele karşılık gelen parlaklık değerine, X ise bu piksellerin konumuna karşılık gelmektedir. Çizgede, belirli bir R yarıçapı içerisine düşen pikseller birbirleriyle ilişkilendirilmektedirler. Bu yüzden çizge kısmi bir çizgedir ve bağ enerjileri boğumlar arası benzerliğin $[0,1]$ arasında temsil edildiği bir fonksiyonun çıktısıdır. Bölütleme, çizgenin bağlar arası kesiklerle daha küçük çizgelere döngüsel bir şekilde bölünmesi sonucu elde edilir. Her aşamada, bir V çizgesi iki alt çizgeye (A ve B), $V=A \cup B$, $A \cap B = \emptyset$ koşulları ile bölünmektedir. Bölünmeyi boğumlar arası bağları koparan kesikler sağlar. Bir kesik genellikle birden fazla bağ boyunca yer alır ve dolayısıyla kopan boğum sayısı birden fazladır, Şekil 35. Bunun yanında, her kesik kopan bağ enerjilerinin toplamı ile elde edilen bir ceza değerine sahiptir, Denklem (2).

$$Cut(A, B) = \sum_{i \in A, j \in B} w(i, j) \quad (2)$$

DKGB yöntemi, döngülü parçalama sırasında düzgeli kesik ölçümünü (3) eniyilemeyi amaçlamaktadır.

$$Ncut(A, B) = \frac{Cut(A, B)}{TotalW(A, V)} + \frac{Cut(A, B)}{TotalW(B, V)} \quad (3)$$

Denklem (3)'te, $TotalW(A, V)$ A grubundaki boğumlar ile tüm çizgedeki boğumlar V arasındaki toplam bağ enerjisine karşılık gelmektedir. Düzgeli kesik eniyilemesi tam olarak np-bütünlükte bir problemdir; fakat değişken değişikliği ve problemin reel değerli alanda formüle edilmesi ile ayrık değerli yaklaşık bir çözüm elde edilebilir [3]. Yeni formülasyon Denklem (4)'te verilmektedir.



Şekil 35: Çizgede örnek bir kesik.

$$Ncut = \frac{y^T (D - W)y}{y^T Dy} \quad (4)$$

Denklem (4)'te D , N boğumlu bir çizgeye ait her boğuma bağlı toplam bağ enerjisini belirten $N \times N$ diyagonal bir matrise karşılık gelmektedir. W , boğumlar arası benzerliklerin gösterildiği bağ enerjilerini tutan simetrik $N \times N$ benzerlik matrisini belirtmektedir. Son olarak, y boğumların belirli bir alt çizgeye ait olma derecesini gösteren $N \times 1$ bir matrisi temsil etmektedir. Denklem (4)'teki formülasyonun eniyilenmesi aşağıdaki genellenmiş öznelik değer sistemi ile yaklaşık olarak elde edilebilmektedir [3].

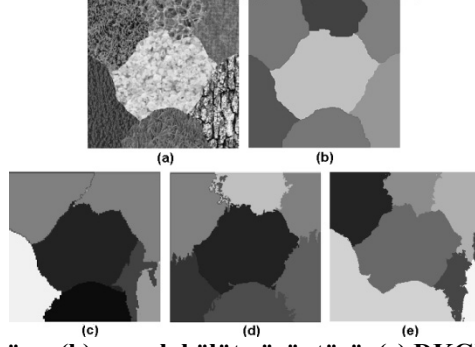
$$D^{-1/2} (D - W) D^{-1/2} y = \lambda y \quad (5)$$

Sözkonusu çizgenin parçalara ayrılması, öznelik değeri ayrıştırılması ve en küçük ikinci öznelik değerine karşılık gelen öznelik vektörü sayesinde olmaktadır. [3]'te bölütlemeyi sağlamak amacıyla bu kriter, her alt çizgenin parçalanması için kullanılmaktadır.

3.1.2. Deneyler

Önerilen yöntemin performansını ölçmek amacıyla farklı görüntüler üzerinde deneyler yapılmıştır. Bu deneylerde, geliştirilen düzgeli kesik yöntemi, DKGB algoritması [3] ve çoklu ölçekli düzgeli kesik yöntemi [6] ile bölütleme kalitesi ve işlem süresi açısından karşılaştırılmıştır. Bölütleme kalitesi, elde edilen bölüt ile doğru bölüt arasındaki ortalama karesel hata ölçüt alınarak hesaplanmaktadır. 3, 4, 5'inci şekillerde farklı görüntülere ait bölütlenme sonuçları gösterilmektedir. Özel karşılaştırma temel alındığı zaman önerilen yöntem diğer yöntemlere göre gerçek bölütlere daha benzer sonuçlar vermektedir. Algoritmaların MSE çizimleri Şekil 40'da verilmektedir. Önerilen yöntem *Mozaik* ve *Nesneler* görüntüleri için en az hatayı vermektedir. Bunun yanında, DKGB yöntemi *İnek* görüntüsü için en az hatayı vermektedir. Tablo 2'de, algoritmaların benzerlik matrisi oluşturulmasından sonra yer alan öznelik ayrışımı sırasındaki yürütüm zamanları verilmiştir. Deneyler, 3 GHz CPU (4) Pentium bilgisayarında MATLAB kodları ile yapılmıştır. Tablo 2'ye göre, önerilen yöntem ile yürütüm zamanı hızlı bir şekilde düşmekte, bunun yanında bölütleme kalitesi hem nesnel hem de öznel kriterlere göre artmaktadır.

Son olarak, geliştirilmiş çizge yapısındaki boğumların bağ sayısının kısıtlanmasının sonuçları Şekil 41'de gösterilmektedir. Görüntü, düzensiz bağ ve boğum dağılımından dolayı hatalı bir şekilde bölütlenmekte fakat bağ sınırlandırması bölütlenmeyi düzeltmektedir.



Şekil 36: (a) Mozaik görüntüsü ve (b) gerçek bölüt görüntüsü, (c) DKGB, (d) önerilen yöntem, (e) Çok ölçekli düzgülü kesik algoritması.

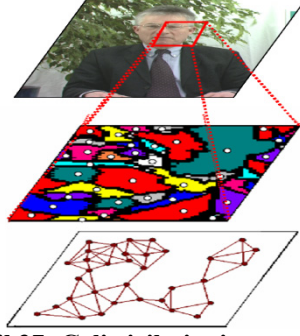
3.1.3. Önerilen bölütleme Yöntemi

DKGB algoritmasında, verilen 200x200 piksellik bir görüntü için elde edilen çizge, 40,000 boğum ve 40,000x40,000 boyutlarında bir benzerlik matrisi ile temsil edilmektedir. Böyle bir matrisin öznitelik değer çözümlenmesi açıkça zaman alıcıdır. Bunun yanında, görüntünün boyutları arttıkça algoritmanın yürütüm süresi daha da artacak ve kullanılacak hafıza ihtiyacı bazı problemlere yol açabilecektir. Görüntü indirgenmesi, hız ve hafıza problemlerine bir çözüm olarak sunulabilmektedir [3]. Ne yazık ki, görüntü indirgenmesi yerel parlaklık bilgisinin kaybolmasına yol açmakta ve bölütleme sonucunda nesne sınırları bozulabilmektedir. Bu çalışmada, karmaşıklık probleminin herhangi bir yerel bilgi kaybolmadan çözümü amaçlanmaktadır. Öncelikle; çizge, piksel yerine aşırı bölütleme sonucunda (yerel bölütleme) elde edilmiş bölgeler ile oluşturulmaktadır. Bu sayede, boğum sayısı epey azaltılmış olup aynı zamanda yerel özellikler korunmuş olur. Aşırı bölütleme, görüntünün anlamsal özelliği olmayan fakat renk açısından türdeş olan bölgelerin iyi bir şekilde tanımlandığı, göreceli olarak daha fazla bölütün elde edildiği bölütleme işlemi olarak kullanılmaktadır. Yeni çizge yapısı (kırmızı çerçeve için) Şekil 37'de bir örnek ile gösterilmiştir. Rasgele renklendirilmiş bölgeler aşırı bölütler olup, bu bölütlerin merkezleri çizgede boğum olarak tanımlanmıştır.

Çizgedeki her bir boğum küçük bir piksel grubunu S_i temsil etmekte olup bu bölgenin ortalama parlaklığı S_i (I^i) boğumun karakteristik özelliği olarak kullanılmaktadır. Bu aşamada, boğum karakteristiği olarak moment ve histogram gibi daha yüksek dereceli açıklayıcılar da kullanılabilir. Boğumlar arası bağ enerjileri, bölütlerin konum mesafeleri dikkate alınmaksızın aşağıdaki fonksiyon aracılığıyla hesaplanır.

$$w_{i,j} = \begin{cases} e^{-|I^i - I^j|^2 / \sigma_i} & |X(i) - X(j)|^2 < R \\ 0 & \text{diğer yerlerde} \end{cases} \quad (6)$$

Çizge yapısının yeni konfigürasyonu genellikle, boğumların düzensiz bir şekilde dağılım göstermelerine yol açar. Düzensiz dağılım, Şekil 37'de de açıkça gösterildiği gibi bölüt büyüklüklerinin değişken olmasından kaynaklanmaktadır.



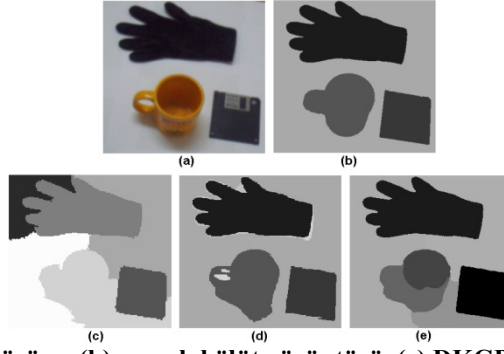
Şekil 37: Geliştirilmiş çizge yapısı

S_i 'lerin düzensiz dağılımı, bağ enerjilerini dikkate almadan daha çok bağ sayısına sahip boğumları birbirine bağlanması yönünde bir eğilime yol açmaktadır. Bu eğilim, DKGB algoritmasının formülasyonuna dayanmaktadır; eniyileme problemi analiz edildiği zaman $TotalW(A,V)$ normalleştirme parametresi her bir boğumun toplam bağ enerjilerine bağlı olduğu görülmektedir. Çizgenin, piksellerle modellendiği koşulda her boğumun komşu sayısı, dolayısıyla sahip olduğu bağ sayısı, aynı olduğu için bölütleme sürecinde bağ enerjileri eşit oranda ağırlığa sahiptirler. Düzensiz bir çizgede ise, farklı bağ sayılarından dolayı, bağ sayısı yüksek olan her boğumun toplam bağ enerjisi de her ne kadar düşük bağ enerjilerine sahip olsa da, toplama bağlı olarak yüksek olacaktır. Bu fenomenden dolayı, bağ sayısı çok olan boğumlardaki düşük enerjili bağlar; bağ sayısı az olan boğumlardaki yüksek enerjili bağlara üstünlük kazanmaktadır. Bunun yanında, böyle bir eşitsizlik öznel vektörlerinin çözümlenmesindeki dengesizliklere sebep olmaktadır. Bu dengesizlikler de, her ne kadar düzgeli kesik değeri eniyilense de, bölütlemeye hatalara yol açmaktadır. Düzensizlik ve bağ sayılarındaki farklılıklar düşünüldüğü zaman, eniyilenecek düzgeli kesik değerinin bölünecek gruplardaki toplam bağ sayısı ile de normalleştirilmesi de gerekmektedir, Denklem (7). Fakat, bu yeni formülasyonun, değişken değiştirme yöntemi ile açık halde bir çözümü sağlanamamaktadır. Dolayısıyla, boğumların aynı sayıda bağa sahip olmalarının zorlanmasıyla yaklaşık bir çözüm elde edilmektedir. Fakat, çizgenin yönsüz olması ve benzerlik matrisinin simetrik olmasından dolayı böyle bir çözümü pratikte uygulamak çok kolay değildir. Bağ sayıları eşitliği, her boğumun bağ sayısını belirli bir değerle kısıtlanması sayesinde yaklaşık olarak sağlanabilir. Böyle bir kısıtlama, bağ sayısının değişikliğini önemli bir miktarda azaltacaktır. Dolayısıyla, düşük enerjili bağların yüksek enerjili bağlara olabilecek üstünlüğü kısıtlanmaktadır. Kısıtlama, her ne kadar boğumların bağ sayılarını eşitleyemese de deneyler düzensizliğin büyük bir ölçüde azaltıldığını göstermektedir.

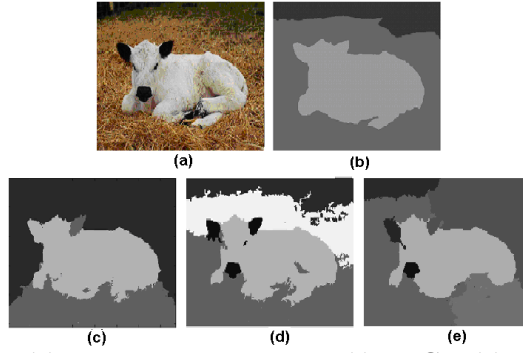
$$\begin{aligned}
 Ncut(A, B) &= \frac{Cut(A,B)}{TotalW(A,V)/N_A} + \frac{Cut(A,B)}{TotalW(B,V)/N_B} \\
 &= \frac{N_A \cdot Cut(A,B)}{TotalW(A,V)} + \frac{N_B \cdot Cut(A,B)}{TotalW(B,V)}
 \end{aligned} \tag{7}$$

3.1.4. Sonuçlar

Önerilen yöntemin sağladığı performans artışı deneyler aracılığıyla açık bir şekilde görülmektedir. Çizge yapısında piksel yerine, bölütlerin kullanımı hem çizgenin karmaşıklığının azaltmakta hem de yerel anlamda benzer renkli piksellerin birleştirilmesi açısından iyi bir başlangıç sağlamaktadır. Bu sayede, yerel bölgeler herhangi bir bilgi kaybı olmadan modellenmekte ve hesaplama karmaşıklığı büyük bir oranda azaltılmaktadır. Sonuç olarak, önerilen yöntem iyi bilinen DKGB algoritması üzerinde bariz bir üstünlük sağlamaktadır.



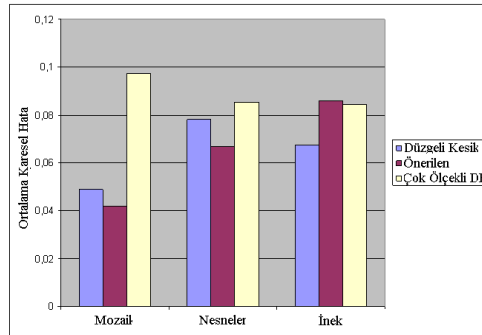
Şekil 38: (a) Nesneler görüntüsü ve (b) gerçek bölüt görüntüsü, (c) DKGB, (d) önerilen yöntem, (e) Çok ölçekli düzgülü kesik algoritması.



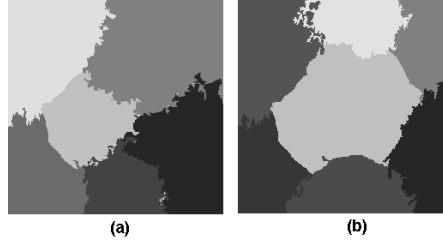
Şekil 39: (a) İnek görüntüsü ve (b) gerçek bölüt görüntüsü, (c) DKGB, (d) önerilen yöntem, (e) Çok ölçekli düzgülü kesik algoritması.

Tablo 2: Her üç yöntemin de öznelilik değeri ayrışımı sırasındaki yürütüm zamanları

Süre (saniye)	<i>Mozaik</i>	<i>Nesneler</i>	<i>İnek</i>
DKGB	26,4	24,4	20,8
Önerilen	2,05	1,41	2,24
Çok ölçekli	8,14	12	8



Şekil 40: Bölütleme yöntemlere bağlı olarak elde edilen MSE değerleri.



Şekil 41: Mozaik görüntüsünün bölütlenme sonuçları (a) sınırlandırılmamış bağ ve (b) sınırlandırılmış bağ durumu için.

3.1.5. Referanslar

- [1] K S. Fu and J.K. Mui, "A Survey on Image Segmentation" *Pattern Recognition, Vol. 13, pp. 3-16, 1981.*
- [2] Ozan Ersoy, "Image Segmentation with Improved Region Modelling" *Master of Science Thesis, METU 2004.*
- [3] C. Dorin, M.Peter , "Mean Shift : A robust approach toward feature space analysis", *IEEE Transactions on pattern analysis and machine intelligence vol:24,no:5, May 2002.*
- [4] J.Shi and J.Malik "Normalized cuts and image segmentation" *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun 1997.*
- [5] M.Stoer and F.Wagner, "A simple minimum cut" *Algorithms –ESA'94 pages 141-147, 1994.*
- [6] S.Sarkar and P.Soundararajan,"Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata", *IEEE Transactions on Pattern Analysis and Machine Intelligence,22(5):504- 525, May 2000.*
- [7] Timothee Cour, Florence Benezit, Jianbo Shi, "Spectral Segmenatation with Multiscale Graph Decompositon" *IEEE International Conference on Computer Vision and Pattern Recognition, 2005.*
- [8] Y. Gdalyahu, D. Weinshall, and M. Werman. "Stochastic Image Segmentation by Typical Cuts", *In IEEE Conference on Computer Vision and Pattern Recognition, pages 596–601*
- [9] S.Wang and J.M.Siskind, "Image Segmentation with Ratio Cut ", *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 25, No 6, June 2003*
- [10] O. Veksler, "Image Segmentation by Nested Cuts," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pp. 339-344, 2000.*

3.2. KATMANSAL HAREKETLİ NESNE BÖLÜTLEMESİ

Videonun bileşen nesnelere ayrıştırılması, bilgisayarla görme ve video işleme alanlarındaki en zor problemlerden biridir. Bu şekildeki bir bölütleme, uzam-zamansal verinin doğal ve özlü bir ifadesi olduğu için, bu konuda giderek artan yoğunlukta çalışmalar yapılmaktadır. Videonun nesne-tabanlı ifadesinin, video kodlama, çerçeve hızı arttırımı, 2.5B sahne ifadesi gibi çok çeşitli uygulamaları mevcuttur.

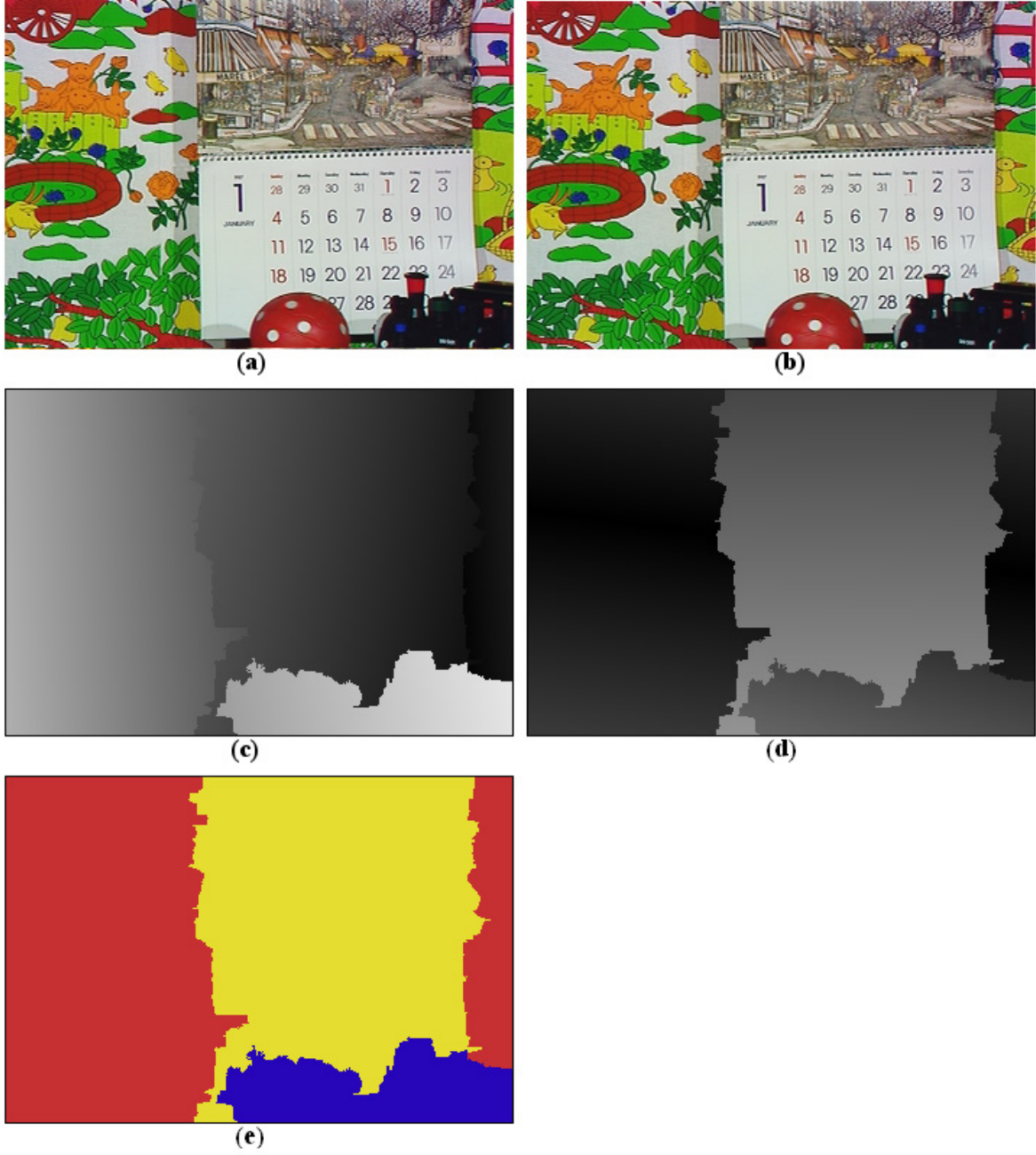
Video nesne bölütleme, özetle, $2B+t$ uzam-zamansal alanda guruplama olarak ele alınabilir. Bu açıdan, video nesne bölütleme algoritmaları, uzamsal ve zamansal bölütleme işlemlerinin uygulanma sıralarını dikkate alarak, üç ana sınıfa ayrılabilir [1]: uzamsal bölütlemeye öncelik verenler, zamansal yörüngeleri guruplayanlar ve doğrudan uzam-zamansal alanda çalışanlar.

İlk sınıfa dahil olan metodlar, çerçeve tabanlı bölütlemeyi takiben zamansal yayılma fikrine dayalıdır. Bu metodlar, renk, doku ve hareket tabanlı uzamsal bölütlemeye öncelik verdikleri için görüntü bölütleme yaklaşımlarının bir devamı niteliğindedir.

Zamansal guruplama ile başlayan metodlar, ilk önce görüntülerde göze çarpan öznitelik noktalarını bulur ve daha sonra da çerçeveler boyunca bu noktaları takip eder. Uzamsal bölütleme adımı ise hareket benzerliklerini kullanarak ya da hareket modeli oturtarak, yörüngelerin gruplanması ile gerçekleştirilir. Zamansal guruplamaya öncelik veren metodların dezavantajı sadece çevrimdışı işlemeye uygun olmalarıdır.

Uzam-zamansal metodlar, $2B+t$ uzayında, uzam ve zaman verisini eşzamanlı olarak kullanırlar. Genel olarak, ortalama kayma (mean shift) ve çizge kesme (graph cut) gibi guruplama algoritmalarını kullanır. Bu metodların iki dezavantajı, yüksek hesaplama gücü gerektirmeleri ve zamansal öncelikli guruplama metodları gibi çevrimiçi işlemeye uygun olmamalarıdır.

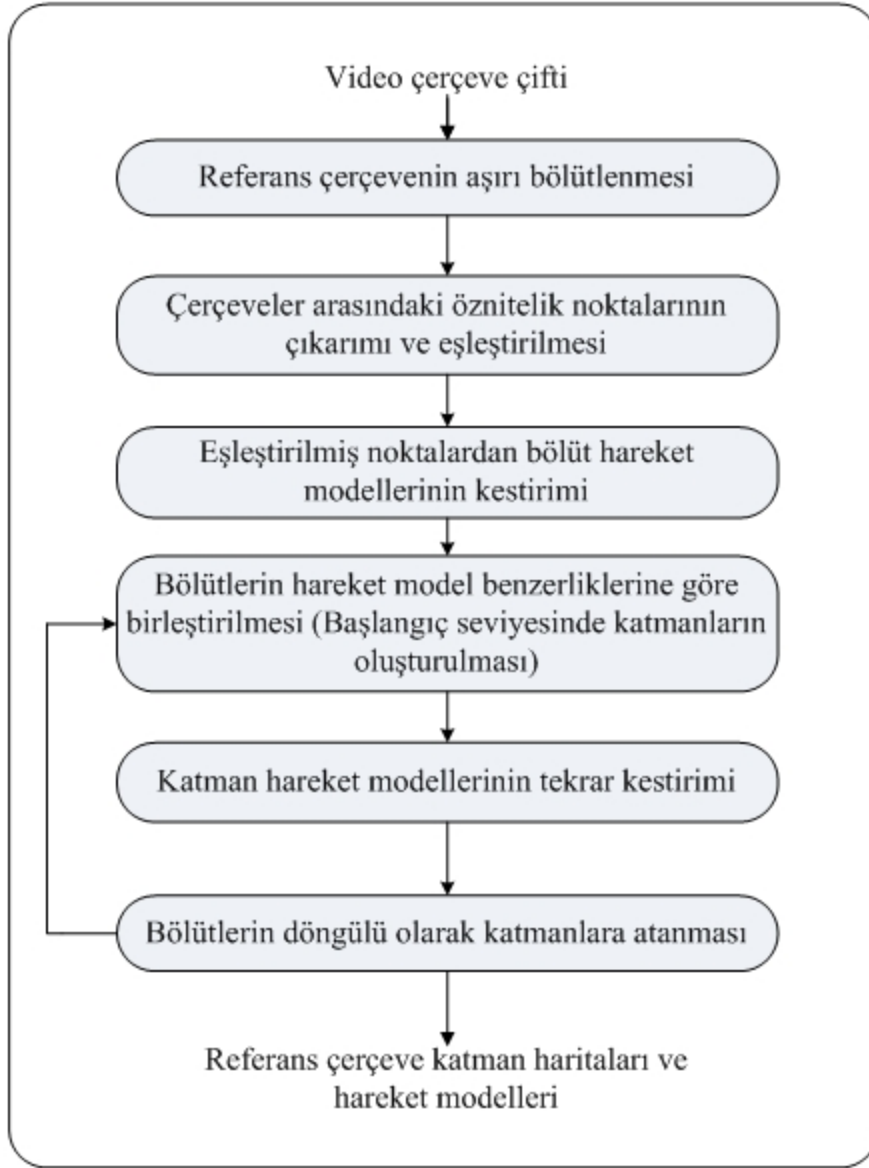
Bu proje kapsamında, çevrimiçi işlemeye uygun oldukları ve genelde ayarlanabilir hesaplama gücüne müsaade ettikleri için uzamsal bölütlemeye öncelik veren bir algoritma incelenmiştir. Bu guruptaki algoritmalar, hareket bölütleme yaklaşımlarına göre ikiye ayrılabilir: hareket benzerliklerini kullananlar [2] [3] [4] ve hareket modeli oturtanlar [5] [6] [7] [8]. Hareket modeli kullanan metodlar da kendi icinde piksel veya bölge tabanlı yaklaşımlar olarak ikiye ayrılabilir. Piksel tabanlı yaklaşımlarda ceza fonksiyonu pikseller üzerinden tanımlanmıştır ve bölütleme kararı her piksel için ayrı verilir. Fakat bu şekilde bir yaklaşım özellikle düşük ve yüksek renk gradyanına sahip bölgelerde hatalı sonuçlar doğurmaktadır. Şekil 42'de piksel tabanlı katman atama kararı veren iyi bilinen bir algoritmanın ([5]) Mobile Calender videosunun bir çerçeve çifti için ulaştığı sonuçlar verilmiştir.



Şekil 42: Wang ve Adelson [5] hareket bölütleme algoritmasının Mobile Calendar videosu için sonuçları. (a)-(b) Orjinal dizinin 3. ve 5. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Orjinal referans çerçeveye ((b)'de gösterilen 5. çerçeve) ait kestirilen hareket katmanları.

Bölge tabanlı hareket bölütleme algoritmalarında ise öncelikle uzamsal boyutta renk ve/veya doku gibi özellikler bakımından benzer nitelikteki pikseller guruplanır ve küçük bölütler oluşturulur. Katman atama işlemi de pikseller yerine benzer özellikteki piksellerin oluşturduğu bölütler üzerinden gerçekleştirilir. Dolayısı ile, bölge tabanlı hareket bölütleme algoritmaları piksel tabanlılara göre daha gürbüzdür ve daha yüksek hassasiyette sonuçlar vermektedir.

Bu proje kapsamında, Bleyer'in [8] kapatmaları dikkate alan bölge tabanlı hareket bölütleme algoritmasının, zamansal boyuttaki tutarlılığı sağlayacak şekilde düzenlenmiş bir versiyonu kullanılmıştır. Algoritmanın akış çizeneği Şekil 43'de verilmiştir. Algoritma iki ana basamakta ele alınabilir: Bölüt tabanlı iklendirme aşaması ve döngülü hareket katmanı bölütleme aşaması.



Şekil 43: Önerilen algoritmanın akış çizeneği [8].

Bölüt tabanlı iklendirme aşaması referans çerçevenin (burada geri yönlü hareket kestirimi kullanıldığı için zamanda ikinci çerçeve) ortalama kayma (mean-shift) algoritması [2] ile aşırı (çok fazla sayıda bölgelere) bölütlenmesi ve bölütlerdeki öznitelik noktalarının Harris köşe detektörü [10] ile bulunması ile başlar. Daha sonra, bu noktaların piramitsel Kanade-Lucas-Tomasi algoritması [11] kullanılarak referans olmayan çerçevedeki eşlerinin bulunması ile devam eder. Sonuçta her bir bölüt için elde edilen ve aykırı örnekler içermesi beklenen nokta eşlerinden RANSAC algoritması [12] [13] kullanılarak bölütlerin hareket modelleri (affine veya homography) kestirilir. Bölüt haritaları ve hareket modelleri bir sonraki aşamada katmanları oluşturmak için kullanılır.

Döngülü hareket katmanı bölütlenme aşaması üç basamakta incelenebilir: (a) benzer hareket modelleri olan bölütlerin birleştirilerek ilk katmanların oluşturulması, (b) katmanlar içerisindeki hareket modellerinin kestirimi ve (c) bölütlerin bir ceza fonksiyonu vasıtası ile katmanlara atanması. Dikkat edilecek olursa, birinci ve üçüncü basamaklar katman haritalarını kestirmeyi amaçlarken, ikinci basamak katman hareket modellerinin kestirimini hedefler. Dolayısı ile,

tahmin edilecek iki öge (katman harita ve hareket modelleri) döngüsel olarak sırayla kestirilmektedir.

Nesne hareketlerinin 6 parametrelili affine veya 8 parametrelili homography modelleri ile ifade edilebileceği varsayımı altında, aynı nesneye ait olan bölütlerin benzer hareket modellerine sahip olması beklenir. Bu çıkarımı takiben, bir hareket modeli benzerlik ölçütünü baz alan ortalama kayma algoritması [14] kullanılarak ilk aşama katmanlar (bölüt kümeleri) oluşturulur. İlk aşama katmanlar için hareket modelleri, katman içerisine düşen öznelik nokta eşleri kullanılarak yine RANSAC algoritması ile kestirilir.

Döngünün son aşamasında piksel tabanlı ve çoklu bileşenli bir ceza fonksiyonu kullanılarak bölütler katmanlara atanır. Ceza fonksiyonunun bileşenleri şunlardır:

- Renk benzerlik terimi,
- Kapatma terimi,
- Uzamsal düzgünlük terimi,
- Zamansal düzgünlük terimi.

Uzamsal düzgünlük terimi, olabildiğince düz nesne kenarları elde edilmesini amaçlamaktadır. Bunun için, farklı katmanlara atanmış iki piksel için bir ceza değeri uygulanmaktadır. Uzamsal düzgünlük terimi şu şekilde tanımlanmıştır:

$$T_{Udüzgünlük} = \sum_{\substack{(p_i, p_j) \in N_4 \\ i < j}} \begin{cases} \lambda_s & : \text{Katman}(p_i) \neq \text{Katman}(p_j) \\ 0 & : \text{Katman}(p_i) = \text{Katman}(p_j) \end{cases}$$

Burada (p_i, p_j) referans görüntüde komşu bir piksel çiftini, N_4 4-komşuluğu, $\text{Katman}(p)$, p pikselini içeren katmanın numarasını dönen fonksiyonu ve λ_s uzamsal katman süreksizliği için tanımlanmış sabit bir ceza değerini ifade etmektedir.

Ceza fonksiyonunun renk benzerlik ve kapatma terimleri, referans çerçevenin geri yönde (referans olmayan görüntü düzlemine) bükülmesini gerektirmektedir. Bunun için, referans görüntü üzerindeki katmanlar (donukluk ve renk haritaları) o anki katman hareket modelleri kullanılarak geri yönde bükülürler. Bu çalışmada, renk haritalarını bükürken bicubic aradeğerleme algoritması kullanılmıştır.

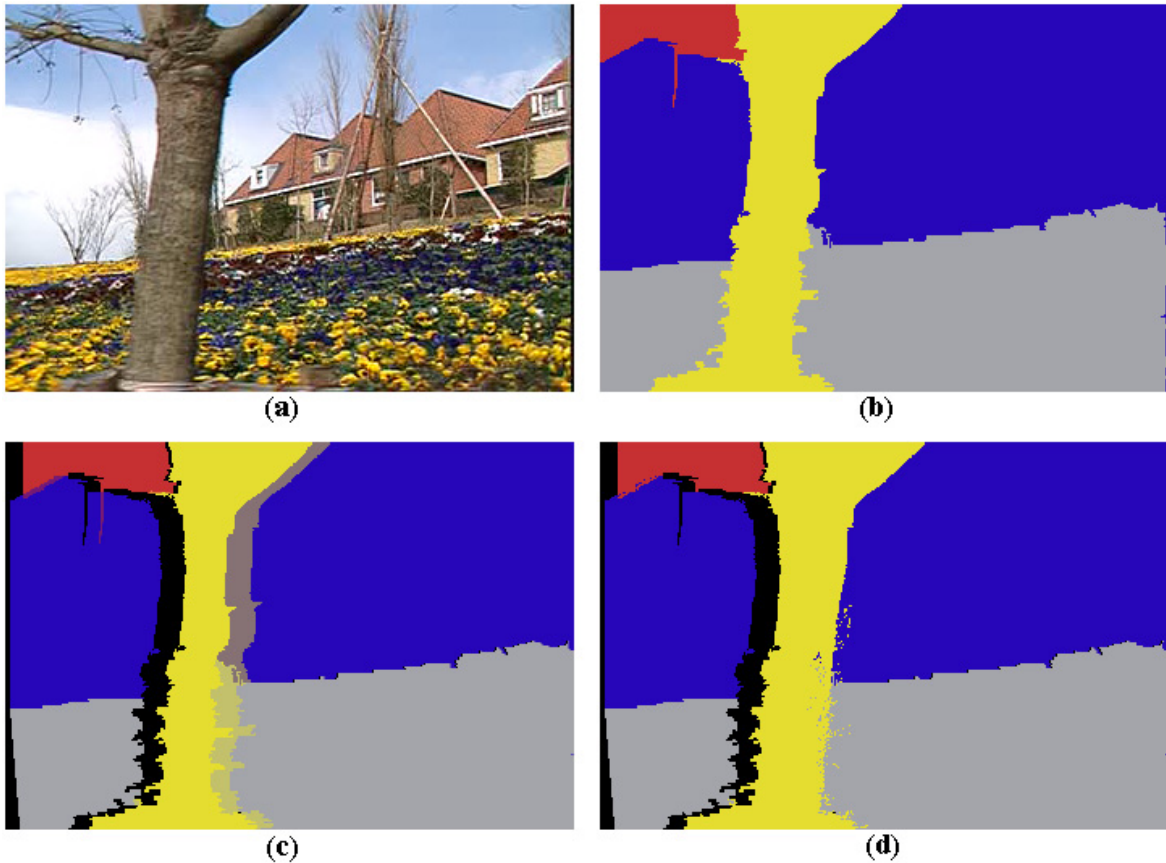
Örnek olarak, Şekil 44'de, Flower Garden dizisinin 324. çerçevesi için kestirilen katmaların bükülmesi gösterilmiştir. Şekil 44.c'de görüldüğü gibi, bükülmüş katmanların oluşturduğu haritalar (dizinin 320. çerçevesine denk gelmektedir), siyah renk ile gösterilen açılmış (uncovered) ve karma renkler ile gösterilen kapanmış (covered) alanlar içermektedir. Kapanmış piksellerde hangi katmanın üstte olduğu kararı, bükülmüş katmanın pikseli ile referans olmayan görüntünün pikseli arasındaki renk değer benzerliği kullanılarak verilir. İki pikselin benzeşmezlik değeri, renk bileşenlerinin değerlerinin mutlak farkının toplamı olarak tanımlanmıştır [8]:

$$Dis(\bar{x}) = \left| I_R^W(\bar{x}) - I_R^N(\bar{x}) \right| + \left| I_G^W(\bar{x}) - I_G^N(\bar{x}) \right| + \left| I_B^W(\bar{x}) - I_B^N(\bar{x}) \right|$$

Burada, $I_R^N(\bar{x})$, $I_G^N(\bar{x})$ ve $I_B^N(\bar{x})$, sırası ile, referans olmayan görüntünün \bar{x} pikselindeki kırmızı, yeşil ve mavi renk bileşen değerlerini ifade etmektedir. Benzer şekilde, $I_R^W(\bar{x})$, $I_G^W(\bar{x})$ ve $I_B^W(\bar{x})$, referans görüntünün hareket katmanlarından birinin bükülmüş haritalarının \bar{x} pikselindeki kırmızı, yeşil ve mavi renk bileşen değerlerini ifade etmektedir. Bükülmüş katmanların üst üste geldiği piksellerde, minimum benzeşmezlik değerini veren katmanın en üstte olduğu kabul edilmektedir. Buna göre ceza fonksiyonunun renk benzerlik terimi şu şekildedir:

$$T_{Renk} = \sum_x Dis(\bar{x})_{\min}$$

Burada, $Dis(\bar{x})_{\min}$ terimi, \bar{x} pikselindeki benzeşmezlik değerinin en küçük olduğu katman (en üstteki katman) için hesaplanmakta ve toplama işlemi bütün görüntü üzerinde gerçekleştirilmektedir.



Şekil 44: Katman bükme işlemi. (a) Flower Garden dizisinin 324. Çerçevesi. (b) 23. Döngüde kestirilen katmanlar. Katman hareketinin ifadesi için affine model kullanılmıştır. (c) Zamanda geri yönde (4 çerçeve) bükülen katmanlar. Siyah ve karma renkteki alanlar 320. çerçevedeki sırası ile açılmış (uncovered) ve kapanmış (covered) pikselleri ifade etmektedir. (d) Bükülmüş katmanların, kapanmış alanlardaki görünür katman belirlendikten sonraki hali.

Ceza fonksiyonunun kapatma terimi, kapanmış ve açılmış alanlardaki pikseller için atanan ceza değerlerinin toplamıdır.

$$T_{Kapat} = \sum_{\substack{\bar{x} \\ P(\bar{x})=\phi}} \lambda_o + \sum_{\substack{\bar{x} \\ |P(\bar{x})|>1}} \lambda_o (|P(\bar{x})|-1)$$

Burada, λ_o sabit piksel kapanma/açılma ceza değerini, $|A|$, A kümesinin eleman sayısını ve $P(\bar{x})$, \bar{x} pikselinde var olan katmanların kümesini veren fonksiyonu ifade etmektedir.

Son olarak, zamansal düzlük terimi, zamanda ardışık iki kestirilmiş katman setinin kullanılması ile hesaplanır. Diğer ceza fonksiyonu terimlerindeki gibi, piksel tabanlı bu bileşen, uyumsuz piksellere atanan sabit ceza değerinin (λ_t) toplamı olarak hesaplanmaktadır:

$$T_{Zdizlilik} = \sum_{\substack{\bar{x} \\ P(\bar{x}) \neq \phi}} \lambda_t \left[1 - \frac{|WKatman(\bar{x}) \cap PKatman(\bar{x})|}{|WKatman(\bar{x}) \cup PKatman(\bar{x})|} \right]$$

Burada, $PKatman(\bar{x})$ ve $WKatman(\bar{x})$ terimleri \bar{x} pikselindeki sırası ile zamanda geri katmanı ve geri yönde bükülmüş zamanda ileri (ve görünür) katmanı veren fonksiyonları ifade etmektedir. $X \cap Y$ ve $X \cup Y$ terimleri, sırası ile, X ve Y katmanlarının kesiştiği ve birleştiği bölgeyi ifade etmektedir. Son olarak, $|R|$, R bölgesinin piksel cinsinden alanını ifade etmektedir.

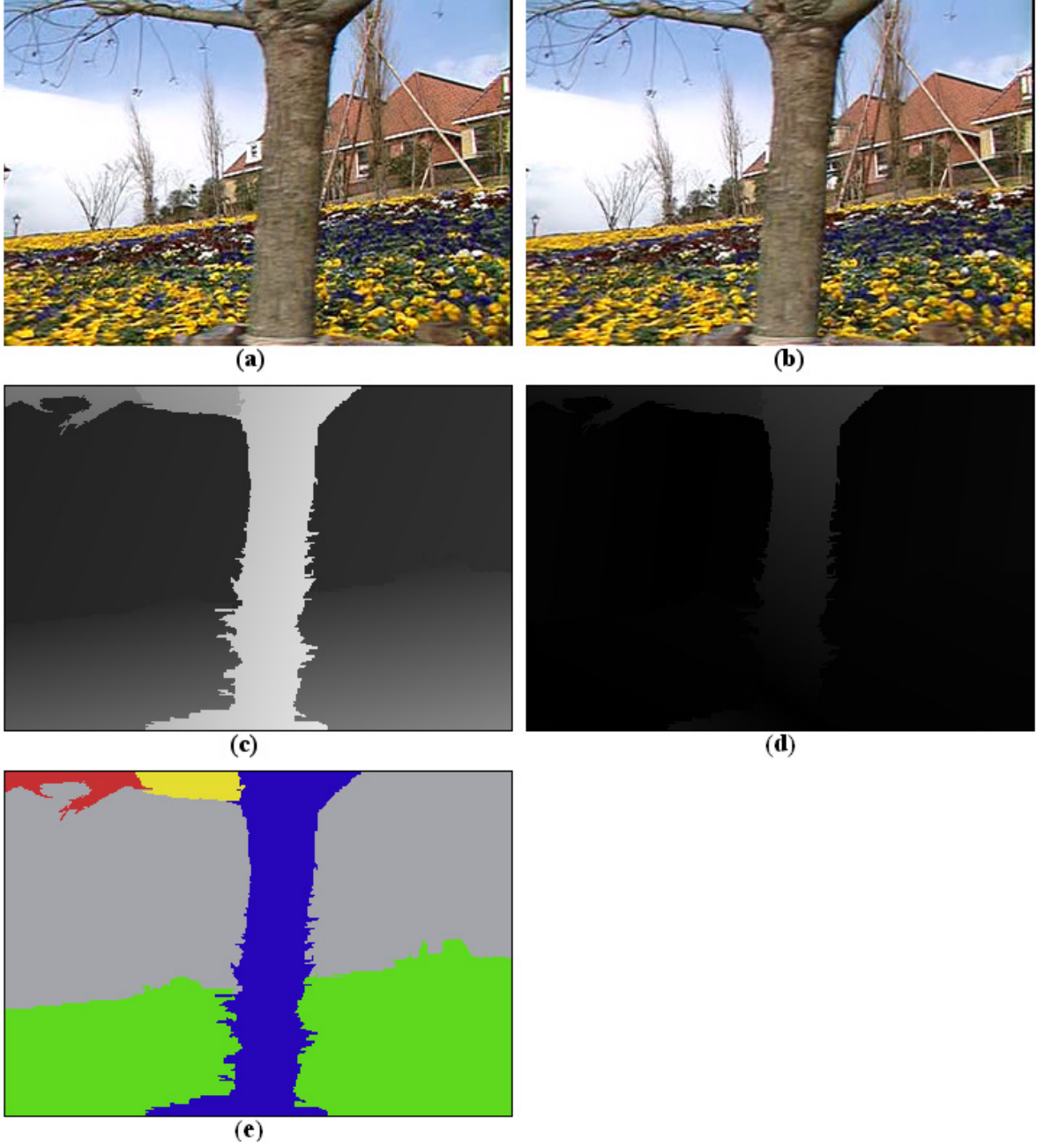
Çoklu bileşenli toplam ceza fonksiyonu, yukarıda anlatılan dört terimin toplamı ile elde edilir:

$$C = T_{Rank} + T_{Kapat} + T_{Udizlilik} + T_{Zdizlilik}$$

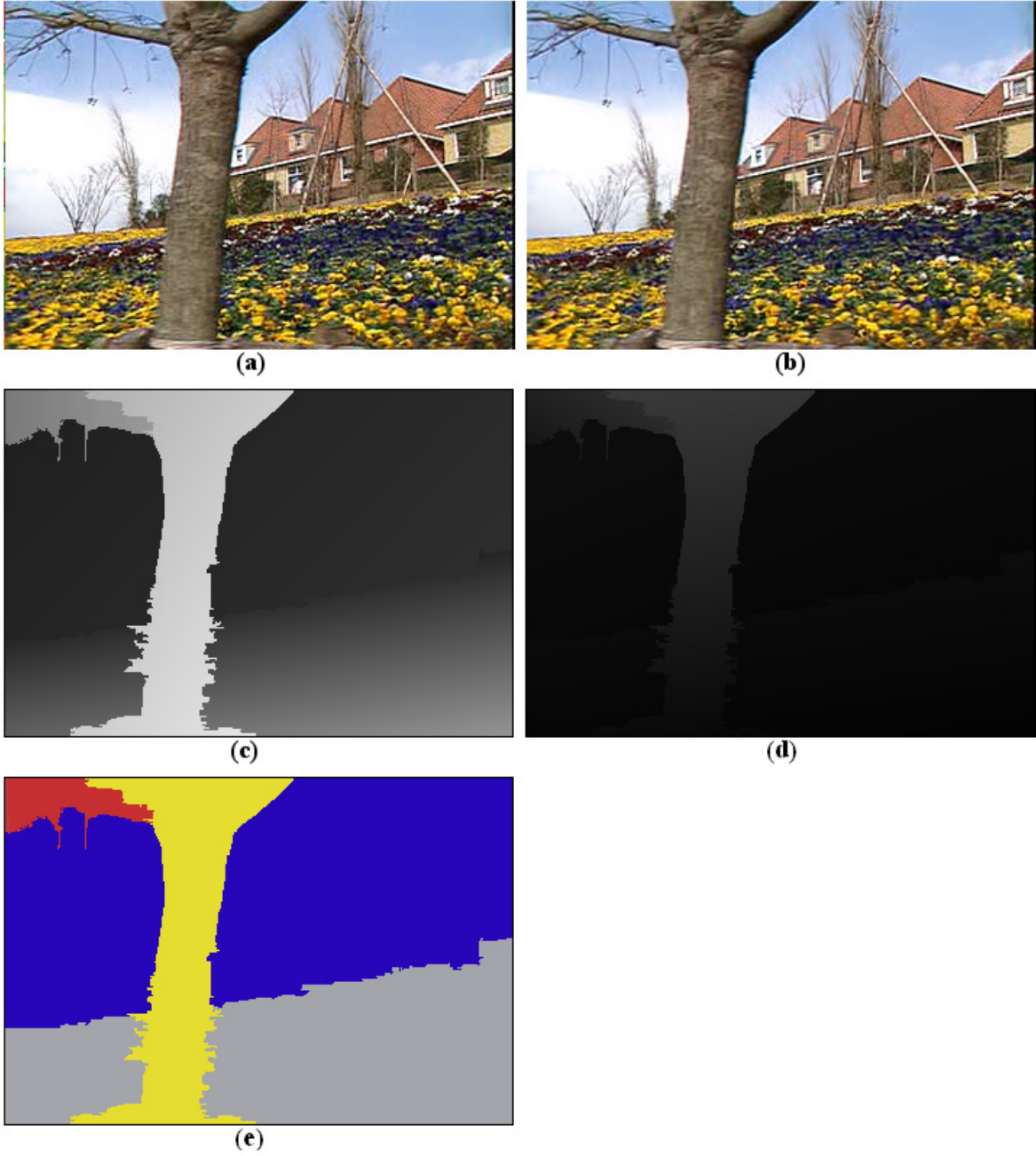
Ceza fonksiyonu, olası kapatmaların etkisini dikkate almak için piksel alanında tanımlanmıştır, fakat optimizasyon, daha gürbüz sonuçlar ve daha yüksek hassasiyette nesne kenarlarının bulunması için bölüt alanında gerçekleştirilmektedir.

3.2.1 Deneysel Sonuçlar

Algoritma sonuçları Şekil 45 - Şekil 48'da *Flower Garden* ve *Mobile Calendar* çerçeve dizileri için verilmiştir. Bütün şekiller için geçerli olmak üzere, (a) ve (b) algoritma girdisi olan orijinal görüntüleri, (c) ve (d) kestirilen hareket alanınının sırası ile mutlak x ve y bileşenlerini, ve (e) referans görüntü için kestirilen hareket katmanlarını ifade etmektedir. Bütün deneylerde, sadece tek sayıdaki çerçeveler kullanılmıştır ve katmanların hareketi affine modeli ile ifade edilmiştir.



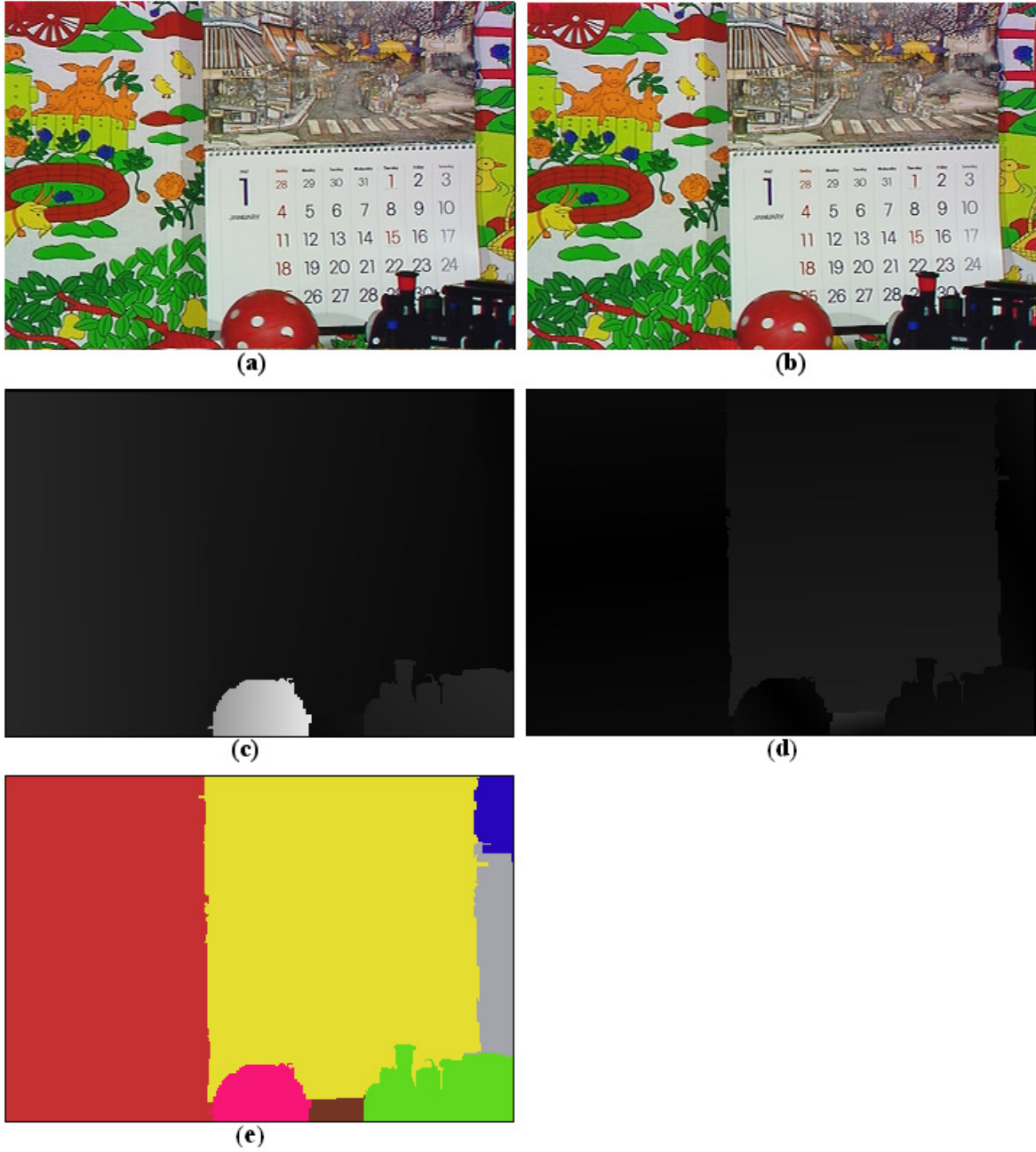
Şekil 45: Hareket bölütleme algoritmasının Flower Garden çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 307. ve 309. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü (burada 309, çerçeve).için kestirilen hareket katmanları.



Şekil 46: Hareket bölütleme algoritmasının Flower Garden çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 319. ve 321. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü için kestirilen hareket katmanları.



Şekil 47: Hareket bölütleme algoritmasının Mobile Calendar çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 3. ve 5. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü için kestirilen hareket katmanları.



Şekil 48: Hareket bölütleme algoritmasının Mobile Calendar çerçeve dizisi için sonuçları. (a)-(b) Orijinal dizinin 11. ve 13. çerçeveleri. (c) Kestirilen hareket alanının mutlak x ve (d) mutlak y bileşenleri. (e) Referans görüntü için kestirilen hareket katmanları.

3.2.2 Sonuç

Deneyisel sonuçlar önerilen algoritmanın birden çok nesnenin yer aldığı ve yaptıkları hareketler sonucu farklı katmanlar oluşturduğu karmaşık sahneler için oldukça başarılı sonuçlar verdiğini göstermektedir. Algoritmanın performansını sağlayan temel unsurun açılma ve kapanma bölgelerini dikkate almak ve buna bağlı olarak daha hassas sınırlar oluşturmak olduğu söylenebilir.

3.2.3 Referanslar

- [1] Megret Remi, DeMenthon Daniel, A Survey of Spatio-Temporal Grouping Techniques, Technical report: LAMP-TR-094, University of Maryland, College Park, 2002.
- [2] D. Wang, Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking, IEEE Trans. on Circuits and Systems for Video Tech., vol. 8, no. 5, pp. 539546, 1998.
- [3] Q. Ke, T. Kanade, A Subspace Approach to Layer Extraction, in IEEE International Conference on Computer Vision and Pattern Recognition, pp. 255262, 2001.
- [4] M. Gelgon, P. Bouthemy, A Region-level Motion-based Graph Representation and Labeling for Tracking a Spatial Image Partition, Pattern Recognition, vol. 33, no. 4, pp. 725740, 2000.
- [5] J. Y. A.Wang and E. H. Adelson, Representing moving images with layers, IEEE Trans. Image Process. 3(5), 625–638, Sept. 1994.
- [6] Y. Altunbasak, P.E. Eren and A.M. Tekalp, Region-based parametric motion segmentation using color information, Graphical Models and Image Processing 60, pp. 13–23, 1998.
- [7] G. D. Borshukov, G. Bozdagi, Y. Altunbasak, and A. M. Tekalp, Motion segmentation bu multistage affine classification, IEEE Trans. Image Processing, vol. 6, pp. 1591–1594, Nov. 1997.
- [8] M. Bleyer, Segmentation-based stereo and motion with oclusions, Ph.D. Thesis, Vienna University of Technology, 2006.
- [9] C. Dorin, M.Peter , Mean Shift : A robust approach toward feature space analysis, IEEE Transactions on pattern analysis and machine intelligence vol:24,no:5, May 2002.
- [10] C. Harris and M. Stephens, A Combined Corner and Edge Detector. Proc. Alvey Vision Conf., Univ. Manchester, pp. 147-151, 1988.
- [11] J.Y. Bouguet, Pyramidal Implementation of the Lucas Kanade feature tracker description of the algorithm, Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.
- [12] M.A. Fischler, R.C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM, vol. 24, no. 6, June 1981.
- [13] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, Second Edition, 2004.
- [14] C. Dorin, M.Peter , Mean Shift : A robust approach toward feature space analysis, IEEE Transactions on pattern analysis and machine intelligence vol:24,no:5, May 2002.

3.3. BASKIN KÜMELER İLE VIDEO ÇEKİM SINIRI BÖLÜTLEME

3.3.1 Video Çekim Sınırı Bulma Literatür Özeti

Video sıkıştırma teknolojisindeki son gelişmeler, sayısal kameraların kullanımındaki artış, yüksek kapasiteli sayısal bilgi sistemlerinin yaygınlaşması ve bunlara ek olarak internet kullanımının hayatın her alanına girmesi, sayısal video verilerinin hacminde ve kullanımındaki artışı da beraberinde getirmiştir. Bu gelişmelerin neticesinde mevcut video verilerini etkin bir şekilde sınıflandıracak, ilgili bilgileri arayıp kullanıcıya sunacak kabiliyette sistemlere olan ihtiyaç ortaya çıkmıştır.

Bununla birlikte, videolar büyük veri hacmine sahip oldukları için, bir videonun içinden gerekli bilgiye ulaşmak çok kolay olmamaktadır. En geçerli yol video dizisini daha küçük parçalara ayırmaktır. Bu da video dizisi içinde hareket sürekliliği içeren, “çekim” olarak adlandırılan parçaların, çekim sezme yöntemlerini kullanarak bulunmasıyla sağlanır.

Çekim sezme, birçok içerik temelli video analiz yönteminin ilk basamağı olarak değerlendirildiği için 15 yılı aşkın bir süredir araştırmacılar tarafından aktif olarak çalışılmaktadır. Önerilen çeşitli algoritmalar ile bu konuda belli bir olgunluğa ve başarı düzeyine ulaşılmıştır. Farklı yaklaşımlar ile geliştirilen algoritmalar içerik olarak ne kadar farklı olursa olsun, bütün algoritmalar aynı çekim içerisinde görsel bir süreklilik olduğu varsayımına dayanmaktadır.

Bu ortak özellikten hareketle, çekim sezme problemi için önerilen çözümleri, önerilen algoritmadan bağımsız olarak, 3 ana bölümde değerlendirmek mümkündür [1]:

- görsel içeriğin temsili,
- süreklilik sinyalinin oluşturulması,
- süreklilik sinyalinin sınıflandırılması.

Süreklilik sinyali en basit olarak ardışık iki çerçeve arasındaki özellik farkına eşit olarak kabul edilebileceği gibi daha karmaşık operasyonlar ile elde edilen ve imge dizisinin aynı kamera çekiminde kaldığı sürece sabit seviyede duran bir sinyale verilen genel terimdir [1].

Mevcut araştırmalar incelendiğinde, görsel içeriğin temsili noktasında belli birtakım özelliklerin tercih edildiğini görebiliyoruz. Önerilen ilk çalışmalarda görsel içerik olarak piksel değerleri kullanılmıştır. Bu yöntemlerde piksellerin parlaklık ya da renk değerlerinin farklarına bakarak çekim geçişleri tespit edilmeye çalışılmıştır [2]. Ortamın aydınlanmasındaki çok küçük değişikliklerin dahi piksel değerlerinde ciddi değişiklikler netice vermesi, sadece belli değerlerin üzerinde değişiklik gösteren piksellerin hesaba katıldığı yeni algoritmaların geliştirilmesine sebep olmuştur [3] [4]. En çok tercih edilen içerik temsillerinden bir tanesi de histogramlardır [5] [6]. Histogramlar resimlerin geneli ile alakalı özellikler olduğundan dolayı lokal değişikliklerden ve kamera hareketlerinden daha az etkilenmektedir. Bu alanda kullanılan bir diğer özellik ise kenar bilgisidir. Çekim sınırının sağında ve solundaki çerçevelerin kenar yerleşimleri farklı olduğundan çekim değişikliklerinde belli yerlerde kenar noktaları kaybolmakta ve belli yerlerde yeni kenar noktaları oluşmaktadır. Bu

değişikliklerin oranına bakarak çekim sezme yapabilen algoritmalar önerilmiştir [7] [8]. Aynı çekim içerisinde hareket vektörlerinin de süreklilik arzettiği, çekim sınırında hareket vektörlerinde de ani değişiklikler olduğunun tespit edilmesi üzerine çerçevelerdeki hareket bilgisini içerik temsili olarak kullanan algoritmalar da geliştirmiştir [9] [10]. Bunun özellikler dışında, sıkıştırılmış video verisi üzerinde çalışan algoritmalar da mevcuttur. Bu algoritmalar genel olarak sıkıştırılmış video verisi içerisinde hazır olarak bulunan özellikleri içerik temsili olarak kullanmışlardır. DCT katsayıları, DC resmi, bit oran bilgisi makro blok bilgisi gibi özellikler en çok tercih edilen özelliklerdir [11].

Görsel içerik için hangi özelliğin kullanılacağı belirlendikten sonra, bu özellik kullanılarak süreklilik sinyali oluşturulmaktadır. Bu alanda geliştirilen ilk algoritmalar [3] [4] [7] [10] genel olarak ardışık iki çerçeve arasında karşılaştırma yapma esasına dayanmaktadır. Bu methodlarda süreklilik sinyali basit olarak ardışık iki çerçeve arasındaki özellik farkına eşittir. Bu yöntemin kamera ya da cisim hareketlerini ayırtmaktaki zorlukları ve ani aydınlanma değişikliklerini çekim sınırı olarak algılamasından dolayı sonradan önerilen algoritmalarda süreklilik sinyali oluşturulurken sadece ardışık iki çerçeve değil belli bir komşulukta bütün çerçevelerin bilgisi kullanılmıştır. Özellikle son yıllarda çizge temelli yaklaşımlar ile geliştirilen çekim sezme yöntemleri bu türden algoritmalar [1].

Süreklilik sinyalinden çekim sınırlarının belirlenmesinde kullanılan yöntemleri de 2 başlık altında toplayabiliriz. İlk önerilen yöntemler genel olarak süreklilik sinyalinin belli bir eşik değeri ile karşılaştırılması esasına dayanan algoritmalar [3] [4] [7] [10]. Son geliştirilen yöntemler ise çekim sezme problemine bir örüntü tanıma problemi olarak yaklaşmakta ve bu alandaki metodları özellikle makina öğrenme algoritmalarını kullanmaktadırlar. Destek vektör makinaları ve k-means yöntemleri bu alanda kullanılan tekniklerdir [1] [12] [13].

3.3.2 Baskın Kümeler

Örüntü tanıma ve yapay görü problemlerinin çözümünde çizge temelli yaklaşımların kullanımı giderek artmaktadır. Baskın kümeler yöntemi, Pavan ve Pelillo [14] tarafından yeni bir topaklama (clustering) tanımı olarak önerilmiştir. Topağın çizge teorisi temelli tanımı yapıldıktan sonra imge bölütleme alanında önerdikleri yöntemi test etmişlerdir.

Baskın kümeler yönteminde kullanılan topak tanımlaması temelde diğer topak yaklaşımlarından farklı değildir:

1. Bir topak içindeki tüm nesnelere birbirine benzemelidir.
2. Bir topağın dışındaki tüm nesnelere topaktaki nesnelere benzememelidir.

Ağırlıklı ve yönsüz çizge $G(V,E)$, düğümlerden ve bu düğümleri birleştiren kenarlardan oluşmaktadır. N adet düğüm V kümesini oluşturur. Kenarların her birinin belirli bir ağırlığı vardır ve i ile j düğümünü birleştiren bir kenarın ağırlık değeri $e(i,j)$ ile gösterilir. Kenarlar E kümesini oluşturur. Kenar ağırlık değerleri $W(i,j)=e(i,j)$ matrisini oluşturur. W matrisi $N \times N$ simetrik bir matristir ve $W(i,i) = 0$ 'dır.

S boş olmayan bir küme ve $S \subseteq V$ olmak üzere, bir i düğümü bu kümeye ait olsun, $i \in S$. Bir i düğümünün S kümesine göre ortalama ağırlık derecesini veren $awdegS(i)$ fonksiyonu, şu şekilde tanımlanmıştır:

$$awdeg_S(i) = \frac{1}{|S|} \sum_{j \in S} e(i, j) \quad (1)$$

Ayrıca S kümesi içerisinde olmayan bir j düğümü için, $j \notin S$, ϕ_S şu şekilde tanımlanır:

$$\phi_S(i, j) = e(i, j) - awdeg_S(i) \quad (2)$$

ϕ_S fonksiyonu i ve j düğümleri arasındaki benzerliği i düğümünün S kümesi içerisindeki düğümlere olan ortalama benzerliğine göre ölçer. Bir i düğümünün S kümesine göre ağırlığı şu şekilde hesaplanır:

$$a_S(i) = \begin{cases} 1, & \text{eğer } |S|=1, \\ \sum_{j \in S \setminus \{i\}} \phi_{S \setminus \{i\}}(j, i) a_{S \setminus \{i\}}(j), & \text{diğer} \end{cases} \quad (3)$$

S kümesinin toplam ağırlığı şu şekilde tanımlanır:

$$A(S) = \sum_{i \in S} a_S(i) \quad (4)$$

Boş olmayan bir S çizge düğümleri kümesinin ($S \subseteq V$), boş olmayan ve $A(T) > 0$ olan bir T çizge düğümleri kümesini kapsadığını kabul edelim, $T \subseteq S$. S kümesi aşağıdaki iki koşulu sağlarsa baskın küme olarak adlandırılır:

1. $a_S(i) > 0$, $i \in S$ için,
2. $a_{S \cup \{i\}}(i) < 0$, $i \notin S$ için.

Yukarıda yer alan iki koşul, daha önce açıklanan iki koşula karşılık gelmektedir. Birinci koşul bir topak içerisindeki nesnelere benzerliğini, ikinci koşul topak dışındaki nesnelere benzerliğini sağlamaktadır.

Baskın kümenin yukarıda açıklanan denklemler yolu ile bulunması çok fazla hesaplama yükü getirmektedir. Özellikle bir çizgedeki düğüm sayısına bağlı olarak uygulanabilir bir çözüm olmayabilir. Baskın kümenin bulunması için kullanılacak bir başka yaklaşım tekrarlayıcı dinamiği (*replicator dynamics*) olarak adlandırılan düzgün ve sürekli optimizasyon yönteminin (straightforward continuous optimization techniques) kullanılmasıdır [14]. V kümesi içerisindeki düğümlerin ağırlıklı karakteristik vektörü x ile gösterilmiştir. Bu vektördeki değer 0'dan büyük olduğu zaman o düğüm baskın kümenin içinde olduğu kabul edilir. Bunun için x vektörünün ilk değeri ($t=0$ anında) atanır. Zaman içerisinde x vektörü aşağıdaki denklem ile değişime uğrar. Ardışık iki zaman aralığındaki değişim yeteri kadar küçük olduğunda işlem sonlanır. İşlem sonlandığı andaki x değerlerine göre çizge iki çizgeye üleştirilir.

$$x_i(t+1) = x_i(t) \frac{(\mathbf{W}\mathbf{x})_i}{\mathbf{x}(t)^T \mathbf{W}\mathbf{x}(t)} \quad (5)$$

3.3.3 Baskın Kümeler Yöntemi ile Çekim Sınırı Sezme

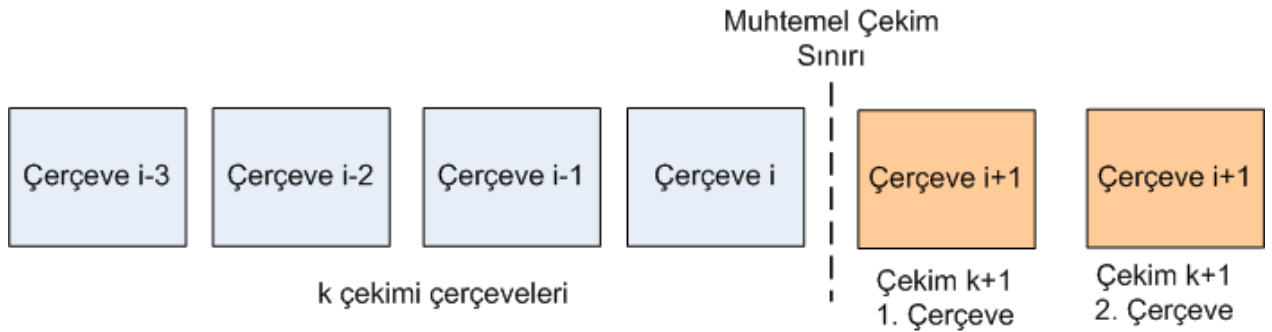
Önerilen yöntem şu şekilde özetlenebilir:

1. Çekim sınırı olmaya aday pozisyonlar belirlenir:

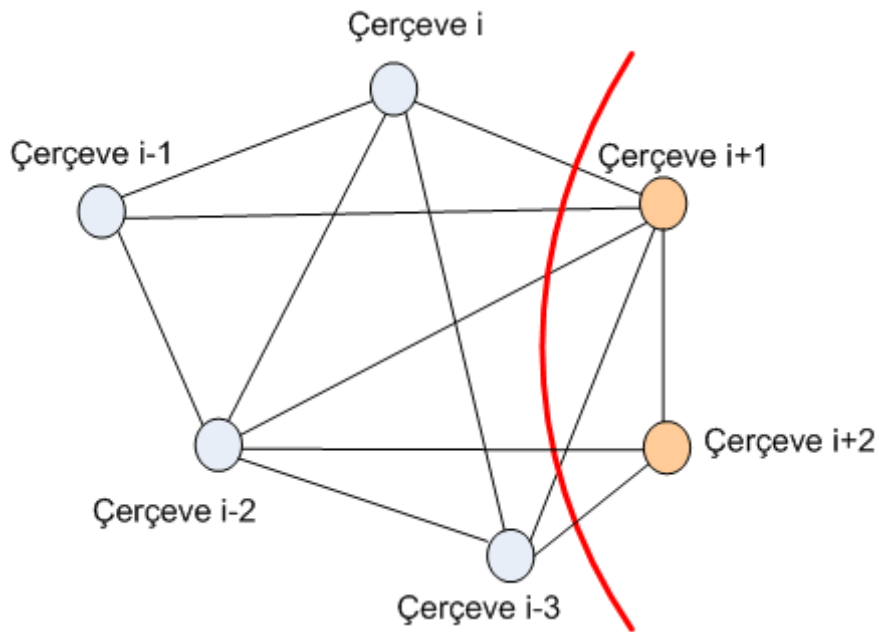
İki çerçeve arasında bir çekim sınırı olup olmadığı mutlak fark toplamına (sum of absolute difference) dayalı olarak yapılır. İki çerçevenin piksel değerleri arasındaki fark bulunur. Bu fark belli bir eşik değerinden büyük ise diğer adımlarda anlatıldığı şekilde yöntem uygulanmaya devam edilir. Eğer mutlak fark toplamı belirlenen eşik değerinden küçük ise bu iki çerçeve arasında bir çekim sınırı olmadığına karar verilerek bir sonraki çerçeveye geçilir. Bundan sonra adımlar uygulanmaz.

Böyle bir önışleme adımı sayesinde çekim sınırı olma ihtimali çok düşük olan çerçeveler atlanarak algoritmanın hızı artırılmıştır.

2. Aday pozisyonlar için, çekim sınırının solundan 4 çerçeve ve sağından 2 çerçeve alınarak (Şekil 49) çizge gösterimi (*graph representation*) oluşturulur (Şekil 50).



Şekil 49 – Aday Çekim Sınırı



Şekil 50 – Aday Çekim Sınırı İçin Çizge Gösterimi

3. Video içerik temsili için özellik belirlenerek belirlenen özelliğe göre ağırlıklı benzerlik matrisi oluşturulur.
4. Başlangıç durum vektörü olarak $x_{t=0} = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6]$ alınarak tekrarlayıcı dinamiği yaklaşımı ile belirlenen 6 nokta için baskın küme bulunur. Bu işlem sırasında iteratif olarak durum vektörü güncellenir. Durum vektörünün bütün elemanları için güncelleme değeri belli bir eşik değerinin altında olduğu zaman ya da toplam iterasyon sayısı 15'e ulaştığı zaman iterasyonlar durdurulur.
5. Gerçek çekim sınırları için baskın kümenin ilk 4 çerçeve olarak bulunması beklenmektedir. Bu yüzden durum vektörünün son halinde ilk 4 eleman için değerlerin başlangıç değeri olan 1/6'dan büyük, 5. ve 6. elemanların değerlerinin ise belli bir eşik değerinden küçük olmasını gerekmektedir. Bu durumu sağlayan noktalar çekim sınırı olarak tanımlanır.

Bir çerçevede içerik olarak en önemli bilginin çerçevenin merkezinde olduğu varsayımından hareketle çerçeve üzerinde bir ilgi alanı tanımlanmış ve çerçeveler üzerinde gerçekleştirilecek bütün işlemler sadece bu ilgi alanı üzerinde gerçekleştirilmiştir. Bunun yanında belirlenen ilgi alanı da x ve y yönünde 2 ile geri ölçeklendirilerek (*downsampling*) edilerek bütün işlemler bu küçük ölçekli çerçeveler üzerinde gerçekleştirilmiştir. Bu sayede hem videonun merkezindeki önemli içeriğe odaklanılmış, hem yanlış çekim sezmeye sebep olan video efektlerinin (altyazılar, video-in-video efekt gibi) etkisi azaltılmış, hem de algoritma daha hızlı çalışır hale getirilmiştir.

Bu yöntem TRECVID 2003 çekim sınırı sezme test videoları üzerinde test edilmiş olup sonuçlar aşağıdaki tabloda özetlenmiştir.

Tablo 3 – Baskın Kümeler Test Sonuçları

#	Piksel Farkı			Histogram Farkı			Hareket Farkı			Piksel ve Hareket farkı Birlikte		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
1	0,80	0,81	0,80	0,78	0,54	0,64	0,87	0,81	0,84	0,87	0,72	0,79
2	0,92	0,87	0,89	0,87	0,64	0,74	0,96	0,84	0,90	0,95	0,83	0,89
3	0,87	0,84	0,85	0,88	0,69	0,77	0,95	0,84	0,89	0,93	0,82	0,87
4	0,94	0,85	0,89	0,88	0,65	0,75	0,99	0,81	0,89	0,97	0,79	0,87
5	0,87	0,82	0,84	0,85	0,53	0,65	0,94	0,72	0,82	0,94	0,71	0,81
6	0,86	0,75	0,80	0,81	0,60	0,69	0,91	0,72	0,80	0,90	0,73	0,81
7	0,91	0,85	0,88	0,89	0,68	0,77	0,97	0,78	0,86	0,97	0,77	0,86
8	0,96	0,90	0,93	0,90	0,76	0,82	0,99	0,87	0,93	0,99	0,83	0,90
Ortalama	0,89	0,84	0,86	0,86	0,64	0,73	0,95	0,80	0,87	0,94	0,78	0,85

Test sonuçları hassasiyet (P), hatırlama (R) ve F1 kriterleri ile elde edilmiştir. Testler sırasında içerik temsili olarak piksel farkı, histogram farkı ve hareket farkı tercih edilmiş ayrıca iki özelliğin bir arada kullanıldığı bir test de gerçekleştirilmiştir.

3.3.4 Sonuç

Test sonuçları hareket farkı kullanıldığı zaman baskın kümeler yönteminin en iyi sonucu verdiğini göstermektedir. Bunun yanında hareket farkının kullanıldığı yöntemin hız olarak en yavaş yöntem olduğu gözlenmiştir.

3.3.5 Referanslar

- [1] J. Yuan, H. Wang, L. Xiao et al., "A Formal Study of Shot Boundary Detection," IEEE Transactions on Circuits and Systems for Video Technology, pp. 168-186, 2007.
- [2] T. Kikukawa, and S. Kawafuchi, "Development of an automatic summary editing system for the audio-visual resources," *Transactions on Electronics and Information J75-A* pp. 204-212, 1992.
- [3] K. Otsuji, Y. Tonomura, and Y. Ohba, "Video browsing using brightness data," in *Proc. SPIE VCIP'91*, vol. 1606, pp. 980-989, 1991.
- [4] T. Kikukawa, and S. Kawafuchi, "Development of an automatic summary editing system for the audio-visual resources," *Transactions on Electronics and Information J75-A* pp. 204-212, 1992.
- [5] H. J. Zhang, A. Kankanhalli, and S. Smoliar, "Automatic Partitioning of Full-Motion Video," *Multimedia Systems*, vol. 1, pp. 10-28, 1993.
- [6] J. Mas, and G. Fernandez, "Video Shot Boundary Detection based on Color Histogram," *Notebook Papers TRECVID2003*, 2003.
- [7] R. Zabih, J. Miller, and K. Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," *Proc. ACM Multimedia 95*, pp. 189-200, 1995.
- [8] R. Lienhart, "Comparison of Automatic Shot Boundary Detection Algorithms," *Image and Video Processing VII*, 1999.
- [9] B. Shahraray, "Scene change detection and content-based sampling of video sequences," *Proc. IS&T/SPIE 2419*, pp. 2-13, 1995.
- [10] Y. Kawai, H. Sumiyosi, and N. Yagi, "Shot Boundary Detection at TRECVID 2007," 2007.
- [11] I. Koprinska, and S. Carrato, "Temporal Video Segmentation: A Survey," *Signal Processing: Image Communication*, 16, 2001.
- [12] B. C. Song, and J. B. Ra, "Automatic Shot Change Detection Algorithm Using Multi-stage Clustering for MPEG-Compressed Videos," *Journal of Visual Communication and Image Representation*, vol. 12, no. 3, pp. 364-385, 2002.
- [13] C. Petersohn, "Fraunhofer HHI at Trecvid 2004: Shot Boundary Detection System," *TREC Video Retrieval Evaluation Online Proceedings*, 2004.
- [14] M. Pavan, and M. Pelillo, "Dominant Sets and Pairwise Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 167-172, 2007.

4. UYGULAMALAR

Raporda řu ana kadar sunulan arařtırmalar, herhangi bir uygulamada ara basamaklarda kullanılabilir temel tekniklerle iliřkilidir. Raporun bu b3l3m3nde 5 ayrı uygulama ile g3rsel veriden anlamsal bilgi ıkarılmaya y3nelik alıřmalar anlatılacaktır. Bu alıřmalardan ilki řekil bilgisi kullanarak otomatik hedef (hava tařıtı) tanıma y3neliktir. İkinci uygulama alanı ise video dizileri 3zerindeki yazıların bulunup, tanınması ve dolayısıyla imge 3zerindeki kavramlar ile iliřkilendirilmesi ile alakalıdır. 33nc3 uygulama g3venlik kameralarından anlamsal bilgileri ıkarabilmek iin 3rt3řen nesnelerin ayrıřtırılmasına y3nelik bir yaklařım sunulmaktadır. Projede d3rd3nc3 bir uygulama olarak, TV yayınlarını birbirlerinden ayırabilmek ve ierikleri hakkında ıkarımlar yapabilmek amacıyla opak ve saydam logoları ayırabilen bir algoritma sunulmuřtur. Son olarak, son yıllarda 3nem kazanan bir verinin iřlenmiř kopyalarının veritabanında bulunabilmesi problemine y3nelin bir 3z3m 3nerisi ortaya konmuřtur.

4.1. řEKİL BİLGİSİ KULLANARAK HEDEF TANIMA

Otomatik hedef tanıma problemi, farklı sınıflara ait hedeflerin ait oldukları sınıfların bulunması olarak tanımlanır. Bu problemin 3z3m3nde, daha 3nceden elde edilmiř ve ait oldukları sınıflar bir uzman tarafından iřaretlenmiř g3r3nt3ler (eđitim k3mesi) kullanılarak, yeni gelen bir g3r3nt3n3n olası sınıflardan hangisine ait olduđunun belirlenmesi yolu izlenir.

Tanım problemi'nin 3z3mlerinde iki temel unsur bulunur: G3r3nt3lerin tanıma kullanılacak *3zellikleri* ve *tanım yöntemi*. Bu iki unsur, problemin dođasına ve ihtiyalara g3re belirlenir.

G3r3nt3lerin ait oldukları sınıfların karakteristik 3zellikleri g3r3nt3ler 3zerinde eřitli operasyonlar yapılarak bulunabilir. Bu operasyonlar, temelde, ardıřık ya da tekil, dođrusal ya da dođrusal olmayan filtreleme y3ntemleridir. Bu 3zelliklere, optik kenarlar ve k3řeler, renk dađılımı, uzamsal frekans ieriđi, nesne silüetinin b3lgesel dađılımı ve sınırları 3rnek olarak verilebilir. Bu 3zelliklerden her birisi farklı tanıma problemlerinde farklı deđerlerde bilgi sađlar.

Bu alıřmada řu ana kadar kullanılan y3ntem, hedeflerin tanınmasında silüetlerinin kullanılmasına dayalıdır. Nesnelerin silüetlerinden ıkarılan 3zelliklerin tanıma ne kadar faydalı olduđu incelenmiřtir.

Yapılan deneyler, silüetlerden ıkarılan bilgilerin nesnelere ayırmada ne kadar ayırıcı bilgiler sađlandıđının anlařılmasına y3neliktir. Literat3rde bařarılı olduđu ifade edilmiř B3l3m 2.1'de tanımlanan 3znitelikler denenmiřtir. Tanım yöntemi olarak "En yakın komřu" y3nteminin 3 farklı hali kullanılmıřtır. Bu 3zelliklerin elde edilme yolları ve tanıma y3ntemleri ařađıda 3zetlenmiřtir.

4.1.1. Sınıflandırma Y3ntemleri

Bu alıřma, řu ana kadar, řekil tabanlı y3ntemlerde hangi 3zelliklerin kullanılmasının daha faydalı olacađının anlařılması 3zerinde durmuřtur. Bu nedenle kullanılan sınıflandırıcılar, ařitlendirilmemiřtir. alıřmada kullanılan sınıflandırıcılar, *en yakın komřu* algoritmasını ve iki t3revidir.

En yakın komşu (EYK) yöntemi, yeni gelen görüntüden çıkarılan özellik vektörü ile eğitim kümesinden oluşturulan özellik vektörleri arasındaki uzaklıkların (benzerliklerin) hesaplanması ve en yakın (benzer) vektöre sahip eğitim kümesi elemanının ait olduğu sınıfın yeni gelen görüntü atanması şeklinde uygulanır.

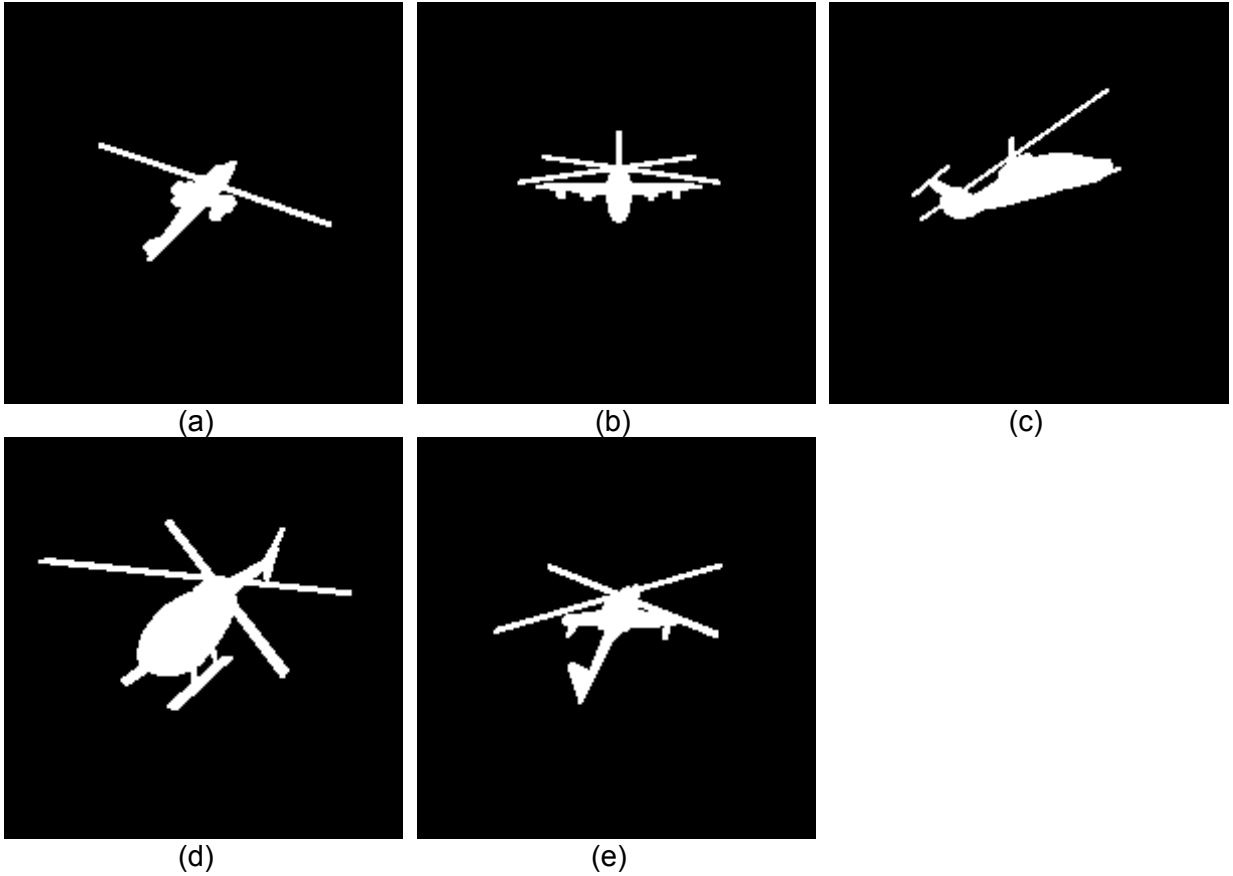
k-EYK yöntemi, EYK'nın oylama içeren bir versiyonudur. Yeni gelen görüntüye en yakın k tane eğitim kümesi görüntüsü alınır. Bu görüntülerin ait oldukları sınıflara bakılarak, en sık karşılaşılan sınıf yeni gelen görüntüye atanır.

Bundan farklı olarak, bir üçüncü yöntem de yukarıda anlatılan betimleyicilerin testi sırasında uygulanmıştır. İlgili bölümde, bu yöntem anlatılacaktır.

4.1.2 Deneyler ve Sonuçlar

1. Kullanılan deney seti

Bu çalışmada yapılan deneylerde, 5 farklı helikopter ve 10 farklı uçak sınıfı kullanılmıştır. Eğitim kümesi için, her bir sınıftan 122 ile 124 arasında değişen sayılarda silüet kullanılmıştır. Bu silüetler, araçların VRML modellerinden, yarı küre üzerinde düzenli aralıklarla yerleştirilmiş kamera pozisyonları kullanılarak elde edilmiştir. Toplam görüntü sayısı 1844'tür. Söz edilen sınıflara ait birer silüet Şekil 51'de ve Şekil 52'de verilmiştir.

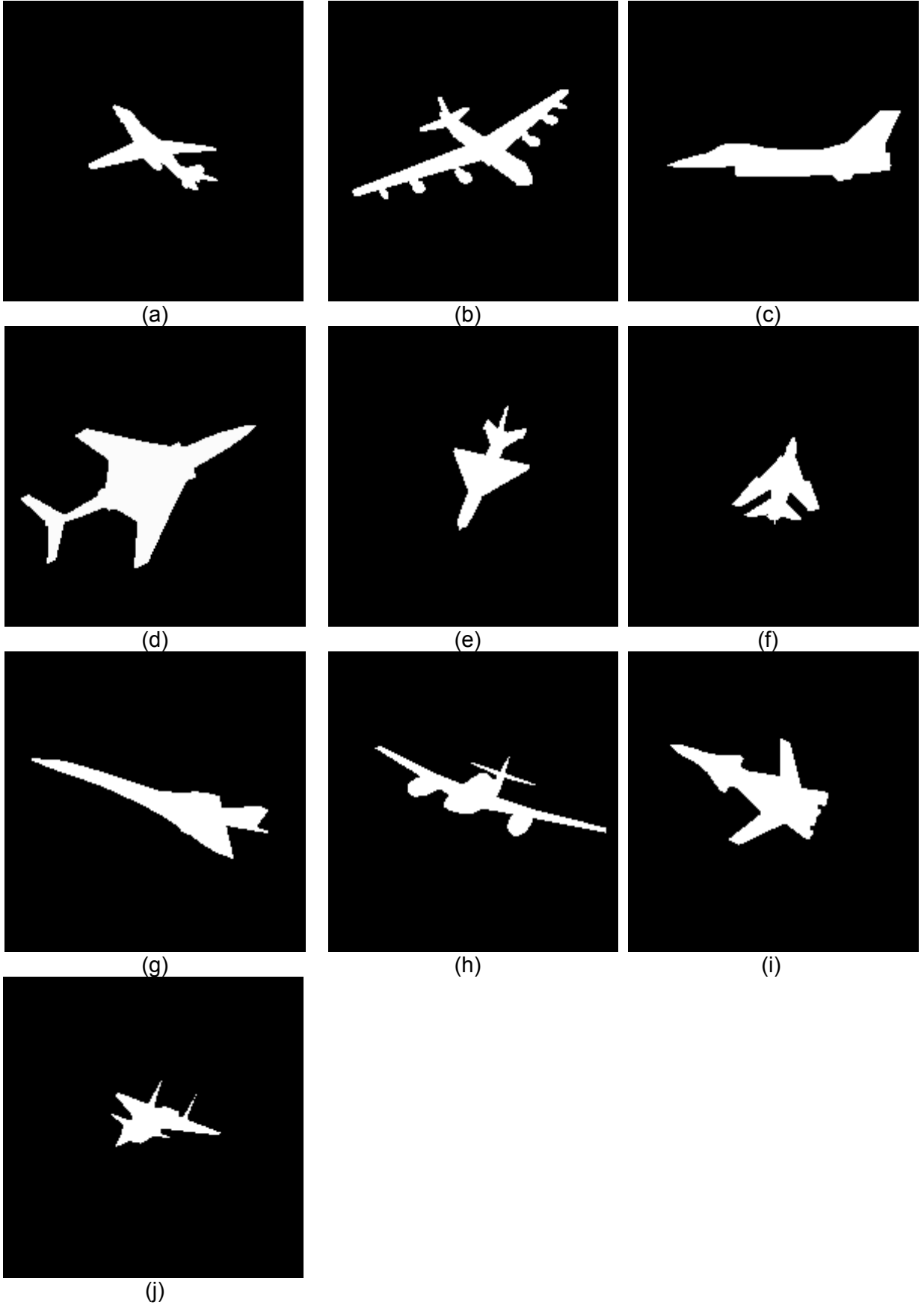


Şekil 51: Çalışmada kullanılan helikopter görüntüleri. Her sınıf için bir örnek verilmiştir.

Yöntemlerin denenmesi için uygulanan yöntem şöyledir: eğitim kümesindeki her bir görüntüye ait özellik vektörü, eğitim kümesinin diğer bütün elemanlarının özellik vektörüyle

karşılaştırılmış, kendisine *i) EYK* , *ii) k-EYK* sonucu ile gerçek sınıf bilgisinin örtüşüp örtüşmediğine bakılmıştır. Doğru ve hatalı sınıflandırmalar ilgili karıştırma matrislerine kaydedilmiştir.

Pratik uygulaması olduğu için, *EYK* türevi olan başka bir test de şu şekilde uygulanmıştır: Her test görüntüsüne en yakın *k* tane görüntünün ait olduğu sınıflara bakılmış, bu *k* sınıfın içinde bir tane bile doğru sınıf varsa, bu bir başarı sayılmıştır. Bu tür bir deney, (*k* küçük olduğu durumlarda) insanla etkileşen sistemlerde uygulaması olması açısından anlamlıdır. Bu test, metnin bundan sonraki bölümünde, “en az bir” anlamında, *EAB* olarak anılacaktır. *EAB* ölçütü kullanıcı etkileşimli sistemlerde, sistemin en az bir adet doğru sunabilmesi başarı olarak kabul etme varsayımı üzerine dayanmaktadır.



Şekil 52: Çalışmada kullanılan uçak görüntüleri. Her sınıf için bir örnek verilmiştir.

2. Deneysel Sonuçlar

Yapılan deneylerde elde edilen sonuçlar aşağıdaki karıştırma matrisleri ile verilmiştir. Karıştırma matrislerinin satırları gerçek sınıfı, sütunları ise sistemin kararını ifade etmektedir.

i. ARD Betimleyicisi

a. EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	113	0	4	1	0	1	0	0	0	0	0	0	0	2	1
Helikopter 1	0	112	3	0	0	2	0	1	0	2	0	0	0	2	0
Helikopter 2	0	9	111	1	0	1	0	0	0	0	1	0	0	0	0
Helikopter 3	1	0	1	116	1	0	0	0	0	1	0	0	1	0	1
Helikopter 4	0	0	1	0	121	0	0	0	0	0	0	0	0	0	0
Uçak 0	0	0	0	0	0	110	2	0	1	2	2	0	2	0	5
Uçak 1	0	1	0	0	0	2	115	2	1	0	0	1	0	1	0
Uçak 2	0	1	0	0	0	0	3	111	0	0	0	0	2	4	1
Uçak 3	0	0	0	0	0	1	0	0	113	0	1	2	3	0	2
Uçak 4	0	0	0	0	0	4	1	0	2	107	1	4	5	0	0
Uçak 5	0	1	0	0	0	0	0	0	1	3	118	1	0	0	0
Uçak 6	0	1	0	0	1	0	0	0	0	3	3	112	3	0	1
Uçak 7	0	1	0	0	0	1	1	0	3	3	0	5	106	2	2
Uçak 8	1	1	1	1	0	0	2	0	0	0	0	0	0	116	1
Uçak 9	0	1	0	1	0	3	1	1	1	0	2	2	1	1	109

b. k-EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	99	0	1	8	2	4	2	0	0	2	0	0	1	1	2
Helikopter 1	3	98	6	4	0	1	1	2	0	0	4	1	0	1	1
Helikopter 2	5	18	87	3	2	2	1	0	0	1	0	1	1	2	0
Helikopter 3	1	1	2	96	9	1	1	0	0	3	2	0	1	1	4
Helikopter 4	1	1	1	1	108	2	0	0	0	2	0	0	2	0	4
Uçak 0	1	1	0	6	1	99	2	1	2	2	6	2	1	0	0
Uçak 1	2	1	1	1	0	5	104	1	4	1	1	0	0	2	0
Uçak 2	0	1	1	4	0	1	2	84	2	0	4	1	3	17	2
Uçak 3	0	1	0	1	0	4	0	1	86	3	2	5	13	1	5
Uçak 4	0	0	1	1	0	3	2	0	1	101	1	8	6	0	0
Uçak 5	0	1	0	0	0	2	1	2	3	8	97	5	3	2	0
Uçak 6	0	0	1	0	2	3	0	0	0	7	6	99	6	0	0
Uçak 7	0	3	0	2	1	1	4	1	4	10	7	9	77	1	4
Uçak 8	0	1	1	4	0	2	2	12	2	1	5	1	4	88	0

Uçak 9	0	2	0	1	0	7	1	1	3	3	6	4	5	6	84
--------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

c. EAB

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Doğru	119	117	120	119	122	121	122	121	117	122	120	121	118	122	119
Toplam	122	122	123	122	122	124	123	122	122	124	124	124	124	123	123

ii. Hu Momentleri

a. EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	113	0	0	2	0	1	1	4	0	0	0	0	1	0	0
Helikopter 1	0	115	2	0	1	1	0	0	0	1	0	0	0	2	0
Helikopter 2	4	19	74	2	9	1	0	7	2	0	0	1	2	1	1
Helikopter 3	2	2	0	107	2	2	2	2	0	2	0	0	1	0	0
Helikopter 4	0	1	3	0	114	3	0	1	0	0	0	0	0	0	0
Uçak 0	4	1	1	3	2	90	1	3	1	6	4	0	5	2	1
Uçak 1	1	0	0	0	0	0	121	0	0	1	0	0	0	0	0
Uçak 2	2	3	0	1	0	1	1	113	0	1	0	0	0	0	0
Uçak 3	0	0	0	0	0	1	0	0	118	0	0	0	2	0	1
Uçak 4	1	1	0	1	0	2	2	0	0	86	8	6	9	6	2
Uçak 5	0	0	0	0	0	1	0	0	0	5	117	0	1	0	0
Uçak 6	1	0	0	0	0	2	0	0	2	10	1	104	2	0	2
Uçak 7	1	2	1	2	1	8	2	1	6	8	5	9	53	5	20
Uçak 8	1	0	0	0	0	0	0	0	0	1	0	0	0	120	1
Uçak 9	0	0	1	2	0	1	0	0	1	1	1	1	4	0	111

b. k-EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	46	0	4	17	10	12	13	14	0	0	2	0	0	4	0
Helikopter 1	7	63	16	12	4	2	2	7	4	2	3	0	0	0	0
Helikopter 2	10	28	49	6	6	2	2	5	2	2	1	4	2	3	1
Helikopter 3	19	11	4	62	9	3	4	6	0	1	1	0	0	2	0
Helikopter 4	13	3	6	9	75	7	2	3	1	1	0	0	0	2	0
Uçak 0	11	1	1	5	2	56	4	3	2	18	11	0	2	7	1
Uçak 1	17	0	5	6	0	4	71	14	0	0	0	0	2	2	2
Uçak 2	12	6	11	13	10	3	17	34	6	0	0	4	0	6	0
Uçak 3	3	0	2	4	6	3	3	6	72	4	0	6	4	2	7
Uçak 4	3	2	0	4	0	7	0	0	2	77	17	5	2	4	1
Uçak 5	6	3	0	0	0	10	0	2	2	13	78	2	2	6	0
Uçak 6	1	2	4	0	0	8	0	2	2	16	14	63	5	2	5
Uçak 7	1	2	3	4	2	12	3	0	7	9	9	17	35	4	16
Uçak 8	13	4	0	7	4	9	4	7	6	3	14	7	1	44	0
Uçak 9	4	2	3	6	1	9	0	0	8	8	8	12	6	1	55

c. EAB

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Doğru	117	119	109	112	121	109	121	116	119	99	119	114	85	123	117
Toplam	122	122	123	122	122	124	123	122	122	124	124	124	124	123	123

iii. EÖU Betimleyicisi

a. EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	69	4	1	15	21	1	3	3	0	1	0	0	2	2	0
Helikopter 1	9	39	14	8	15	3	7	10	2	2	2	1	5	5	0
Helikopter 2	5	18	27	11	30	5	4	11	0	0	2	2	2	6	0
Helikopter 3	17	11	11	53	15	0	2	8	0	1	0	1	0	3	0
Helikopter 4	20	6	22	7	57	0	2	5	0	0	0	0	0	3	0
Uçak 0	1	1	6	0	1	59	0	1	10	8	4	10	11	5	7
Uçak 1	6	13	10	6	4	5	31	3	2	2	1	5	1	34	0
Uçak 2	3	14	8	12	11	4	8	51	1	1	1	0	0	6	2
Uçak 3	1	1	2	0	0	18	10	2	55	5	8	8	8	4	0
Uçak 4	0	1	1	1	0	7	1	0	12	69	4	8	8	8	4
Uçak 5	0	1	2	1	0	5	0	2	10	19	42	17	16	4	5
Uçak 6	0	0	1	1	0	15	2	1	6	9	15	44	14	8	8
Uçak 7	0	2	3	0	1	15	0	1	6	11	5	10	50	14	6
Uçak 8	4	4	3	2	1	7	14	3	3	4	6	5	4	62	1
Uçak 9	0	0	2	0	1	8	0	0	5	9	10	8	13	5	62

b. k-EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	72	3	2	21	18	0	2	3	0	1	0	0	0	0	0
Helikopter 1	15	42	17	7	11	2	6	9	2	3	3	0	0	5	0
Helikopter 2	9	34	18	10	22	3	6	14	1	1	2	1	0	2	0
Helikopter 3	28	17	6	49	11	1	2	6	0	0	0	0	0	1	1
Helikopter 4	19	10	13	13	59	0	1	6	0	0	0	0	0	1	0
Uçak 0	1	3	5	0	1	67	2	4	14	8	0	9	6	2	2
Uçak 1	14	11	9	11	2	3	28	5	0	1	1	1	2	35	0
Uçak 2	10	23	11	13	7	5	7	36	0	0	1	0	1	7	1
Uçak 3	4	5	4	1	0	22	5	3	59	11	3	4	0	1	0
Uçak 4	1	0	0	3	0	15	2	0	16	59	6	6	11	3	2
Uçak 5	0	2	4	1	0	10	1	0	15	20	40	15	12	2	2
Uçak 6	0	0	5	3	0	25	4	1	2	10	24	40	2	6	2
Uçak 7	2	3	4	2	0	19	1	0	9	15	10	10	42	5	2
Uçak 8	10	10	6	6	0	3	16	5	3	1	6	6	7	44	0
Uçak 9	4	1	1	1	0	12	1	3	13	10	7	13	11	3	43

c. EAB

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4										
						Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Doğru	103	81	76	100	105	100	73	90	101	108	83	102	91	107	90
Toplam	122	122	123	122	122	124	123	122	122	124	124	124	124	123	123

iv. Fourier Betimleyicisi

a. EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4										
						Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	17	3	9	5	7	17	1	11	8	8	10	8	5	11	2
Helikopter 1	4	32	0	0	0	7	17	11	9	3	13	6	0	8	12
Helikopter 2	8	0	31	20	7	16	0	1	5	6	2	7	14	4	2
Helikopter 3	5	0	12	28	8	13	1	2	7	8	5	8	13	7	5
Helikopter 4	6	0	11	7	53	5	1	1	15	5	1	5	12	0	0
Uçak 0	18	8	17	11	3	10	10	0	6	6	2	6	9	10	8
Uçak 1	1	11	0	0	0	6	59	2	7	6	18	5	0	6	2
Uçak 2	7	7	2	2	0	4	4	25	3	9	17	15	1	9	17
Uçak 3	5	10	6	6	11	5	7	3	21	7	3	6	19	8	5
Uçak 4	11	4	7	12	6	9	4	13	6	8	6	12	12	6	8
Uçak 5	11	10	2	6	1	3	14	10	8	6	18	10	1	9	15
Uçak 6	8	6	5	11	5	4	5	18	7	8	12	2	10	11	12
Uçak 7	6	0	15	17	11	10	0	2	11	9	2	12	23	4	2
Uçak 8	10	14	1	8	0	5	4	12	9	9	9	12	2	13	15
Uçak 9	6	16	3	4	0	7	4	16	9	3	13	15	1	9	17

b. k-EYK

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Helikopter 0	28	7	17	7	0	8	3	13	5	5	6	5	4	9	5
Helikopter 1	7	60	0	2	0	3	21	2	0	1	10	1	0	4	11
Helikopter 2	23	1	52	10	5	6	0	3	3	4	2	4	7	3	0
Helikopter 3	26	2	22	28	7	5	1	2	2	6	5	4	9	2	1
Helikopter 4	8	0	14	13	65	3	1	1	5	3	1	1	7	0	0
Uçak 0	27	14	22	4	3	9	8	5	4	7	1	1	5	7	7
Uçak 1	1	19	0	0	0	6	82	2	1	0	6	0	0	0	6
Uçak 2	23	17	3	3	2	6	4	30	1	2	9	1	0	7	14
Uçak 3	6	14	10	10	12	3	12	4	9	7	2	4	20	7	2
Uçak 4	23	7	8	12	6	6	7	9	2	11	7	8	6	5	7
Uçak 5	14	29	3	5	1	4	19	20	0	2	8	5	0	5	9
Uçak 6	25	8	10	11	2	7	8	15	5	8	12	0	3	6	4
Uçak 7	18	2	21	10	13	8	0	2	18	7	3	2	16	2	2
Uçak 8	23	15	6	7	0	11	8	15	4	3	8	2	1	8	12
Uçak 9	16	27	2	2	0	5	10	19	2	3	11	3	1	7	15

c. EAB

	Helikopter 0	Helikopter 1	Helikopter 2	Helikopter 3	Helikopter 4	Uçak 0	Uçak 1	Uçak 2	Uçak 3	Uçak 4	Uçak 5	Uçak 6	Uçak 7	Uçak 8	Uçak 9
Doğru	55	86	85	69	86	46	111	71	51	49	61	25	68	50	67
Toplam	122	122	123	122	122	124	123	122	122	124	124	124	124	123	123

3. Sonuç

DeneySEL sonuçlardan görüleceği gibi, Hu betimleyicileri gürültü içermeyen ve yalnızca kendi çekim açısının eğitim setinde bulunmaması senaryosu için en başarılı sonucu vermekte, ARD onu takip etmektedir. İki yöntemin de bölgesel özellikler üzerine kurulu olması dikkat çekicidir. Öte yandan, Fourier betimleyicisinin kabul edilebilir sonuçlar vermediği gözlenmiştir. Yine de, EÖU ile, uçaktan helikopteri ayırt etme problemi için, %80'in üzerinde bir başarı sağlanabilmektedir.

Bu deneylerde yöntemleri başarısızlığa uğratan en önemli neden, iki boyutlu cisimlerin değil, araçların üç boyutlu uzayda farklı kamera konumlarından alınmış görüntülerinin kullanılması, ve bir aracın bütün açılardan alınan görüntülerinin aynı sınıfa dahil sayılmasıdır. Bu nedenle, farklı araçların benzer açılardan alınmış görüntüleri, aynı aracın farklı açılardan alınmış görüntülerinden daha fazla benzerlik gösterebilmektedir. k-NN yönteminin NN yöntemine göre daha az başarılı olmasının altında bu sonuç yatmaktadır.

4.2. GÖRÜNTÜLERDEN YAZI BULMA VE TANIMA

Sayısal görüntülerdeki en önemli bilgi kaynaklarından birisi görüntü içerisinde bulunan yazılardır. Bu yazıların otomatik olarak bulunması ve tanınması görüntü veritabanlarının oluşturulmasını ve endekslenmesini sağlamaktadırlar. Bu tip yazılar, görüntünün bir parçası olarak karşımıza çıkabileceği gibi görüntüye daha sonradan eklenmiş de olabilmektedirler: daha sonradan eklenmiş, yapay yazılar (*overlay text*) ve sahnenin içerisinde bulunan yazılar (*scene text*). Sonradan eklenmiş, yapay yazılara, çeşitli görüntülerde karşılaşılan başlık, alt yazı, maç sonucu vb. örnek olarak verilebilmektedir. Sahne içerisindeki yazılara örnek olarak da, görüntünün içerisinde bulunan reklam panoları, araç plakaları, kamyon ve otobüslerin üzerindeki yazılar vb. gösterilebilir.

4.2.1 Görüntülerden Yazı Bulma

Sayısal görüntülerde karmaşık bir sahne içinde yazıların yerini belirleme problemi, bu yazıların tanınma problemini ilk basamağıdır. İnsanların bilmedikleri bir alfabe için bile yazıları kendine has özellikleri nedeniyle farkedebilmeleri, yazı bulma probleminin kendine özel bir yanı olduğu ve alfabeden bağımsız olarak bulunabilmesinin mümkün olduğunu düşündürmektedir.

4.2.1.1 Literatürdeki Benzer Çalışmalar

Literatürde bulunan görüntü içerisindeki yazıların bulunması ve tanınması ile ilgili çalışmalar, problemin alanına, problemin çözümünde kullandıkları temel yöntemlere göre sınıflandırılabilirler. Problem alanına göre – trafik levhalarında [1],[2]; araçlar üzerinde [3] ve plakalarında [4],[5],[6]; internet üzerinde bulunan resimlerde [7],[8] ve video görüntülerinde [9],[11]; yazının bulunması ve tanınması üzerine çalışmalar yapılmıştır. Bütün bu çalışmalarda kullanılan metodlar, her ne kadar bir biri ile yakın ilişkili olsa da, temel olarak iki grupta toplanabilir. Bunlar Bağıntılı Bileşen (*Connected Component*) ve Doku (*Texture*) yaklaşımlarıdır.

Bağıntılı Bileşen Yaklaşımı:

Bağıntılı Bileşen yaklaşımında aynı renge sahip kapalı alanlar ayrıştırılarak harflere ulaşmak hedeflenmektedir. Bu sebeple, bu yaklaşım, yüksek performans değerleri için yüksek çözünürlüğe ve yine yüksek kontrastlı alanlara ihtiyaç duymaktadır. Doğal bir sonuç olarak da düşük kontrast bulunan durumlarda hatalı bölütleme sonuçları görülmektedir [13],[14].

Zhong, Karu ve Jain'in [15] önerdikleri metotta, renk uzayının histogramdaki tepe noktaları kullanılarak nicemlenmesi kullanılmaktadır. Bu metotta komşu renkler histogramdaki bu tepe noktaları ile birleştirilmektedir. Burada yazının, renk uzayın dar

bir alanı kapsadığı varsayımı yapılmaktadır. Daha sonra renk bileşenleri bazı geometrik özellikler kullanılarak yazı bölgesi olarak sınıflandırılmaktadır. Doğal olarak bu yaklaşımda, eğer renk nicemleme adımında bir problem ile karşılaşır ve nicemleme de hata olursa (bu durum düşük kontrast ya da yazının küçük bir alan kaplaması durumlarında rahatlıkla karşılaşılabilmektedir), sınıflandırmada hatalar oluşmaktadır. Kim [16] ise renk topak analizi ve bunu takip eden bir şekil analizi ile yazı bölgelerini belirlemeye çalışmaktadır. Bu metodun en temel sakıncası ise, deneysel olarak bulunan bir çok eşik değerinin performansı fazlasıyla etkilemesidir.

Lienhart ve Stuber'in çalışmalarında [11],[12]; filmlerin başlangıç ve bitiş görüntülerinde bulunan jenerik ve adlar dizinindeki yazıların bulunması hedef alınmıştır. Önerdikleri metodda hem uzamsal hem de zamansal olarak renk bölütlemesi yapılmaktadır. Her ne kadar sağlam kuramsal temelleri olan bir çalışma olsa da, yazı alanlarını tek bir bağıntılı bileşen olarak düşünmeleri ve güçlü kontrasta sahip olmayan yazıların olduğu, üzerinde gürültü ve ya fazlasıyla karmaşık bir arka plana sahip olan durumlarda yeterli performans değerine ulaşamıyor olması genel görüntülerde uygulanması için yeterli derecede gürbüz olamamasına neden olmaktadır. Zhou, Lopresiti ve Tasdizen'in çalışmasında [8] ise yine renk bölütlemesi kullanılmaktadır. Bu araştırmada renk bölütlemesi için minimum kapsayan ağaç tekniği (Minimum Spanning Tree) kullanılmıştır ve renkler arasındaki uzamsal uzaklık bölütleme sürecinde dikkate alınmıştır.

Literatürdeki bazı çalışmalarda video görüntülerinden elde edilen birden fazla kare üzerinde bağıntılı bileşen metodu uygulanmaktadır. Böyle bir çalışmaya örnek olarak Ariki ve Teranishi'nin çalışması [17] gösterilebilir. Kullandıkları metotta, yazının arka plana göre daha parlak olduğu gibi basit varsayımlardan yola çıkılmaktadır ve video karelerinin farkı alınarak yazılı alanlar bulunmaktadır. Kurakake, Kuwano ve Odaka [18] ise video karelerini birbiriyle alt bölgelere ayırıp daha sonra her bir bölgede renk histogramı farkı uygulanmaktadır. Eğer birbirini takip eden iki karedeki bölgeler arasında önemli bir değişiklik fark edilirse, sonraki karede yazı olduğu kabul edilmektedir. Daha sonraki adımda da bu bölgede bazı uzamsal metodlar uygulanarak yazılı alanlar, gerçekten varsa, bulunuyor.

Dokuya Dayalı Yaklaşım:

Görüntülerde yazının bulunmasında doku yaklaşımı, yazının aslında farklı bir doku sınıfı olduğunu varsaymaktadır. Bir çok farklı dokuya dayalı sınıflandırma yöntemleri arasındaki ana farklılık dokunun gösterim şeklidir. Literatürdeki, farklı modeller arasından değişik yönelimdeki Gabor süzgeçleri [19], belirli yönler için basit ayırıt sezimi [20] ya da dalgacık dönüşümü [22] gibi metodlar dokuları tanımlamak için kullanılmaktadır.

Chaddha, Sharma, Agrawal ve Gupta [23] JPEG ile sıkıştırılmış imgelerde yazının bulunması için frekans bileşenlerini doku tanımlayıcıları olarak kullanmaktadır. Metodlarında öncelikli olarak JPEG imgenin macro blokları içerisindeki her bir bloğun DCT katsayıları bulunmaktadır. DCT katsayıların mutlak değerleri toplanarak elde edilen ölçü yüksek frekanslı alanları bir diğer deyişle yazılı alanları tanımlamakta kullanılmaktadır. Daha sonra deneysel olarak elde edilen bir eşik değeri ile bir bloğun yazı olup olmadığına karar verilmektedir.

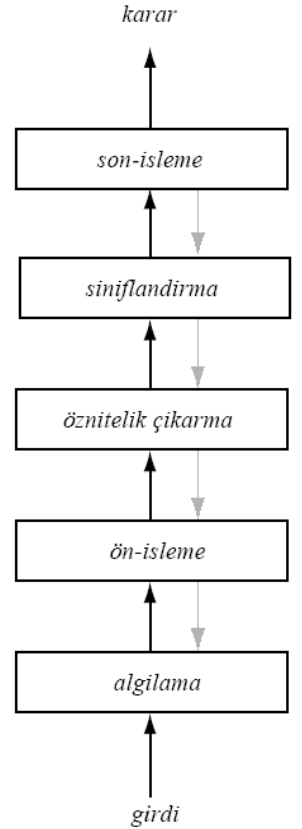
Bir başka önemli doku tanımlayıcısı olarak da blokların frekans içeriği ile ilgili bilgi taşıyan dalgacık dönüşümünden elde edilen katsayılar kullanılmaktadır. Li ve Doermann [9],[10] video görüntülerindeki referans karelerinde yazı bulmaktadırlar. Çalışmalarında dalgacık dönüşümü ile öznitelik vektörü bulunmakta ve daha sonrada sinir ağı ile sınıflandırma yapılmaktadır.

Hauptmann ve Smith [24] ise video üzerinde bulunan dikey ve yatay ayrıtların yoğunluklarına göre yazılı alanları ayırt etmeye çalışmaktadırlar. Wu, Manmatha, ve Risemann [22] ise doku bölütmesi kullanarak yazılı alanları bulmaktadır. Bunun için ayırt sezimi karakter çomaklarını bulmak için kullanılmakta daha sonra da bulunan bu karakter çomakları birleştirilerek yazılı alanlar oluşturulmaktadır. Ancak bu yöntem ile düşük çözünürlüklü ve dış çekim görüntülerde çok fazla yanlış sonuçla karşılaşmaktadır.

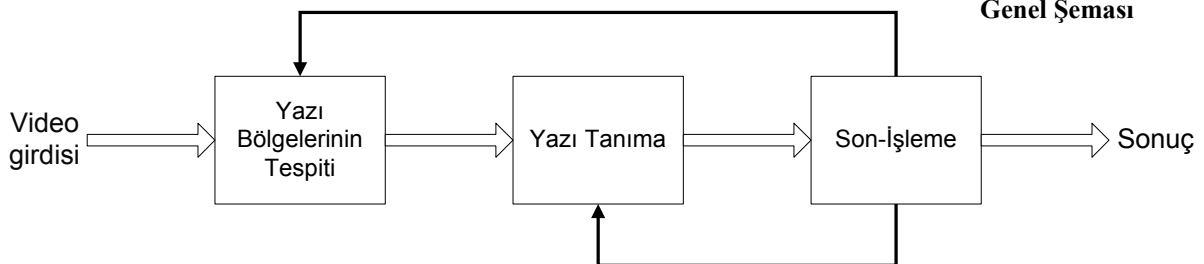
4.2.1.2 Önerilen Yaklaşım:

Her türlü yakalanmış görüntüde yazı bulma problemi, genel bir örüntü tanıma probleminin özel bir uyarlaması olarak görülebilir. Buna göre çözüm Şekil 54’de gösterilen blok çizeneğindeki adımlardan oluşacaktır. Bu adımlar sırası ile “Algılama”, “Ön-İşleme”, “Öznitelik Çıkarma”, “Sınıflandırma” ve “Son-İşleme”dir. “Algılama” adımında görüntünün yakalanması sağlanmaktadır, “Öznitelik Çıkarma” adımında yazı olan alanların sahip olduğu belirleyici özelliklerin çıkartılma işlemi gerçekleştirilmektedir. “Sınıflandırma” adımı ile de bu elde edilen özellikler kullanılarak sınıflandırılacak olan vektörün hangi sınıfa – yazı ya da yazı olmayan – ait olduğu belirlenmektedir. Son olarak “Son-İşleme” adımında belirlenen yazı bölgeleri sınırlayıcı kutular haline getirilip gerekli imge işleme metodları ile – eşikleme vs. – Optik Karakter Tanıma tarafından işlenecek hale getirilmektedir. Böylelikle sistemin sonucu olarak ASCII biçiminde yazı, yakalanan görüntülerden elde edilmektedir.

Problem için önerilen çözüm yöntemi Şekil 55’de gösterilmiştir. Önerilen bu sistemde öncelikli olarak daha önce bahsedilen genel örüntü tanıma metodu kullanılarak yazı bölgelerinin tespiti yapılmaktadır. Bunun için farklı öznitelik vektörleri çıkartma yöntemleri ile yine farklı sınıflandırma yöntemleri kullanılmaya çalışılmıştır, yazı bölgelerinin tespiti ile ilgili daha ayrıntılı bilgi ilerleyen bölümlerde verilmektedir. Daha sonra elde edilen yazı bölgelerinde yazı tanıma sistemi kullanılarak ASCII biçiminde yazı elde edilmektedir. En son olarak ASCII biçimindeki yazı üzerinde kullanılacak çeşitli son işleme metodları ile genel sistemin başarımını arttırmak amaçlanmaktadır.



Şekil 53 : Örüntü Tanıma Genel Şeması



Şekil 54: Önerilen Çözüm Yöntemi

Çözüm adımlarının açıklanmasından önce, elde edilen performansın değerlendirilmesi şu şekilde özetlenebilir.

4.2.1.3 Görüntü içinde Yazı Bulma Problemi için Performans Analizi :

Görüntü içindeki yazıların tespiti çalışmalarında önce ilk yapılması gereken sonuçların performanslarını ölçülmesi için gerekli kriterlerin belirlenmesi olacaktır. Yakalanmış video görüntülerinde metin tanıma probleminde kullanılacak performans analizleri iki farklı şekilde yapılmıştır. Bunlardan ilki blokların ne kadar doğruluk oranları ile sınıflandırıldığı ve ikincisi de sınırlama kutularının ne kadar doğrulukla bulduklarıdır.

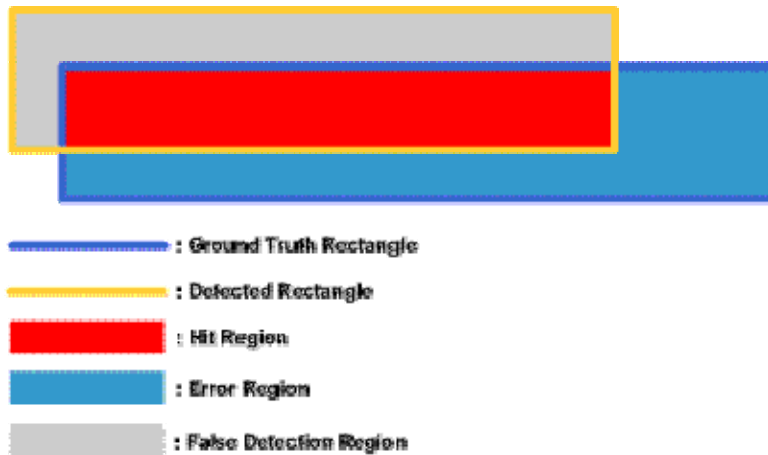
4.2.1.3.1 Sınırlama Kutusu Performans Analizi:

Bu değerlendirme metodunda [32] üç performans kriteri kullanılmaktadır; *Doğruluk Oranı*, *Hata Oranı*, *Yanlış Bulma Oranı*. Performans analizi için gerekli olan alanlar Şekil 56'da detaylı olarak gösterilmektedir; burada "Ground Truth Rectangle" gerçekte resim üzerinde bulunan yazı bölgelerini "Detected Rectangle" ise yazı bulma algoritmalarının sonuçlarını göstermektedir.

1. Doğruluk Oranı:

Doğruluk Oranı "Hit Region" olarak gösterilen bölgenin alanının otomatik olarak bulunan yazı alanına karşılık gelen "Ground Truth Rectangle" alanına oranıdır. Diğer bir deyişle, eğer d otomatik olarak bulunan bölge, g gerçekte olması gereken alan ve sona olarak da o bu iki bölgenin kesişimi ise Doğruluk Oranı (HR);

$$HR = \frac{\sum_{g \in T_g} \frac{A(o)}{A(g)}}{N}$$



Şekil 55 : Performans analizi için tanımlanan bölgeler : (Mavi çizgi) Yazının gerçek sınırları, (Sarı çizgi) Yazının tespit edilen sınırları, (Kırmızı bölge) Doğru tespit edilen alan, (Mavi bölge) Tespit edilemeyen alan, (Gri bölge) Yanlış tespit edilen alan

burada T_g gerçekte yazı bölgelerinin kümesi, N ise hakiki yazı alanlarının toplam sayısı ve $A(\bullet)$ da alan operatörüdür.

2. Hata Oranı:

Hata Oranı "Error Region" olarak gösterilen bölge yani otomatik olarak bulunamayan bölge ile bu alanla eşleşen hakiki yazı alanına oranıdır. Daha uygun bir şekilde Hata Oranı (ER);

$$ER = \frac{\sum_{g \in T_g} \frac{A(g-o)}{A(g)}}{N}$$

kullanılarak hesaplanır.

3. Yanlış Bulma Oranı:

Son olarak Yanlış Bulma Oranı (FAR) "False Detection Region" olarak gösterilen yanlış saptanmış alan ile araştırılan toplam alana oranıdır.

$$FAR = \frac{\sum_{d \in T_d} A(d-o)}{A(I)}$$

kullanılarak hesaplanır.

4.2.1.4 Yazı Bölgesi Bulma Algoritması:

Önerilen yazı bölgesi bulma algoritması temel olarak doku temelli yaklaşımlardandır. Bu amaçla farklı doku öznitelikleri ve sınıflandırıcılar kullanılarak yazı bölgelerinin bulunma performansın sınanacaktır.

4.2.1.4.1 Dokusal Öznitelik Çıkarma:

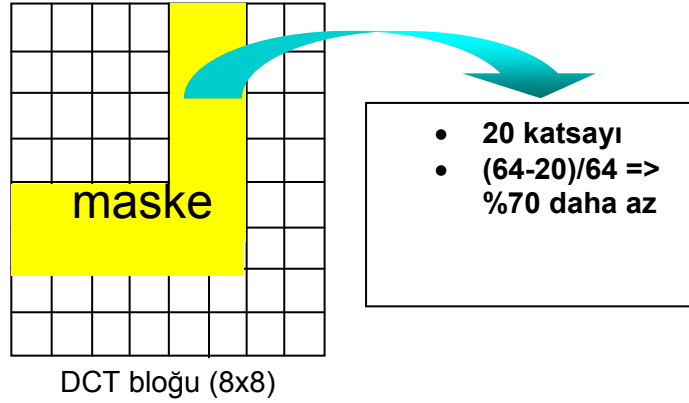
Dokusal öznitelikler olarak ayırık kosinüs dönüşümü, Haar dalgacık dönüşümü, homojen doku tanımlayıcısı ve ayırık histogram tanımlayıcısı kullanılacaktır.

4.2.1.4.1.1 Ayırık Kosinüs Dönüşümü (Discrete Cosine Transform):

Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform) imge kodlama sistemleri olarak bilinen dönüşüm kodlamaları arasında yaygın olarak kullanılan bir dönüşümdür. Bu yitimli dönüşüm metodu sınıflandırma bloğunu beslemek için gerekli öznitelik vektörlerini çıkarmak için kullanılacaktır. Bu amaçla imgeden alınan her bir 8×8 bloğun DCT katsayıları hesaplanıp doğrudan öznitelik vektörü olarak kullanılacaktır.

Burada dikkat edilmesi gereken bir nokta DCT tanım kümesinde, ani değişiklikler büyük indeksli katsayılara denk gelmektedir. Böylelikle, bu düşünceden yararlanılarak, çoğu sonradan eklenmiş yazının bu ani değişiklikleri içerisinde barındırdığını düşünecek olursak sistemin performansında önemli iyileştirmelere gidebilmek mümkün olmaktadır. Sonuç olarak

yalnızca orta ve yüksek indeksli katsayılar yazı bölgelerini nitelendirmek için yeterli ve yararlı olacaktır. Bu noktadan çıkarak daha iyi nitelendirmek ve performansı arttırmak için bir dizi deneyler yapılmıştır. Bu deneylerde 50 imgeden elde edilen DCT katsayıları çeşitli maskeler kullanılarak süzülmüştür ve daha sonra alınan ters dönüşüm incelenmiştir. Buradaki amaç yazılı alanları en az sayıda katsayı ile en iyi şekilde nitelendirebilmektir. Bu çalışma sonucunda Şekil 57'deki maskenin uygunluğuna karar verilmiştir.



Şekil 56: Ayrık Kosinüs Dönüşümü Öznitelik Maskesi

Şekil 58'de bu kavram örnek imge üzerinde gösterilmektedir. Maske kullanılarak hesaplanan 20 DCT katsayısı kullanılarak elde edilen ters dönüşümde yazılı bölgeler yeterince belli olmaktadır. Böylelikle 64 boyutlu öznitelik vektörü yerine 20 boyutlu vektör kullanılması sağlanmış olmaktadır.



Şekil 57: Örnek İmge ve 20 katsayıdan elde edilen ters dönüşüm.

4.2.1.4.1.2 Haar Dalgacık Dönüşümü (Haar Wavelet Transform):

Dalgacık dönüşümünün, örnek seyreltme ile ardışık yaklaşıklamalar sağlanması ve ayrışmaları ortaya çıkarma yeteneğinin bulunması kullanılma nedenlerinden birkaçıdır. Farklı uygulamalarda problemin durumuna göre farklı dalgacık dönüşümleri kullanılabilir. Bizim durumumuzda Haar dalgacığı kullanılacaktır; bundaki sebeplerden en önemlisi yazı bölgelerini temsil eden çizgi parçalarını bulmakta yeterli olmasıdır. Ayrıca dönüşümün, basit bir maske işlemi ile elde edilebiliyor olması, Haar dalgacık dönüşümünü hesaplama karmaşıklığı olarak da verimli olmasını sağlıyor.

Ölçekleme ve dalgacık işlevi sırasıyla aşağıda belirtilmiştir;

$$\phi(x) = \sum_{k \in Z} p_k \phi(2x - k) = \phi(2x) + \phi(2x - 1) \quad (1)$$

$$W_H(x) = \sum_{k \in Z} q_k \phi(2x - k) = \phi(2x) - \phi(2x - 1) \quad (2)$$

with,

$$\phi(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$I(x, y)$ imgesi için;

$$I(x, y) = \begin{bmatrix} i_{0,0} & i_{0,1} & \cdots & i_{0,2N-1} \\ i_{1,0} & i_{1,1} & \cdots & i_{1,2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ i_{2N-1,0} & i_{2N-1,1} & \cdots & i_{2N-1,2N-1} \end{bmatrix}_{2N \times 2N} \quad (4)$$

İki boyutlu Haar Dalgacık dönüşümünden ortaya çıkan seviyeler aşağıdaki denklemlerle kolaylıkla elde edilmektedir;

$$LL_{x,y} = \frac{1}{4} (i_{2x,2y} + i_{2x,2y+1} + i_{2x+1,2y} + i_{2x+1,2y+1}) \quad (5)$$

$$LH_{x,y} = \frac{1}{4} (i_{2x,2y} - i_{2x,2y+1} + i_{2x+1,2y} - i_{2x+1,2y+1}) \quad (6)$$

$$HL_{x,y} = \frac{1}{4} (i_{2x,2y} + i_{2x,2y+1} - i_{2x+1,2y} - i_{2x+1,2y+1}) \quad (7)$$

$$HH_{x,y} = \frac{1}{4} (i_{2x,2y} - i_{2x,2y+1} - i_{2x+1,2y} + i_{2x+1,2y+1}) \quad (8)$$

Burada dikkat edilecek en önemli nokta bu dönüşüm ile elde edilen hesaplama verimi,kazanımıdır. Bu elde edilen dönüşüm kullanılarak öznitelik vektörü; blokların ortalama değeri,değişintileri ve 3. merkezi momentleri bulunarak oluşturulmaktadır.

$$M(\hat{I}) = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \hat{I}(i, j) \quad (9)$$

$$\mu_2(\hat{I}) = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} (\hat{I}(i, j) - M(I))^2 \quad (10)$$

$$\mu_3(\hat{I}) = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} (\hat{I}(i, j) - M(I))^3 \quad (11)$$

Öznitelik vektörünün bileşenleri, dönüşümden elde edilen bütün bantlarda hesaplanan değerlerden oluşmaktadır. Burada blok 8×8 olarak alındığı için dalgacık dönüşümü en fazla 2 seviye olmaktadır. Böylelikle herhangi bir değişiklik yapmadan bu öznitelikler hesaplama yollarıyla 24 boyutlu bir vektör elde edilmektedir. Fakat bazı bantlarda yazıyı tanımlayan daha fazla bilgi olduğu göz önüne alındığında bu 24 boyutlu vektör rahatlıkla 12 boyuta indirilebilmektedir.

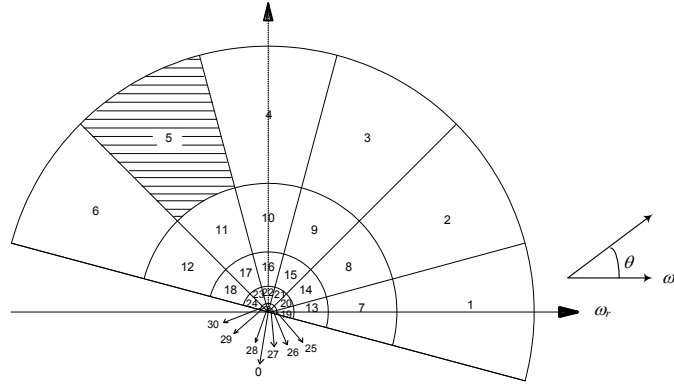
4.2.1.4.1.3 Homojen Doku Tanımlayıcısı (Homogeneous Texture Descriptor -HTD):

Şu an kullanılmakta olan MPEG-7 doku tanımlayıcılarından homogeneous texture descriptor imgenin, ortalama değer ve standart sapmasını ve imgenin Fourier dönüşümünün enerji ve enerji değişiminden oluşmaktadır. Yapılan çalışmalar bu tanımlayıcı kullanılarak bulunan imge sorgularının daha başarılı olduğunu ortaya koymuştur. Tanımlayıcının elde edilme metodu – kutupsal koordinatlar vs. – ve ayrıca doku özelliklerini insan görme duyusuna benzer bir şekilde tanımlaması, yazı tanıma probleminde öznitelik vektörü olarak kullanılabilirliğini düşündürmektedir. Bundaki sebeplerden birisi yazının okunabilir ve algılanabilir olabilmesi için belirli bir doku özelliği taşıması gerektirmesidir.

HTD ile doku özniteliklerinin çıkarılması için öncelikli olarak imgenin Radon dönüşümü ve bunu izleyen tek boyulu Fourier dönüşümü uygulanmaktadır. Daha sonra iki boyutlu frekans bölgesinde $F(\omega, \theta)$ 'dan merkezi dilimler alınmaktadır. Bu noktada doku tanımlayıcısı Şekil 59'de gösterilmekte olan her bir banttandır elde edilen değerlerden oluşmaktadır. Öznitelik vektörü de bu bantlardan elde edilen enerji ve enerji değişimi değerlerini içermektedir.

$$TD = [f_{DC}, f_{SD}, e_1, e_2, \dots, e_{30}, d_1, d_2, \dots, d_{30}]$$

Burada $[e_1, e_2, \dots, e_{30}]$ her bir banttaki enerji değerlerini ve $[d_1, d_2, \dots, d_{30}]$ ise bu bantlardaki enerji değişimi değerlerine karşılık gelmektedir. Elde edilen bu 62 boyutlu vektör öznitelik vektörü olarak kullanılmaktadır.



Şekil 58: HTD Bantları

4.2.1.4.1.4 Ayrıt Histogram Tanımlayıcısı (Edge Histogram Descriptor):

Bir diğer MPEG-7 tanımlayıcılarından olan edge histogram tanımlayıcısı bir imgede bulunan ayrıt çeşitlerinin dağılımını sunmaktadır. Sonradan eklenen yazının belirli ayrıtlardan oluşuyor olması bu MPEG-7 tanımlayıcısının etkin sonuçlar doğurabileceğini düşündürmektedir.

Şekil 59'da görülen 5 farklı ayrıtın imge – blok – üzerindeki dağılımları bulunarak öznitelik vektörü elde edilmektedir. Tahmin edileceği üzere öznitelik vektörünün elde edilmesi basit

çarpma ve toplama işleminden ibaret olduğu için hesaplama süresi hızlıdır. Ancak yapılan çalışmalarda bu öznelik vektörünün fazlasıyla genel bir sonuç çıkarmasından dolayı sonuçlar yetersiz görünmektedir.



Şekil 59: Ayırt Haritaları

4.2.1.4.2 Sınıflandırıcılar:

Öznelik çıkarma adımından sonra gelen adım sınıflandırma adımı olmaktadır. Bu kısımda da birbirinde farklı sınıflandırıcıların performansları değerlendirilecektir.

4.2.1.4.2.1 Bayes Sınıflandırıcısı (Bayesian):

Sınıflandırma problemlerinde, her bir durum c sınıftan biri ile ilişkilendirilmekte ve genellikle her bir α_i ise gerçek durumun (sınıfın), ω_i olduğunda alınan karar olarak yorumlanmaktadır. Buna göre, eğer α_i davranışı seçilmişse ve gerçek durum ω_j ise, alınan karar yalnızca $i = j$ olduğunda doğrudur. Bu durumda, eğer hatalar en aza indirgenmek isteniyorsa, o zaman hata olasılığını, hata oranı, en aza indirgeyen karar kuralını aramak doğaldır.

Bayes karar kuramına göre ortalama hata olasılığını en aza indirmek için sonsal olasılığı $P(\omega_i | x)$ en yüksek değere ulaştıran i seçilmelidir. Bir diğer deyişle *en az hata oranı için*:

$$\text{Eğer } P(\omega_i | x) > P(\omega_j | x) \text{ her } i \neq j, \omega_i \text{ seçilir} \quad (12)$$

Bayes sınıflandırıcısının yapısı önsel olasılıklar ve koşullu olasılıklar $p(x | \omega_i)$ ile belirlenmektedir. Çeşitli olasılık yoğunluk işlevleri arasında şimdiye kadar en çok dikkati çekmiş olanı çok değişkenli Gauss dağılımıdır. Genel olarak d boyutlu Gauss dağılımı (13) 'deki gibi gösterilmektedir.

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu)^t \Sigma^{-1}(x - \mu)\right] \quad (13)$$

Burada x d -boyutlu bir vektör, μ yine d -boyutlu ortalama değer vektörü, Σ ise d 'ye d boyutlu ortak değişinti matrisini, $|\Sigma|$ ve Σ^{-1} de sırasıyla ortak değişinti matrisinin belirlenini ve evriniğini göstermektedir.

Öğrenme Süreci

Öğrenme sürecinde, ortalama vektörleri ve ortak değişinti matrisleri, sınıflara göre – yazı ve yazı olmayan – öğrenme kümesindeki öznitelik vektörleri kullanılarak hesaplanmaktadır.

Sınıflandırma

En az hata oranına göre sınıflandırma için, aşağıdaki farklı seçeneklerden her biri aynı sınıflandırma sonucunu vermektedir, fakat bunlardan bazıları anlamak veya hesaplama bakımından diğerlerinden daha basit olabilir:

$$g_i(x) = P(\omega_i | x) = \frac{p(x | \omega_i)P(\omega_i)}{\sum_{j=1}^c p(x | \omega_j)P(\omega_j)} \quad (14)$$

$$g_i(x) = p(x | \omega_i)P(\omega_i) \quad (15)$$

$$g_i(x) = \ln p(x | \omega_i) + \ln P(\omega_i) \quad (16)$$

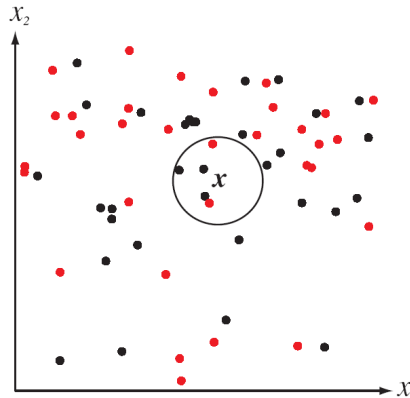
En son denklem (16) çok değişkenli Gauss dağılımlarına göre kolaylıkla hesaplanabilmektedir. Böylelikle aşağıdaki eşitlik elde edilmektedir;

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i) \quad (17)$$

Bu denklemdeki $|\Sigma_i|$, Σ_i^{-1} , $P(\omega_i)$ ve μ_i değerleri öğrenme sürecinde her bir sınıf için – yazı ve ya yazı olmayan – hesaplanmıştır, x vektörü ise sınıflandırılacak olan öznitelik vektörüdür.

4.2.1.4.2.2 K Yakın Komşuluk (k-Nearest Neighbour):

İsminden de anlaşılacağı üzere k-yakın komşuluk sınıflandırıcısı, x öznitelik vektörünün bulunduğu sınıfı en yakınında bulunan k komşu vektörün buldukları sınıflar içerisinde en fazla görülen olarak karar verir. Bir diğer değişle karar, en yakın k komşu vektörün incelenerek herbirinin oyu kullanılarak verilmektedir. Şekil 61'de örnek bir problem gösterilmektedir.



Şekil 60: İki sınıf bulunan bir problemde x vektörünün sınıfı bu durumda siyah olarak karar verilir [29].

4.2.1.4.2.3 Öğrenen Vektör Nicemleyicisi (Learning Vector Quantization):

LVQ sınıflandırma algoritması rekabete dayanan bir öğrenem sürecini temsil etmektedir. Vektör nicemleme, LVQ algoritmasında, sınıfların yoğunluk işlevlerini yaklaşıklamak yerine, doğrudan sınıf sınırlarını uzayda belirlemek için kullanılmaktadır. LVQ'nun öğrenme süreci ve kesinliği aşağıdaki kriterlere bağlıdır;

- i. Yaklaşık olarak en uygun sayıda kod tablosu vektörünün her bir sınıfa ilk değerleri ile atanması.
- ii. En uygun öğrenme hızı ve öğrenme sürecini durdurmak için gerekli en uygun kriter.

LVQ algoritması genel olarak 3 süreçle özetlenebilir; *Kod Tablosu Vektörlerinin İklendirilmesi, Öğrenme Süreci ve Sınıflandırma.*

Kod Tablosu Vektörlerinin İklendirilmesi

Kod tablosu vektörleri, basitçe anlatmak gerekirse, uzaydaki değişik sınıf bölgelerini tanımlamak için gerekli referans noktalarını sağlamaktadırlar. Deneysel olarak, çoğu tatbiki örüntü tanıma uygulamasında, farklı sınıflardaki örneklerin önsel olasılıkları ne kadar farklı olursa olsun, her sınıf için eşit sayıda kod tablosu vektörü ile başlamak iyi bir strateji olarak görülmüştür. Kod tablosundaki vektörlerin toplam sayısı, mevcut hesaplama gücüne ve tanıma süresine bağlı olarak karar verilmektedir.

Kod tablosundaki vektörlerin en uygun değerleri uyarlanır bir süreçte heasplanmaktadır. Bu vektörlerin iklendirilmesi için, öğrenme kümesinden bazı örnekler seçilerek kullanılmaktadır. Doğal olarak seçilen iklendirme için seçilen örnek vektörler doğru sınıf sınırlarının içinde olması gerekmektedir, fakat sınıf sınırları öğrenme sürecinden önce bilinemediği için yaklaşık kritere uygun atama yapılmaktadır. *k-Yaklaşık Komşu* (k-nearest neighbor) sınıflandırma metodu örnek seçilmiş vektöre karşı diğer vektörleri karşılaştırarak, yalnızca ve yalnızca çoğunluk k komşusu kendi bulunduğu sınıfa ait olduğunda, bu vektör bulunduğu sınıfın kod tablosu vektörü için ilk değer olarak seçilmektedir. Daha sonra sınıflar için en uygun nicemleme sınırları kabaca, sınıfların yoğunluk işlevlerine bağlı olmaksızın, aynı boyutta olmalıdır. Bunu sağlamak için bazı kod tablosu vektörleri eklenmekte ve bazıları da çıkarılmaktadır. Bu stratejinin sınıflandırma doğruluğunu arttırdığı görülmüştür.

Öğrenme Süreci

Bu süreç, kod tablosundaki vektörlerinin öğrenme kümesindeki vektörlere olan Öklit uzaklıkları kullanılarak değiştirilmesini içerir. Bu işlem aşağıda matematiksel olarak açıklanmaktadır;

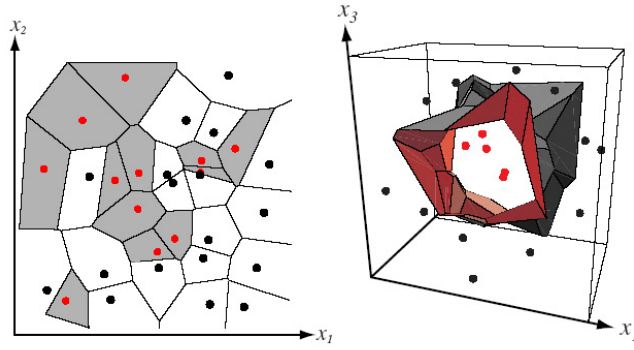
$$c = \arg \min_i \{ \|x - m_i\| \} \quad (18)$$

burada;

m_i : kod tablosu vektörü

x : öğrenme kümesinden alınmış girdi vektörü

Burada değinilmesi gereken önemli bir nokta da en az kodlama bozuluma sahip vektör nicemlemesi Voronoi nicemleyicisi olarak adlandırılmaktadır. Öklit metriği temel alınan en yakın komşuluk metodu kullanılarak ayrılan bölütler Voronoi hücreleridir. Sınıf bilgileri kullanılarak Voronoi vektörleri hareket ettirilmektedir bu da sınıflandırıcının karar sınırlarını iyileştirmektedir. Şekil 62'de iki sınıf problemi için sunulan grafiksel gösterim bulunmaktadır.



Şekil 61:Voronoi bölgeleri [29].

Burada m_c ve x minimum olarak varsayılırsa, bir diğer deyişle m_c rekabeti kazanan kod tablosu vektörü ise,

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \quad (19)$$

eğer x & m_c aynı sınıflarda ise,

$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)] \quad (20)$$

eğer x & m_c farklı sınıflarda ise,

$$m_i(t+1) = m_i(t) \quad (21)$$

ve diğer vektörler için.

(19),(20) ve (21) kullanılarak, kod tablosu vektörleri değiştirilmektedir.

Ayrıca öğrenme hızı $\alpha(t)$ de;

$$\alpha(t) = \alpha(t-1) / (1 + s(t)\alpha(t-1)) \quad (22)$$

burada $s(t) = -1$ farklı sınıflar için, $s(t) = +1$ aynı sınıflar için.

Her bir kod tablosu vektörü için ayrı ve değeri birden küçük öğrenme hızı, $\alpha_i(t)$, kullanılmaktadır.

Sınıflandırma

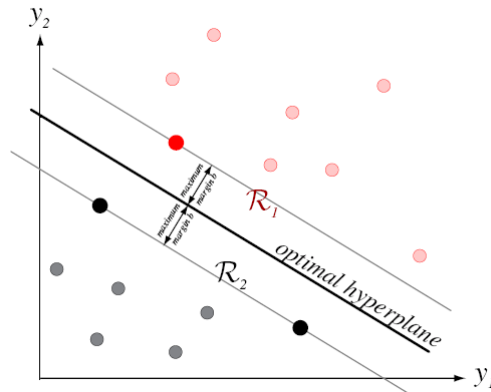
Sınıflandırma sürecinde, daha önce öğrenme sürecinde elde edilen her bir kod tablosu vektörleri kullanılarak sınıflandırılacak öznelik vektörü ile Öklit Uzaklığı bulunarak karşılaştırılmaktadır.

4.2.1.4.2.4 Destek Vektör Makinaları (Support Vector Machine):

Kuramsal açıdan, Destek Vektörü öğrenme şekli, örnekler üzerinden öğrenmenin ana fikrinden doğan ve aynı zamanda da temel fakat sade fikirlerden baz almaktadır; pratik açıdan da uygulamalarda yüksek performanslara ulaşabilmektedir. SVM sinir ağları, radyal taban fonksiyonları gibi kompleks modeller kullanmakla birlikte matematiksel olarak analizlerine imkan sağlamaktadır. SVM'ler örüntünün belirli bir ön-işleme ile daha yüksek boyuta sahip bir öznelik uzayında temsil edilmesini ve böylelikle de hyperplaneler ile rahatlıkla sınıflandırılmasını sağlamaktadır. Bu dönüşümü sağlayan gönderimler "kernel" olarak adlandırılmaktadırlar.

Öğrenme Süreci

SVMlerin öğrenme sırasındaki amacı, sınıfları ayıran en büyük marja sahip olan üstdüzlemseli (hyperplane) bulmaktır; böylelikle sınıfları daha genel olarak ayırt etmesi mümkün olmaktadır. Öğrenme sürecinde elde edilen destek vektörleri üstdüzlemselle eşit uzaklıkta olmaktadır daha da önemlisi bu vektörler hem üstdüzlemseli tanımlamakta, hem de sınıflandırması en zor örüntüler olmaktadır. Böylelikle bu örüntüler – destek vektörleri – sınıflandırma hakkında en fazla bilgi taşıyanlar olmaktadır. Şekil 63'de destek vektörleri ve en iyi sınıflandıran üstdüzlemsel görülmektedir.



Şekil 62: Destek vektörleri ve optimal ayırıcı hiper düzlemler [29]

SVMLerde öğrenme süreci öncelikli olarak girdi olarak sunulan örüntülerin daha yüksek boyutlu bir uzaya gönderimini sağlayan kernel işlevinin seçimi ile başlamaktadır. Bunun seçilmesi, problemin doğası ile bağlantılı olup farklı taban işlevler arasından – polinomsal tabanlar vs. – seçilmektedir.

$$L(a, \alpha) = \frac{1}{2} \|a\|^2 - \sum_{k=1}^n \alpha_k [z_k a^t y_k - 1]$$

Öğrenme süreci $L(a, \alpha)$ 'yı en küçük değer ulaştırmak olarak açıklanabilir. Burada y_k daha yüksek boyutlu uzaya gönderilmiş örüntüler, a ise ağırlık vektörü, z_k ise her bir örüntü vektörü için $z_k = \pm 1$ 'dir. Değişik bir formülleme yoluyla da $L(\alpha)$ kısıta bağlı olarak en yüksek değere ulaştıran değerler bulunmaktadır.

$$L(a) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j z_k z_j y_j^t y_k$$

$$\sum_{k=1}^n z_k \alpha_k = 0 \quad \alpha_k \geq 0, k = 1, \dots, n$$

4.2.1.4.3 Son İşlemler:

Bu adımda sınıflandırılmış – yazı ya da yazı olmayan – bloklar çeşitli temel imge işleme metodlar kullanılarak yazı bölgeleri haline dönüştürülmektedir. Bu amaçla sistem öncelikli olarak bağıntılı bileşenleri bulmakta daha sonra da bulunan bu bağıntılı yazı bölgelerini sınırlayıcı kutulara içerisine almaktadır.

4.2.1.4.3.1 Bağıntılı Bileşen (Connected Component):

Bağıntılı Bileşenler, her pikselin diğer bütün piksellerle bağıntılı olduğu küme olarak tanımlanmaktadır. Bu metodla elde edilen bölgeler yazıyı sınırlayan kutular için aday bölgeleri oluşturmaktadır. Bunun için bileşen etiketleme algoritması bütün bağıntılı bileşenleri bulmakta ve herbir ayırık kümenin elemanı olan piksele, her biri farklı olmak üzere bir etiket vermektedir. Bizim durumumuzda her bir blok etiketlenmek üzere kullanılmaktadır.

4.2.1.4.3.2 Sınırlayıcı Kutuların Çıkarılması (Bounding Box Extraction):

Elde ettiğim sınıflandırma sonuçları içerisinde kalan imge alanında yazı birden fazla satırdan oluşmuş olabilmektedir. OCR sisteminden önce sınıflandırma adımı sonuçlarının bu halde tutulması herhangi bir dezavantaj yaratmamaktadır. Ancak OCR sistemi için yazının daha sade elemanlara bir diğer deyişle satırlara, kelimelere ayrılması gerekmektedir. Bu amaçla yazı bloklarından yazı elemanlarını elde etmek için izdüşüm profilleri kullanılacaktır. Burada yazı bölgelerinin altında kalan imge alanının yerine Sobel ayırıt haritaları kullanılacaktır.

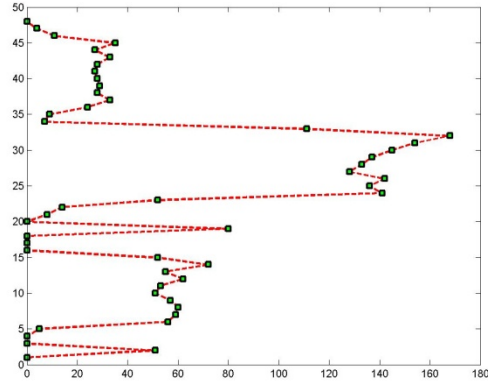
Şekil 64'da sınıflandırma sonrası elde edilen yazı blokları gösterilmektedir. Şekil 65'de bağıntılı bileşenler metodu ile elde edilen alanlardan birisinin Sobel ayrıt haritası gösterilmektedir. Bu noktada dikey izdüşüm profili kullanılarak Şekil 66 elde edilmiştir. Bu şekilden de görüldüğü üzere satırlar rahatlıkla ayrıştırılmaktadır. Şekil 67'de elde edilen sonucu göstermektedir.



Şekil 63: Sınıflandırma Sonucu



Şekil 64: Sobel Ayrıt Haritası



Şekil 65: Dikey İzdüşüm Profili



Şekil 66: Dikey izdüşüm haritası kullanılarak elde edilen sonuç

4.2.1.5 Karşılaştırmalı Deneyler:

Bu performans metodları kullanılarak 4 farklı deney yapılmıştır. İki blok temelli performans analizi kullanılarak elde edilen ve öznelilik çıkarma metodları ile sınıflandırıcıları karşılaştıran deneydir. Bu deneyde toplam – 152 yakalanmış kareden elde edilmiş – her biri 8×8 pixel boyutun da olan 240768 blok eğitim için, – 91 yakalanmış kareden elde edilmiş – yine herbiri 8×8 pixel boyutunda olan 144144 blok ise testler için kullanılmıştır.

Tablo 4’de “Ayrık Kosinüs Dönüşümü” (DCT), “Edge Histogram Descriptor” [28] ve “Homogeneous Texture Descriptor” [28] ile sınıflandırıcılar olarak “En Yakın Ortalama” (Nearest Mean) [29], “k-Yakın Komşuluk”(k-Nearest Neighbor) [29], “Bayeşçi” (Bayesian) [29] ve “Öğrenen Vektör Nicemleyicisi” (Learning Vector Quantization) [30] kullanılmıştır. Bu tablodaki detaylı sonuçlardan; DCT’nin test edilen bütün sınıflandırıcılar ile tutarlı ve başarılı sonuçlar verdiği görülmüştür. Ayrıca, LVQ’nun yine test edilen bütün öznelilik çıkarma metodları ile tutarlı ve en yüksek sonuçları verdiği görülmüştür.

Tablo 4: Çeşitli Öznelilik Çıkarma metodları ile sınıflandırıcıların ortak performansları

	Bayeşçi		En Yakın Ortalama		k-Yakın Komşuluk		Öğrenen Vektör Nicemleyicisi	
	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı
DCT	77.52%	89.02%	42.01%	65.44%	73.03%	85.56%	76.99%	83.38%
Edge Histogram	8.31%	94.63%	73.74%	59.97%	73.88%	61.47%	53.14%	78.47%
HTD	87.462	51.79%	81.24%	42.56%	73.23%	84.24%	83.17%	63.88%

Tablo 5 ve Tablo 6’da sonuçları verilen deneylerde ise, “Dalgacık Dönüşümü” ile elde edilen öznelilik vektörünün sınıflandırma için ne kadar başarılı olduğu ve elde edilen vektörün hangi elemanlarının sınıflandırma için daha fazla bilgi taşıdığı yani sınıflandırma başarısına etkisi daha yüksek olduğu araştırılmıştır. Bu deneylerde bir önceki deneyde olduğu gibi – 152 yakalanmış kareden elde edilmiş 8×8 pixel boyutunda 240768 blok eğitim için, – 91 yakalanmış kareden elde edilmiş yine 8×8 boyutunda 144144 blok ise testler için kullanılmıştır, ayrıca dalgacık dönüşümü olarak “Haar Dalgacığı” kullanılmıştır.

Tablo 5: Haar Dalgacık Dönüşümü seviyeleri ile çeşitli sınıflandırıcıların karşılaştırmalı performansları.

	Bayeşçi		En Yakın Ortalama		k-Yakın Komşuluk		Öğrenen Vektör Nicemleyicisi	
	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı
Haar Dalgacık Dönüşümü								
LL’lar hariç	57.99%	92.28%	46.98%	90.60%	75.46%	87.23%	83.68%	80.01%
Yalnızca HH’lar	49.54%	93.82%	53.53%	92.44%	87.77%	80.03%	92.43%	74.32%
Yalnızca LH’lar	38.59%	96.01%	43.80%	92.54%	81.18%	84.22%	90.37%	71.41%
Yalnızca HL’lar	39.74%	94.06%	39.78%	90.21%	70.67%	81.31%	84.52%	77.10%
LH’lar ve HL’lar	49.41%	93.37%	46.94%	90.57%	76.82%	84.44%	80.67%	83.07%
Doermann [31]	42.92%	94.12%	46.98%	90.60%	75.40%	86.09%	83.10%	81.68%

Tablo 6: Haar Dalgacık Dönüşümü ile elde edilen öznelik vektörleri ile çeşitli sınıflandırıcıların karşılaştırmalı performansları.

Haar Dalgacık Dönüşümü	Bayesçi		En Yakın Ortalama		k-Yakın Komşuluk		Öğrenen Nicemleyicisi	
	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı	Yazı Bulma Oranı	Yazı Olmayan Blokları Bulma Oranı
Yalnızca Ortalama Değerler	56.71%	91.05%	50.15%	54.56%	71.61%	82.80%	79.59%	74.00%
Yalnızca Değişimler	44.34%	93.49%	62.12%	91.55%	82.80%	81.86%	88.81%	74.10%
Yalnızca 3. Merkezi Momentler	31.97%	96.29%	48.29%	88.59%	77.90%	84.87%	88.51%	77.29%
Ortalama Değerler ve 3. Merkezi Momentler	49.41%	93.41%	48.29%	88.59%	78.13%	84.44%	90.10%	73.68%
Değişimler ve 3. Merkezi Momentler	42.92%	94.12%	46.98%	90.60%	75.40%	86.09%	84.25%	81.28%
Ortalama Değerler ve Değişimler	59.31%	91.43%	62.12%	91.55%	84.08%	80.34%	86.92%	78.26%

Tablo 7’de ise 91 yakalanmış görüntüde uygulanan 4 farklı yazı bulma metodundan elde edilen performans sonuçları gösterilmiştir.

Tablo 7: Sınırlandırma Kutusu Performans Analiz Sonuçları

	Doğruluk Oranı	Hata Oranı	Yanlış Bulma Oranı
Haar + LVQ	0.773659	0.248320	0.755057
Haar + Bayesçi	0.460551	0.598273	0.449927
DCT + LVQ	0.739595	0.282382	0.734520
DCT + Bayesçi	0.69599	0.325988	0.563812

4.2.1.6 Yakalanan Görüntüler Üzerinde Elde Edilen Bütünleşik Sonuçlar:

Yukarıda anlatılan çalışmalarda bloklar birbirlerinden bağımsız olarak değerlendirilmiş ve çıkan sonuçların performansları irdelenmiştir. Bu adımda yapılan çalışmalarda bağımsız tespit edilen blokların çeşitli basit kurallar ile birleştirilmesi sonunda elde edilen görüntü kareleri görülmektedir (Şekil 68).



Şekil 67 : Tipik tanıma sonuçları

4.2.2 Görüntü İçindeki Yerleri Belirlenmiş Bölgelerin Eşiklenmesi:

Yakalanmış video görüntülerden metin tanıma probleminde performansa doğrudan etkisi olan adımlardan birisi de eşiklemedir. Eşikleme ile bağlantılı sorunların bir çoğu gürültü sürecinin, durağan olmaması, ilintili ya da Gaussian olmaması; ortam aydınlatmasındaki problemler; yetersiz farklılıklar ve nesnenin şeklinin görünüm ile uyumsuz olması gibi durumları içermektedir. Son olarak da deneysel verilerin yetersiz olması eşikleme algoritmalarının performans değerlendirmelerine etki etmektedir.

Deneysel sonuçlar, imgenin eşiklenmesinden elde edilen iki seviyeli imgenin Optik Karakter Tanıma (OCR)'dan geçirilerek ortaya çıkan sonuçların karşılaştırılması ile elde edilmiştir. OCR olarak *ABBYY Finereader 7.0 Professional* kullanılmıştır. Deneysel sonuçlar Tablo 8'de detaylı olarak gösterilmektedir. Veriler çeşitli TV kanal görüntülerinden ayrılan karakter öbekleri olarak seçilmiştir. Testin başarı kriteri "yerinde ve doğru harf tahmini" olarak kabul edilmiştir. Eşikleme algoritmaları için bir referensta [27] incelenmiş çok sayıda farklı algoritma test edilmiştir.

Deneysel sonuçlar altında testimizi ilk üç sırada bitiren algoritmalar:

1. Kümeleme methodu olan Llyod [27],
2. Yerel methodlardan Bernsen [1],
3. Entropi methodlarından olan Kapur [26]

olarak belirlenmiştir.

Burada göz ardı edilmemesi gereken nokta, bu metodların başarısının verilen problem – imgelerde bulunan metinlerin tanınması – karşısında ve kullanılan OCR'a göre değerlendirilmiştir. Sonuç olarak, farklı bir uygulamada farklı bir OCR sonucu; bu elde edilen değerlendirmeden farklı bir sonuca neden olabilir. Fakat kullanılan OCR'ın (*ABBYY Finereader 7.0*) piyasadaki en etkili OCR olduğu düşünülürse bu uygulama için sonuçların genel bir değerlendirme olarak kabul edilmesi yanlış olmaz.

Tablo 8: Eşikleme sonuçları

Data #	TELEDIARIO		2		policia		Solbank		DIKKAT RADAR VAR		ONLINE EDITOR		Almanya dan	
	OCR Output	hit	OCR Output	hit	OCR Output	hit	OCR Output	hit	OCR Output	hit	OCR Output	hit	OCR Output	hit
Shape_Rosenfeld	"TELEDIARIO"	7/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"DIKKAT RADAR VAR"	0/16	"ONLINE EDITOR GARY HEWSON"	"Almanya dan"	10/11	1,609
Shape_Sezan	"TELED AR 0"	7/10	"TELED AR 0"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"thul UMI IM"	0/16	"HEWSON"	"Almanya dan"	10/11	2,609
Shape_Olivia	"TELED AR0"	7/10	"TELED AR0"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEW"	0/16	"HEWSON"	"Almanya dan"	3/11	2,275
Shape_Ramesh	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,297
Cluster_Ridler	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,592
Cluster_Otsu	"TELEDIARIO"	3/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,617
Cluster_Lloyd	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	5,297
Cluster_Kittler	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	3,000
Cluster_Yanni	"TELED AR0"	3/10	"TELED AR0"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,639
Cluster_Jawahar a	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	3,036
Cluster_Jawahar b	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,592
Cluster_Jawahar d	"TELEDIARIO"	1/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	0	0,100
Entropy_Pun a	"TELEDIARIO"	2/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	2,574
Entropy_Pun b	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	10/11	1,185
Entropy_Kapur	"TELEDIARIO"	3/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,760
Entropy_Li	"TELED AR0"	8/10	"TELED AR0"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,722
Entropy_Shanbag	"TELED AR0"	8/10	"TELED AR0"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,722
Entropy_Yen	"TELEDIARIO"	8/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	2/11	2,962
Entropy_Brink	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	8/11	0,545
Entropy_Sahoo	"TELEDIARIO"	8/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,660
Attribute_Isai	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	1,438
Attribute_Hertz	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	1,188
Attribute_Ogorman	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	0/11	0,000
Attribute_Huang	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,038
Attribute_Pikaz	"TELEDIARIO"	1	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,149
Attribute_Sezgin	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	2,878
Spatial_Pal a	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	2,000
Spatial_Pal b	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	2,000
Spatial_Abutaleb	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	2,580
Spatial_Beghdadi	"TELEDIARIO"	1/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	7/11	0,736
Local_Yasuda	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	10/11	1,623
Local_White	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	1,688
Local_Jiblack	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	3,547
Local_Bernsen	"TELEDIARIO"	3/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	4,822
Local_Palumbo	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	1,000
Local_Yanowitz	"TELEDIARIO"	8/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	2,238
Local_Kamel	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	0/11	0,000
Local_Sauvola	"TELEDIARIO"	0/10	"TELEDIARIO"	0/45	"POLLICIA"	0/7	"Solbank"	0/7	"HEWSON"	0/16	"HEWSON"	"Almanya dan"	1	1,000
Average Hit:		0,48		0,35		0,57		0,30		0,19		0,84		

4.2.2.1 Sonular

Bu alıřmada grnt iindeki yazıları bulmak iin kullanılan temel algoritmalar test edilmiř ve performansları ortaya konmuřtur. Bu sonulara gre Haar Dalgacık Dnřm ile elde edilen znelik vektrleri bununla birlikte kullanılan LVQ sınıflandırıcısı en iyi sonuları doėurmuřtur. Bununla birlikte en iyi sonularla birlikte sistemin hesaplama karmařıklığı da nemli bir deėerlendirme parametresi olarak gze arpmaktadır. Bu parametreyi de gz nne almak gerekirse, Haar Dalgacık Dnřm ve LVQ ile elde edilen sistem yeterli grnmektedir. Ayrıca performans olarak LVQ ve Haar Dalgacık Dnřmnn arkasında kalmıř olsa da, DCT ve Bayeřçi yaklařım kullanılarak elde edilen sistem hesaplama karmařıklığı en az ve aynı zamanda kabul edilebilir sonular reten sistem olarak karřımıza ıkmaktadır. Yapılan deneylerden elde edilen bařka bir sonu da Ayrıt Histogram Tanımlayıcısının znelik vektr olarak en yakın ortalama sınıflandırması hari btn deneylerde en dřk sonuları verdiėidir.

Genel olarak Haar Dalgacık Dnřmnn bařarılı olması, yazı blgelerini nitelendiren yerel doėru paralarını etkin bir řekilde bulmasıdır. Ayrıca DCT katsayılarından elde edilen znelik vektrnn de daha nce de bahsedildiėi gibi resimde yksek frekans bileřenleri olarak ortaya ıkan ani deėiřiklikleri yani yazı blgelerini etkin olarak nitelendirmesi deneylerde bařarılı sonular elde edilmesini saėlamıřtır. Bununla birlikte Ayrıt Histogram Tanımlayıcısının yalnızca belirli ynlerdeki ayrıtları yeterli bir biimde nitelendirememesi (ki bunun sebebi yazı blgelerinin yalnızca 5 ayrıt tipi ile etkin deėerlendirilemeyecek olması) deney sonularındaki en yksek yanlıř bulma sonularından birisini doėurmuřtur.

Ayrıca LVQ ile Bayeřçi sınıflandırma metodlarının karřılařtırılmasından elde edilen sonular LVQ'nun lehine ıkmıřtır. LVQ ile daha iyi bulma oranının yakalanmasındaki en nemli sebeplerden birisi LVQ metodunun elde edilmiř rneklem uzayını nitelendirirken rneklem uzayını korumasıdır. Buna karřın Bayeřçi sınıflandırıcısının btn uzayı ok deėiřkenli Gauss daėılımına benzetmesi performans analizlerinde LVQ'ya gre daha dřk bulma oranına sahip olmasına neden olmuřtur.

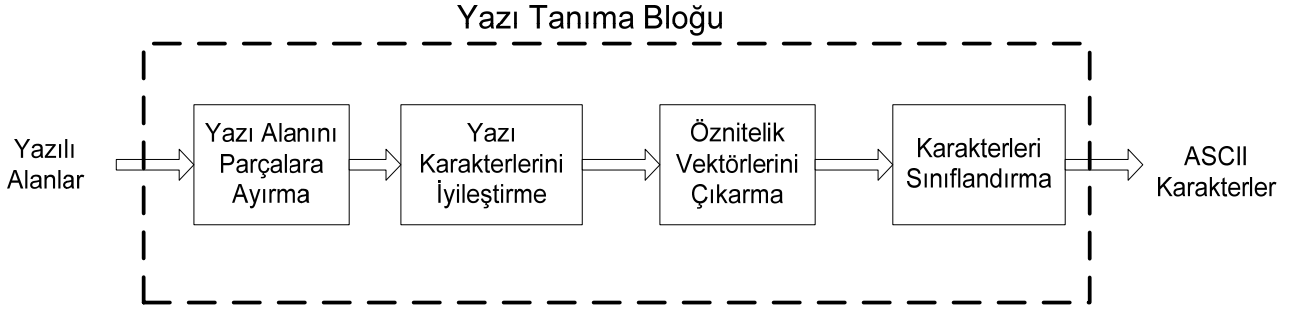
4.2.3 GRNTLERDEN YAZI TANIMA

Yukarıda nerilen sistemde yazının tanınması ařamasında ticari bir rn olan *ABBYY Finereader 7.0 Professional* kullanılmıřtır. ncelikle bu programın cretli olması ve aık bir kaynaėa sahip olmaması ve karakter tanıma adımı zerinde hi bir kontrol imkanı bırakmaması, bu ticari yazılımın hedeflenen sistemin ierisinde yer almasını engellemiřtir. Bu sebeple, yazı tanınması problemi iin aık kaynaėa sahip ve cretsiz olan bir yazılım arařtırılmıřtır. Bu arařtırmaların sonucu olarak "*NIST Form-Based Handprint Recognition System*" [35] kullanılması ve uyarlanması benimsenmiřtir. COST 292 projesi kapsamında yapılacak video yazı tanıma bařlıėı altında zgn optik karakter tanıma (OCR) algoritmaları geliřtirme gibi bir hedef bulunmamaktadır. Ancak, byle bir sistemin iinde yer alacak OCR bloėunun eřitli parametre ve sonuları ile sistemin geri kalan kısmının etkileřmesi son derece gerekli durmaktadır. Raporun ilk blmnde kullanılmaya bařlanan OCR yazılımı kısaca tanıtılacak, daha sonra dřk znrlkl video karelerini byle bir sistemde bařarılı kılabilmek iin kullanılan yntemler anlatılacaktır.

4.2.3.1 NIST Yazı Tanıma Sistemi: (OCR)

NIST Yazı Tanıma Sistem iinde temel rnt tanıma metodlarından biri olan ve dkmanlarda yazı tanıma konusunda geerliliėi kanıtlanmış bir metod olan ok katmanlı

algılayıcı (Multi-Layer Perceptron-MLP) kullanılmaktadır. Ayrıca eğitim setinin yüksek çözünürlükteki el yazısı için tasarlanmış olması, düşük çözünürlüklerde karşılaşılan karakterlerin birbirine dokunması veya karakter kayıklığı gibi problemlerle başa çıkabilmesini de sağlamaktadır. Eğitim kümesinin 1,499 farklı kişiden elde edilen 252,144 karakter gibi geniş ve genel bir veri seti olması ise de dikkat çekicidir.



Şekil 68: NIST Yazı Tanıma Sisteminin Blok Çizeneği

Şekil 69'de bloklar halinde gösterilen NIST Yazı Tanıma Sisteminin açıklaması aşağıda verilmiştir. Bundaki sonraki alt kısımlarda ise bu sistemin alt kısımları kısaca açıklanacaktır.

4.2.3.1.1 Yazı Alanını Parçalara Ayırma:

Bu adımda, sisteme girdi olarak verilen iki seviyeli veri ilk olarak *Bağlantılı Bileşen (Connected Component)* [38] metodu kullanılarak, karakter boyutu çok hızlı bir şekilde bölütlenmektedir. Fakat bu yöntemin doğrudan kullanımı birbirine dokunan karakterleri ve parçalı karakterlerin (noktalı vs. karakterlerin) tanınmasında hatalar oluşturmaktadır. Bağlantılı Bileşen üzerine kurulu olan sistem basit uyarlanır yazı stili [34] kullanarak karakterleri bölütlenmektedir. Böylelikle parçalı karakterler yeniden oluşturulmakta, çoklu karakterler ayrıştırılmakta, ve gürültü bileşenleri ayrılmaktadır. Bu yöntemden elde edilen bir örnek sonuç Şekil 70'de gösterilmektedir.



Şekil 69: Örnek Bölütleme Sonucu

4.2.3.1.2 Yazı Karakterlerini İyileştirme:

Bölütlenmiş karakterler boyut, şekil ve eğiklik bakımında farklılık göstermektedir. Bu yüzden iyileştirme adımıyla karakterlerin tanımlanmasını yalnızca karakterlerin şekilleri ile gerçekleştirmek amaçlanmaktadır. Bölütlenmiş karakterlerin boyutları ihtiyaca göre yukarı ya da aşağı ölçeklenmektedir. Bununla birlikte çizgi kalınlıkları basit morfoloji kullanılarak normalize edilmektedir, bir diğer deyişle fazla kalın olan karakter çizgileri inceltilmekte, fazla ince olan karakterler ise kalınlaştırılmaktadır. Eğiklik farklılıklarının kaldırılması için ise en üst

sol yazı pikseli ile en alt sol yazı pikselinden geçen doğru kullanılmaktadır. Boyut ve eğiklik normalleştirilmesi hakkında daha fazla detay [35]'de açıklanmaktadır.

4.2.3.1.3 Öznitelik Vektörlerinin Çıkarılması:

Karhunen Loève Dönüşümü (Karhunen Loève Transform) ile herbir bölütlenmiş karakterin piksellerden oluşan vektörün boyutu, üzerinde bulunan gürültü arıtılıp tanımlama için en uygun öznitelik vektörü elde edilmektedir [36]. Bunun için iyileştirilmiş karakterlerden oluşan bir eğitici küme kullanılarak ortak değişinti matrisi (*Covariance Matrix*) hesaplanmaktadır. Ortak değişinti matrisinin daha sonra özdeğerleri ve özvektörleri bulunmaktadır. Bu vektörlerden en üstteki n tanesi daha sonra öznitelik çıkarma adımında kullanılmak üzere taban vektörleri olarak seçilmektedir. Öznitelik vektörleri ise daha önceden hesaplanan bu taban vektörleri ile bölütlenmiş karakterlerin vektörlerinin sayılı çarpımı alınarak oluşturulmaktadır.

4.2.3.1.4 Karakterleri Sınıflandırma:

Bu adımda daha önceden hesaplanan öznitelik vektörleri MLP [39] ağına sunulurken (bu ağınlık değerleri daha önceden eğitim safhasında bulunmuştur) sınıflandırma sonucu ile bir güvenilirlik değeri elde edilmektedir.

4.2.3.1.5 NIST OCR sistemine ait Örnek Sonuçlar:

Şekil 71'de sunulan otomatik yazı bulma sonucunda belirlenen sınırlandırıcı kutulardan alınan imge bilgisinin eşikleme safhasından sonra OCR sistemine verilmesi ile elde edilen sonuçlardan bazıları Tablo 9'da gösterilmiştir. Bu tablodaki güvenilirlik değerleri MLP algoritmasının sınıflandırıcısının çıktısı olarak elde edilmektedir. Proje içinde geçmişte kullanılan profesyonel yazılımların hiç birinden bu tip bir bilgiye ulaşma imkanı bulunmamaktadır. Açık kodlu NIST OCR sistemini kullanmanın getirdiği bu önemli avantaj, sistemin daha yüksek başarı göstermesi için kullanılacaktır.



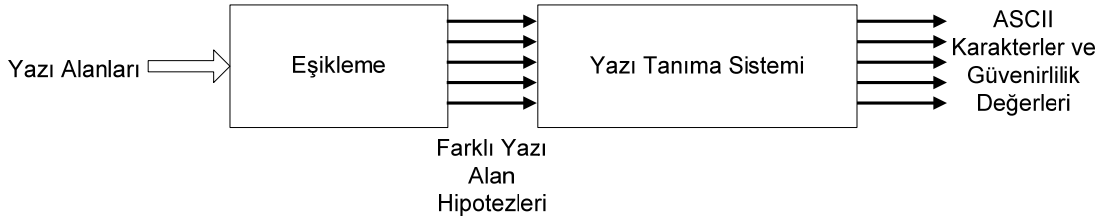
Şekil 70: Örnek Otomatik Yazı Alanlarını Bulma Sonucu

Tablo 9: Örnek OCR Sonuçları

#	OCR Sonucu	Güvenirlilik Değerleri
1	ATTACK	1.00 0.86 0.65 0.98 0.79 0.76
2	IN	0.98 0.73
3	INDIA	0.96 0.99 0.92 0.98 0.91
4	GUARD	0.80 0.80 0.82 0.92 0.83
5	WOUNOEB	0.77 0.90 0.75 0.59 0.57 0.97 0.32
6	BOME	0.87 0.97 0.48 0.20

4.2.3.2 Çok Seviyeli Eşikleme:

Şu an kullanılmakta olan bir çok Optik Karakter Tanıma (OCR) sistemi iki seviyeli veriler üzerinde çalışmaktadır. Bunun en önemli sebebi, OCR teknolojisinin daha çok dökümanlar üzerindeki yazının ASCII biçimine çevrilmesi için kullanılmasıdır. Bu nedenle karmaşık verilerde tipik OCR'lar yüksek performans değerlerine ulaşamamaktadır. Her tip görüntü için yazının otomatik bulunup tanınması probleminde ise, yazılı alanlar rahatlıkla karmaşık arka plana sahip olabilirler. Bununla birlikte, algılama adımında karşılaşılabilecek problemler, örneğin gürültü, v.b., döküman analizinde karşılaşılanlardan daha şiddetli olabilir. Bu tip problemlerden dolayı temel eşikleme metodları sınırlayıcı kutuların içerisinde kalan yazı bölgesi ile arka planı ayırt etmekte yetersiz kalabilmektedirler. Bu problemleri çözebilmek için sınırlayıcı kutuların içerisinde kalan imge verisinin ikiden fazla seviyeye gruplanması hakkında çoklu hipotez yöntemini kullanan çalışmada [44] bölütleme için ortalama eniyileme (Expectation Maximization-EM) ve Gibbs kökenli EM önerilmiştir. Ancak, daha sonraki çalışmalarda [45], bu yöntemlerin k-ortalamlar (k-Means) ile deneysel karşılaştırması yapıldığında K-ortalamlar metodunun en bu yöntemler kadar iyi sonuçlar çıkardığı görülmektedir. Bu nedenle projemiz kapsamında tasarlanan sisteme K-ortalamlar algoritmasının eklenmesine karar verilmiştir.



Şekil 71: Eşikleme ve NIST OCR Sistemi

4.2.3.2.1 K-ortalamlar kullanarak Yazı Alan Hipotezleri Oluşturma

Öznitelik uzayındaki noktalar üzerinde en popüler ve en geniş ölçüde çalışılmış olan bölütleme metodu k-ortalamlar metodudur. d -boyutlu \mathbf{R}^d uzayından alınan n elemanı içeren P kümesi ve k tam sayısı için problem, \mathbf{R}^d uzayında bulunan k noktayı (merkezi), P kümesindeki her bir noktanın buldukları bölütlerin merkezine olan Öklit uzaklıklarının karelerinin ortalamasının en az olacak şekilde belirlemektir. K-ortalamlar probleminin çözümü için kullanılan en popüler metodlardan birisi Llyod algoritmasıdır [43]. Llyod algoritması yerine [37]'de önerilen Llyod ve yerel arama metodları karması olan algoritması deneyler sonucu gösterdiği performansdan dolayı tercih edilmiştir.



Şekil 72: Örnek Otomatik Yazı Bulma Sonucu

Şekil 73'de, elde edilen otomatik yazı bulma sonucunda ortaya çıkan sınırlayıcı kutuların bazıları karmaşık arka plana sahip oldukları için temel eşikleme yöntemleri bazı hataları beraberinde getirebilecektir. Bunun ortadan kaldırılması için önerilen k-ortalamlar metodu uygulandığında Şekil 74'de görülen sonuç elde edilmektedir. Sonraki adımda kümelerin herhangi bir ikili kombinasyonu OCR'a girdi olarak sunulmaktadır. Bu yaklaşımdaki temel amaç, yazıyı ve geri kalanı arka plan olarak nitelendirilen bölgeleri ayırt etmektir. Şekil 74'de gösterilen mavi, siyah ile beyaz renkler arasından siyah ile beyaz kombinasyonu iyi bir çözüm oluşturmaktadır.

YÜZÜKLERİN

Şekil 73: k-ortalamlar sonucunda ortaya çıkan üç küme, üç farklı renk – beyaz, siyah ve mavi – ile işaretlenmiştir.

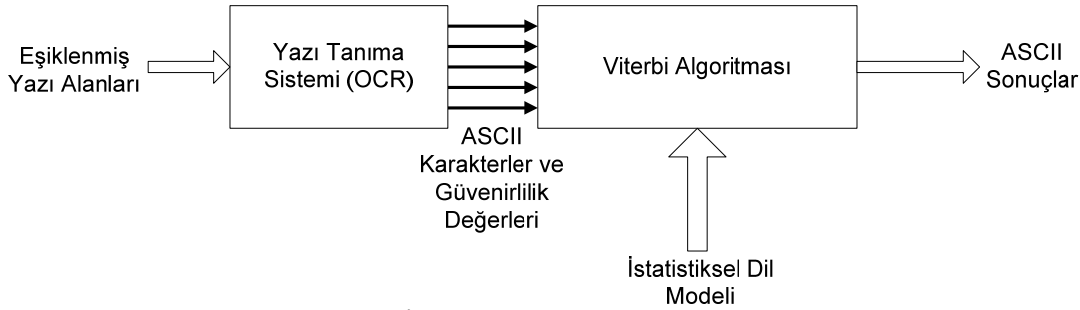
Aşağıdaki şekilde ise (Şekil 75) metodun ortaya çıkardığı sonuçlardan siyah ile beyaz kombinasyonu gösterilmektedir. Bu şekilde elde edilen alternatif hipotezler NIST OCR tarafından işlenerek her birisi için sonuç üretilmektedir, belirli bir kriter ile de elde edilen sonuçlar elenmektedir.

YÜZÜKLERİN

Şekil 74: Üç kümeden siyah ve beyaz renklerle elde edilen bir sonuç.

4.2.3.3 İstatistiksel Dil Modeli ve Viterbi Algoritması

NIST OCR performanslarının artırılması için kullanılan önemli yöntemlerden birisi bağlamsal bilginin kullanılmasıdır. Bu yaklaşımı kuvvetlendiren noktalardan birisi yanlış yazılmış kelimeleri algılama yeteneğimize. Okuyucu zor okunan yazıları cümlelerin içeriğini, sözcük dağılımını ve yazım örüntülerini kullanarak en uygun kelimeyi seçmektedir. Bu düşünceden yola çıkarak ulaşılabilecek yaklaşımlardan birisi sözlük kullanarak, tanınan yazıdaki kelimelerin belirli bir ölçüte göre en yakın karşılığını bulmaktır. Ancak bu yöntemin performansının sözlüğün geçerliliği ile büyüklüğüne bağlı olması ve ayrıca sözlüğün genişlemesinden kaynaklanan hesaplama karmaşıklığındaki artıştan dolayı tercih edilmemektedir. Bir diğer yaklaşım ise dillerin yazılış örüntülerini kullanmaktır. Bir diğer deyişle dilin yazılışını *Markov Süreci (Markov Source)* ile modelleyerek bu modeli yazının tanınmasında kullanmaktır [40]. Bu yöntemde tipik olarak, dilin istatistiksel modelinden ve OCR tarafından tanınan yazının belirli bir ölçüte göre güvenilirliğinden yararlanarak yazının kestirimi gerçekleştirilir.



Şekil 75: İstatistiksel Dil Modeli ve Viterbi Algoritması

OCR çıktısı olarak gözlemlenen harfler (y_0, y_1, \dots, y_n) kullanılarak son (a posteriori) olasılığı $\Pr(x_0, \dots, x_n | y_0, \dots, y_n)$ en yüksek dereceye çıkarılan harflerin, (x_0, x_1, \dots, x_n) , bulunması amaçlanmaktadır. Eğer dilin Markov Süreci olduğunu varsayarsak her bir olasılığı tek tek incelemek yerine yalnızca bir toplamı, Viterbi Algoritması [42] ile en yüksek dereceye çıkarılan seriyi bulmak yeterli olacaktır. Dilin *n. dereceden Markov Süreci* ile modellenmesi aşağıdaki eşitliği doğurmaktadır;

$$\Pr(x_k | x_{k-1}, \dots, x_0) = \Pr(x_k | x_{k-1}, \dots, x_{k-n}) \quad (23)$$

Bu noktadan sonra yapılması gereken bazı önemli varsayımlar bulunmaktadır. Bunlardan birincisi OCR sonuçlarının yazıyla eşzamanlı (senkron) olmasıdır. Yazının her bir elemanı, x_k , doğru sırada tanınıp yalnızca bir tane sonuç, y_k , üretmelidir. Bir diğer varsayım ise, OCR'ın karakter tanımadaki hafızasız olmasıdır.

$$\Pr(y_k | y_0, \dots, y_n, x_0, \dots, x_n) = \Pr(y_k | x_k) \quad (24)$$

Bu varsayımlar ışığında;

$$\Pr(x_0, \dots, x_n | y_0, \dots, y_n) = \frac{\Pr(y_0, \dots, y_n | x_0, \dots, x_n) \Pr(x_0, \dots, x_n)}{\Pr(y_0, \dots, y_n)} \quad (25)$$

$$\Pr(y_0, \dots, y_n | x_0, \dots, x_n) = \prod_{i=0}^{i=n} \Pr(y_i | x_i) \quad (26)$$

(27)'de ise dilin 1. dereceden Markov Süreci olarak modellenmesi kullanılmıştır;

$$\Pr(x_0, \dots, x_n) = \prod_{i=1}^{i=n} \Pr(x_i | x_{i-1}) \quad (27)$$

$$\Pr(y_0, \dots, y_n | x_0, \dots, x_n) \Pr(x_0, \dots, x_n) = \prod_{i=1}^{i=n} \Pr(y_i | x_i) \Pr(x_i | x_{i-1}) \quad (28)$$

$\Pr(y_0, \dots, y_n)$ sabit bir değere sahip olduğundan, aşağıdaki forma indirgeyebiliriz,

$$\begin{aligned} L(x_0, \dots, x_n) &\square \ln [\Pr(y_0, \dots, y_n | x_0, \dots, x_n) \Pr(x_0, \dots, x_n)] \\ &= \sum_{i=0}^n [\ln \Pr(y_i | x_i) + \ln \Pr(x_i | x_{i-1})] \end{aligned} \quad (29)$$

Viterbi Algoritması en büyük olabilirlik kestirimi sağlayan ve çok çeşitli alanda kullanılan bir metoddur. Yazının tanınması probleminde ise MAP kestirimini sağlamak için yukarıda elde edilen sonuçlar ışığı altında işlev kazanmaktadır. Böylelikle (29)'de elde edilen $L(x_0, \dots, x_n)$ fonksiyonunu en yüksek değere ulaştıran karakterler Viterbi Algoritması kullanılarak bulunabilmektedir. Burada $\Pr(y_i | x_i)$ her bir harf için OCR'dan elde edilen güvenilirlik değerleri, $\Pr(x_i | x_{i-1})$ ise istatistiksel dil modeli kullanılarak bulunan, belirli bir harften sonra gelen başka bir harfin geçiş olasılığıdır.

İstatistiksel Dil Modeli çıkarılmasında [41]'de açıklanan sistem kullanılmaktadır. Clarkson ve Rosenfeld'in bu çalışmasında dil modeli ardışık kelimelerin birbiri ile ilişkileri üzerine kurulmuş bulunmaktadır. Ancak önerilen yaklaşımda, daha önce de açıklandığı gibi, kelimeler arası ilişkiden ziyade kelimeler içerisindeki ardışık harflerin ilişkileri kullanılmaktadır.

Tablo 10'de Şekil 71'den elde edilen OCR sonuçları, OCR tarafından üretilen en iyi ilk üç sonuçtaki harflere karşılık gelen güvenilirlik değerleri ve İstatistiksel Dil Modelinin kullanılması ile elde edilen sonuçlar gösterilmektedir.

Tablo 10: İstatistiksel Dil Modeli kullanılarak OCR sonuçlarının iyileştirilmesi

#	OCR Sonucu	Güvenirlilik Değerleri	İDM Sonuçları
1	ATTACK	1.00 0.86 0.65 0.98 0.79 0.76	ATTACK
	FIIIZQU	0.00 0.02 0.23 0.01 0.20 0.23	
	ZLAFBM	0.00 0.02 0.09 0.00 0.01 0.02	
2	IN	0.98 0.73	IN
	XV	0.01 0.06	
	QF	0.01 0.05	
3	INDIA	0.96 0.99 0.92 0.98 0.91	INDIA
	WIBBK	0.01 0.02 0.02 0.01 0.10	
	AUOJD	0.01 0.01 0.01 0.01 0.01	

4	GUARD	0.80 0.80 0.82 0.92 0.83	GUARD
	NOKUO	0.02 0.14 0.09 0.08 0.02	
	RCYGM	0.01 0.06 0.08 0.01 0.01	
5	WOUNOEB	0.77 0.90 0.75 0.59 0.57 0.97 0.32	WOUNDED
	HCNRDIO	0.53 0.17 0.12 0.09 0.34 0.33 0.28	
	ARTEROD	0.24 0.11 0.08 0.09 0.07 0.14 0.20	
6	BOME	0.87 0.97 0.48 0.20	BOMB
	EYYB	0.03 0.03 0.31 0.16	
	WAHV	0.02 0.02 0.19 0.13	

Şekil 77 ve Tablo 11’de de yine OCR tarafından üretilen en iyi ilk üç sonuç, harflere karşılık gelen güvenilirlik değerleri ve İstatistiksel Dil Modelinin kullanılması ile elde edilen sonuçlar gösterilmektedir.



Şekil 76: Örnek Sonuç

Tablo 11: İstatistiksel Dil Modeli kullanılarak OCR sonuçlarının iyileştirilmesi

#	OCR Sonucu	Güvenirlilik Değerleri	İDM Sonuçları
1	OUILAWD	0.98 1.00 0.73 0.69 0.43 0.98 0.82	OUTLAWED
	QBTOLHIP	0.02 0.01 0.26 0.20 0.42 0.03 0.13	
	GRLYXPOQ	0.01 0.00 0.01 0.16 0.05 0.00 0.05	
2	GROUP	0.72 0.98 0.97 0.98 0.97	GROUP
	OFQRA	0.28 0.01 0.01 0.01 0.04	
	QZNMM	0.01 0.01 0.01 0.01 0.00	
3	WEAKENED	0.78 0.99 1.00 0.98 1.00 0.65 1.00 0.87	WEAKENED
	HKFHKKKE	0.18 0.01 0.01 0.12 0.05 0.10 0.09 0.33	
	PWQIBMOQ	0.11 0.01 0.01 0.07 0.05 0.09 0.08 0.29	
4	CRACKDOHB	0.96 0.95 1.00 0.82 0.87 0.64 0.73 0.46 0.33	CRACKDOWN
	MNTMNUDWN	0.04 0.08 0.06 0.24 0.20 0.19 0.20 0.31 0.32	
	QHUBDOQUQ	0.00 0.08 0.06 0.22 0.12 0.18 0.11 0.27 0.12	
5	WINS	0.32 0.96 0.59 1.00	VINS
	VKMZ	0.25 0.07 0.34 0.16	
	HBAG	0.19 0.06 0.15 0.12	
6	CHAMPIONSHIP	0.41 0.87 0.88 0.38 0.52 0.64 0.91 0.84 0.95 0.79	CHAMPIONSHIP
	MMMHMMQHGMRH	0.30 0.10 0.07 0.07 0.12 0.28 0.04 0.08 0.02 0.08	
	CPBWHXZBRKMM	0.29 0.04 0.06 0.06 0.11 0.14 0.02 0.07 0.01 0.07	

4.2.3.3.1 Deneysel Sonular

Son olarak, OCR sisteminin performansı gerek verilerde, yazı bulma sistemimizden ayrı olarak test edilmiřtir. Bu adımda, daha nceden kullanıcı tarafından belirlenen yazı alanlarında OCR sistemi sınanmıřtır. Bylelikle, otomatik grnt iinde yazı yeri bulma problemi ile yazı tanıma performansları birbirinden ayrılmıř olacaktır. Ayrıca, testlerde yazının tanınmasında, bařarımı artırmak iin nerilen sistemlerin de (k-ortalamlar ve İstatistiksel Dil Modeli) birlikte alıřtırılması saėlanmıřtır. Deneyler veri kmesinden rastgele seilen 8 resim zerinde gerekleřtirilmiřtir. Yapılan deneyler sonucu, OCR sisteminin tek bařına, k-Ortalamlar ve/veya İDM ile birlikte kullanılmasına karřılık gelen sonular Tablo 12’de gsterilmektedir.

Tablo 12: OCR, k-ortalamlar ve İDM sonuları

#	# Karakter	OCR Bařarısı	OCR ve k-ortalamlar Bařarısı	OCR ve İDM Bařarısı (2 / 3 gram)	OCR, k-ortalamlar ve İDM Bařarısı (2 / 3 gram)
164	29	29	29	29 / 26	29 / 27
202	157	133	141	133 / 132	140 / 134
232	51	20	38	22 / 23	43 / 42
259	55	34	49	39 / 43	51 / 51
271	101	57	82	66 / 64	80 / 78
281	50	38	43	37 / 34	42 / 42
284	10	4	4	3 / 5	5 / 4
Toplam:	453	315	386	329 / 327	390 / 378

4.2.3.3.2 Sonular

Tablo 12’de gsterilen sonulardan fark edilebileceėi gibi k-ortalamlar metodu OCR sisteminin ve doėal olarak da tm sistemin performansını dikkat ekici bir řekilde ykseltmektedir. Tek bařına OCR’dan elde edilen sonuların bařarısı % 69.4 iken k-ortalamlar ile birlikte sistemin bařarısı % 84.7 seviyesine ıkmıř bulunmaktadır. Her ne kadar OCR ile birlikte İDM kullanıldıėında sonulardaki iyileřme K-ortalamlar kadar dikkat ekici olmasa da (İDM ile birlikte bařarım %72.6 seviyelerine ıkmaktadır), sonulardan da anlaşılacaėı gibi performansta bir miktar iyileřtirme saėlamaktadır. Burada dikkat edilmesi gereken nemli bir nokta, grntlerde bulunabilecek isim ve benzeri kelimelerin istatistiksel dil modeline uymama olasılıėının bulunmasıdır. Bu nedenle İDM kullanılarak bu kelimeler zerinde dil modeli bulunmadıėı veya bilinmediėi durumlarda dzeltmelere gidilemeyeceėidir. Bununla birlikte, istatistiksel dil modelinin elde edilmesi safhasında kullanılan iliřkiler ne kadar dili kapsayacak kadar geniř olursa, performans o kadar artmaktadır. řu an elde edilen sonularda her ne kadar iliřkiler mmkn olduėunca geniř olduėu sylenbilirse de dili tamamen kapsamadıėı da bir gerektir.

4.2.4 Referanslar

- [1] P. Letellier, M. Nadler and J.F. Abramatic. "The Telesign Project" *Proceedings of the IEEE*, 73:813-827, 1985.
- [2] G. Piccioli, E. De Micheli, P. Parodi and M. Campani, "Robust method for road sign detection and recognition", *Image and Vision Computing*, 14:209-254, 1996.
- [3] Mariano, V.Y.; Kasturi, R.; "Detection of text marks on moving vehicles", Proceedings of Seventh International Conference on Document Analysis and Recognition, vol.1 393 – 397 2003.
- [4] E. Lee, P. K. Kim, and H. J. Kim, "Automatic recognition of a car license plate using color image processing", *Proceedings of ICIP*, 301-305, 1994.
- [5] S.K. Kim, D.W. Kim and H.J. Kim, "A recognition of vehicle license plate using a genetic algorithm based segmentation", *Proceedings of ICIP*, 661-664, 1996.
- [6] Y. Dai, N. Zheng, X. Zhang and G. Xuan, "Automatic recognition of province name on the license plate of moving vehicle", Proceedings of ICPR, 927-929, 1988.
- [7] J. Zhou and D. Lopresti, "Extracting text from WWW images", Proceedings of ICDAR, 248-257, 1997.
- [8] J. Zhou, D. Lopresti and T. Tasdizen, "Finding text in color images", Proceedings of SPIE, Document Recognition V, 130-140, 1998.
- [9] H. Li, D. Doermann, and O. Kia, "Automatic Text Detection and Tracking in Digital Video," IEEE Transactions on Image Processing, 9(1): 147-156, 2000.
- [10] H. Li and D. Doermann, "A Video Text Detection System Based on Automated Training," In Proc. International Conference on Pattern Recognition, volume2, pages 223-226, 2000.
- [11] R. Lienhart and F. Stuber, "Automatic Text Recognition in Digital Videos," In Proceedings of SPIE, volume 2666, pages 180 188, 1996.
- [12] R. Lienhart and F. Stuber, "Automatic Text Recognition for Video Indexing," In Proceedings of the ACM International Multimedia Conference & Exhibition, pp. 11 20, 1996.
- [13] R. Lienhart, Effelsberg W., "Automatic Text Segmentation and Text Recognition for Video Indexing," Multimedia Systems, 8, pp. 69-81, 2000.
- [14] Dimitrova N., Agnihotri L., Dorai C., Bolle R., "MPEG-7 Videotext description scheme for superimposed text in images and video," Signal Processing: Image Communication, 16, pp. 137-155, 2000.
- [15] Zhong Y., Karu K., Jain A.K., "Locating Text in Complex Color Images," Pattern Recognition, 28(10), pp. 1523-1535, 1995.
- [16] H.K. Kim, "Efficient Automatic Text Location Method and Content Based Indexing and Structuring of Video Database," Journal of Visual Communications and Image Representation, 7(4): 336-344, December 1996.
- [17] Y. Ariki and T. Teranishi, "Indexing and Classification of TV News Articles based on Telop Recognition," In International Conference on Document Analysis and Recognition, volume 2, pages 422-427, 1997.
- [18] S. Kurakake, H. Kuwano, and K. Odaka, "Recognition and Visual Feature Matching of Text Region in Video for Conceptual Indexing," In Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases I, Vol. SPIE 1908, pages 368-379, 1997.
- [19] Jain A.K., Yu B., "Automatic Text Location in Images and Video Frames," Proc. IEEE Pattern Recognition, Vol. 31, No. 12, pp. 2055-2076, 1998.
- [20] Sato T., Kanade T., Hughes E.K., Smith M.A., Satoh S., "Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions," Multimedia Systems, 7, pp. 385-395, 1999.
- [21] T. Gandhi, R. Kasturi, and S. Antani, "Application of Planar Motion Segmentation for Scene Text Extraction," In Proc. International Conference on Pattern Recognition, volume 3, pages 445-449, 2000.
- [22] Wu V., Manmatha R., Riseman E.M., "TextFinder: An Automatic System To Detect and Recognize Text in Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 21, No. 11, November 1999.
- [23] Chaddha N., Sharma R., Agrawal A., and Gupta A., "Text Segmentation in Mixed-Mode Images," In 28th Asilomar Conference on Signals, Systems and Computers, pp. 1356-1361, October 1995.
- [24] Hauptmann A. and Smith M., "Text, Speech, and Vision for Video Segmentation: The Informedia Project," In AAAI Fall 1995 Symposium on Computational Models for Integrating Language and Vision, 1995.

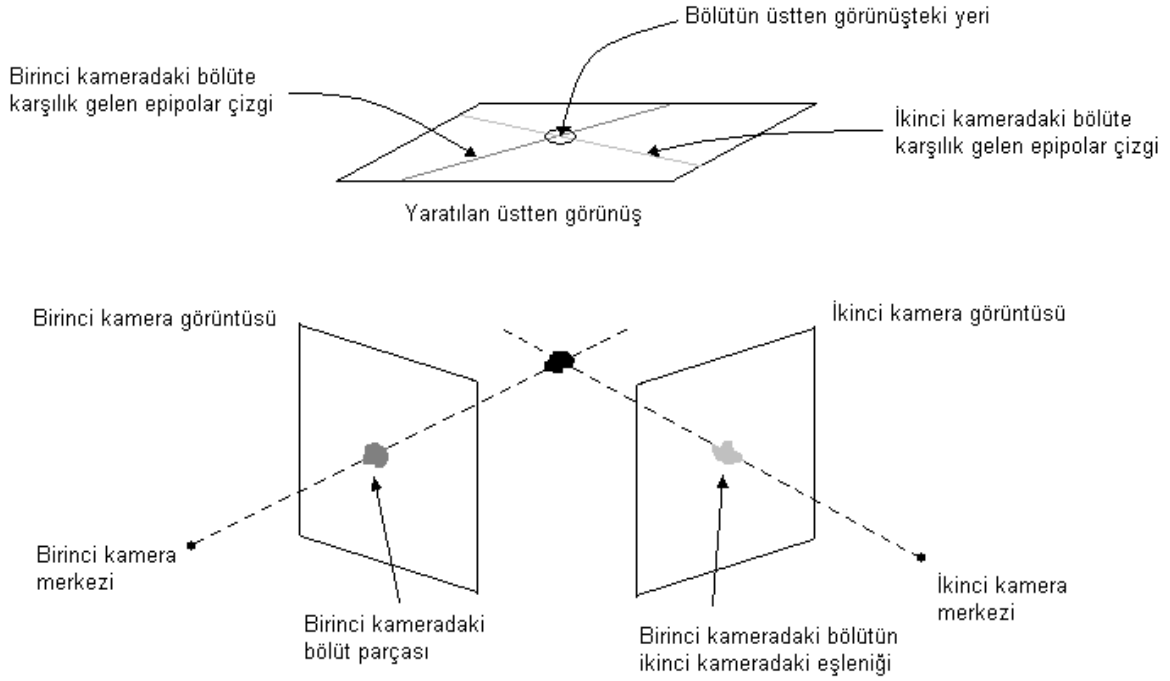
- [25] J. Bernsen, "Dynamic Thresholding of Grey level Images", ICPR'86: Proc. Int. Conf. on Pattern Recognition, Berlin, Germany, 1986, pp:1251-1255.
- [26] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram," Graphical Models and Image Processing, 29 (1985) 273-285
- [27] T. Abak, U. Barış B. Sankur, "The Performance of Thresholding Algorithms for Optical Character Recognition, Int. Conf. on Document Analysis and Recognition", ICDAR'97, Ulm., Germany, 1997, pp:697-700
- [28] B. S. Manjunath, Philippe Salembier, Thomas Sikora "Introduction to MPEG-7: Multimedia Content Description Interface", 2002
- [29] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification," 2nd Ed. Wiley, 2001
- [30] Teuvo Kohonen, Jussi Hynninen, Jari Kangas, Jorma Laaksonen, Kari Torkkola, "LVQ_PAK: The Learning Vector Quantization Program Package", 1995
- [31] Li. H., Doermann D., Kia O., "Automatic Text Detection and Tracking in Digital Video," Computer Vision Lab. Technical Repor, university of Maryland, CAR-TR-900, December 1998.
- [32] Xian-Sheng Hua; Liu Wenyin; Hong-Jiang Zhang, "An Automatic Performance Evaluation Protocol For Video Text Detection Algorithms", IEEE Transactions on Volume 14, Issue 4, April 2004 498 – 507.
- [33] Odobez, J.-M., Datong Chen, "Robust video text segmentation and recognition with multiple hypotheses" Proceedings of International Conference on Image Processing, Volume 2, 433-436, 2002.
- [34] M. D. Garris, "Component-Based Handprint Segmentation Using Adaptive Writing Style Model," NIST Internal Report 5843, June 1996.
- [35] M. D. Garris, J. L. Blue, G. T. Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, and C. L. Wilson, "NIST Form-Based Handprint Recognition System," NIST Internal Report 5469 and CD-ROM, July 1994.
- [36] P. J. Grother, "Karhunen Loève Feature Extraction for Neural Handwritten Character Recognition," NIST Internal Report 4824, April 1992, and in Proceedings of *Applications of Artificial Neural Networks III*, Vol. 1709, pp. 155-166. SPIE, Orlando, April 1992.
- [37] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "A Local Search Approximation Algorithm for k-Means Clustering" *Proc. of the 18th ACM Symp. on Computational Geometry*, 2002, 10-18.
- [38] Ramesh Jain, Rangachar Kasturi Ramesh, Brian G. Schunck "Machine Vision", 1995
- [39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Parallel Distributed Processing, Volume 1: Foundations, edited by D. E. Rumelhart, J. L. McClelland, et. al., MIT Press, Cambridge, pp. 318-362, 1986.
- [40] Neuhoff, D., "The Viterbi algorithm as an aid in text recognition" Information Theory, IEEE Transactions on Volume 21, Issue 2, Mar 1975 Page(s):222 – 226
- [41] P Clarkson, R Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit", Proc. Eurospeech, 1997.
- [42] G. David Forney, "The Viterbi Algorithm", Proceedings of the IEEE, 6:268-278, 1973
- [43] E. Forgey. Cluster analysis of multivariate data: Efficiency vs. interpretability of classification. Biometrics, 21:768, 1965.
- [44] D Chen, JM Odobez, H Boulard, "Text Segmentation and Recognition in Complex Background Based on Markov Random Field." Proc. IAPR 16th Int. Conf. Pattern Recognition, vol4. 227-230, 2002.
- [45] D Chen, JM Odobez, H Boulard "Text Detection and Recognition in Images and Videos" Research Report IDIAP - Pattern Recognition (2004).

4.3 ÖRTÜŞEN ARAÇLARIN BÖLÜTLENMESİNDE ÇOKLU KAMERA KULLANIMI

Günümüzde kullanılmakta olan bütün tek-kamera gözlem ve takip algoritmalarında 3 boyutlu nesnelere 2 boyut bilgisi kullanarak takip etme zorluğu yaşanmaktadır. Bu algoritmaların performansları tamamen ya da kısmen örtüşen nesnelere ve görünüşlerindeki değişimler yüzünden azalmaktadır. Bazı araştırmacılar bu sorunların çözülmesi için çoklu kamera gözlem ve takip algoritmaları geliştirmişlerdir. Böyle bir yaklaşımda [2] insan şekli analizi ve takibi kullanılarak insanların yerinin bulunması yapılmıştır. Diğer bir çözüm olarak [3] ise stereo bilgisini sisteme eklenmiştir. [4] nolu referansda tek kamera takip sistemlerini kameralar arası geçişi sağlayarak, görüntünün en iyi olduğu (örtüşmenin en az yaşandığı) kameranın bulunması geliştirmiştir. Diğer bir çoklu izleme sisteminde ise stereo, renk bölütlemesi ve yüz örüntü bulunması ile bir takip algoritması geliştirmiştir [5]. Diğer bir yaklaşımda, örtüşme sorunu stereo kamera kullanılarak çözülmeye çalışılmıştır [1]. Ayrıca, stereo kamera kullanılarak bulunan derinlik bilgisiyle cisimlerin yerlerinin bulunması üzerine kurulmuş bir algoritma önermiştir [6]. Bu raporda anlatılan çalışmada iki kameradan alınan görüntüler kullanılarak arka plandan bölütlenmiş hareketli cisimlerin renk benzerliği göz önüne alınarak ayrıştırıldıktan sonra, bu bölütlerin derinliklerinin bulunup kuş bakışı görünüşe yerleştirilmeleri önerilmiştir. Bu sayede nesnelere arası örtüşmenin olmadığı bir bakış açısı elde edilmeye çalışılmıştır.

4.3.1. Kuş bakışı Görünüş elde edilmesi :

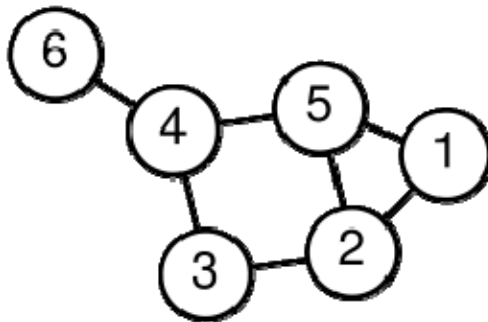
Örtüşen araçların bölütlenmesinde iki kamera görüntüsü kullanarak kuş bakışı görünüş oluşturulması denenmiştir. İlk önce arka plan hareketli cisim ayrıştırılması basit bir 'görüntü çıkarma' algoritması kullanarak yapılmış ve sonuç 'erosion' (kemirme) ve 'dilation' (genleşme) işlemleri kullanılarak temizlenmiştir. Hareketli cisimler (araçlar) arka plandan ayrıştırıldıktan sonra *döngülü en küçük kapsayan ağaç* (Recursive Shortest Spanning Tree, RSST) bölütleme algoritması kullanılarak küçük parçalara bölünmüştür. Daha sonra her küçük parçanın diğer görüntüdeki eşleniği bulunmuştur ve bulunan bölüt çiftlerinin orta noktalarının, yaratılan üstten görünüşteki epipolar çizgileri bulunmuştur. Daha sonra bulunan epipolar çizgiler kesiştirilerek bölüt çiftinin orta noktasının üstten görünüşteki yeri bulunmuştur. Bu algoritmalar kullanılarak nokta bulutlarından oluşan bir üstten görünüş elde edilmiştir. [1]'de yapılan üçgenleme ile derinlik bulma yerine, üst görüntüdeki epipolar çizgilerin kesiştirilmesi kullanılarak derinlik bulma problemi daha kolay ve yüksek başarıyla çözülmüştür. Bir sonraki adım bu nokta kümelerinin doğru bir şekilde gruplanmasıdır.



Şekil 77: Kuş bakışı görünüş yaratılması

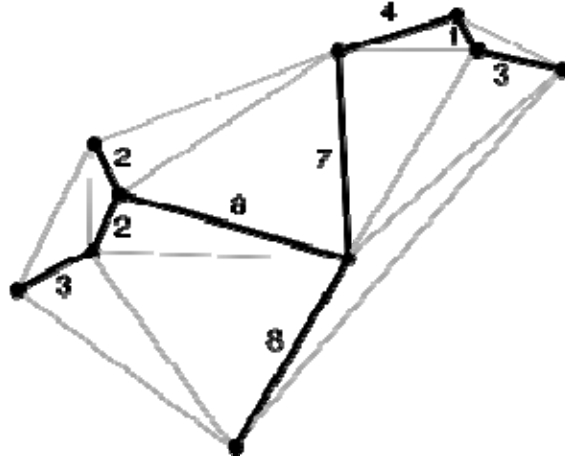
4.3.2. Nokta kümelerinin gruplandırılması

Noktaların gruplandırılması işlemi için 'en küçük kapsayan ağaç' algoritması kullanılmıştır. Çizge sonlu sayıda düğüm ve bunları bağlantılayan yaylardan oluşan bir diyagramdır. Normal bir çizgede, yönsüz varsayılan, A ve B noktalarını bağlayan yay ile B ve A noktalarını bağlayan yay aynıdır. Genellikle çizge, diyagram şeklinde gösterilirken düğümler için noktalar ve yaylar içinde çizgiler kullanılır.



Şekil 78: 6 düğümlü ve 7 bağlantılı bir çizge.

Verilen bir bağlı ve yönsüz çizgenin 'kapsayan' ağacı, tüm düğümleri kapsayan bir alt-çizgedir. Bir çizge birden fazla kapsayan ağaca sahip olabilir. Ayrıca her bağlantıya onun gücünü gösteren bir ağırlık vererek, sık kullanılan ya da muhtemel olan bağlantılar daha belirgin hale getirilmiş olur. Daha sonra, kapsayan ağaca içindeki bağlantıların ağırlıkları toplamına eşit olan bir ağırlık verilir. En küçük ağırlığa sahip kapsayan ağaca "en küçük kapsayan ağaç" denir.



Şekil 79: En küçük kapsayan ağaç örneği

En küçük kapsayan ağaç bulunurken Kruskal'ın algoritması [9] kullanılmıştır. Kruskal'ın algoritması bağlı ve ağırlıklandırılmış çizgenin en küçük kapsayan ağacını bulan bir çizge kuramı algoritmasıdır. Eğer çizge bağlı değilse, en küçük kapsayan ormanı (yani bağlı olmayan her grup için en küçük kapsayan ağaç) bulur. Kruskal'ın algoritması fırsatçı algoritmaların (Greedy algorithms) bir örneğidir.

Algoritma şöyle çalışır :

- Her düğümün ayrı bir ağaç olduğu orman (ağaçlar kümesi) yaratılır
- Çizgedeki tüm bağlantıları içeren bir S kümesi yaratılır
- S boş küme değilken
 - o En küçük ağırlıklı bağlantı S kümesinden çıkarılır
 - o Eğer çıkarılan bağlantı iki farklı ağacı bağlıyorsa, iki ağacı tek ağaç yapacak şekilde ormana eklenir
 - o Eğer iki farklı ağacı bağlamıyorsa atılır.

Algoritmanın çalışması bittiğinde, ormanın sadece bir bileşeni olur ve çizgenin en küçük kapsayan ağacı oluşmuş olur [7],[8],[9].

4.3.3. Yönsüz çizgenin ve en küçük kapsayan ağacın oluşturulması

Bölüt çiftlerinin epipolar çizgileri kesiştirilerek bulunan kuş bakışı görüntünün noktaları yönsüz çizgenin düğümleri olarak tanımlanır. Sistemin daha hızlı çalışması için her NxN lik bölge içinden sadece bir düğüm seçilir. Daha sonra düğümler arasındaki bağlantıların ağırlığı şu şekilde tanımlanır;

i ve j düğümleri arasındaki bağlantının ağırlığı

$$W_{ij} = \alpha H_{diff} + \beta D_{diff}$$

H_{diff} noktaların hue değerleri arasındaki fark ve D_{diff} noktaların birbirlerine olan Öklit uzaklıklarıdır.

Bu ağırlık fonksiyonu kullanılarak birbirlerine bağlanan benzer renge sahip yakın noktalar çizge oluşturur. Daha sonra Kruskal'ın algoritması kullanılarak çizgenin en küçük kapsayan ağacı bulunur ve belli bir eşik değerinin üzerindeki bağlantılar koparılarak örtüşen araç ayrıştırılması tamamlanır.

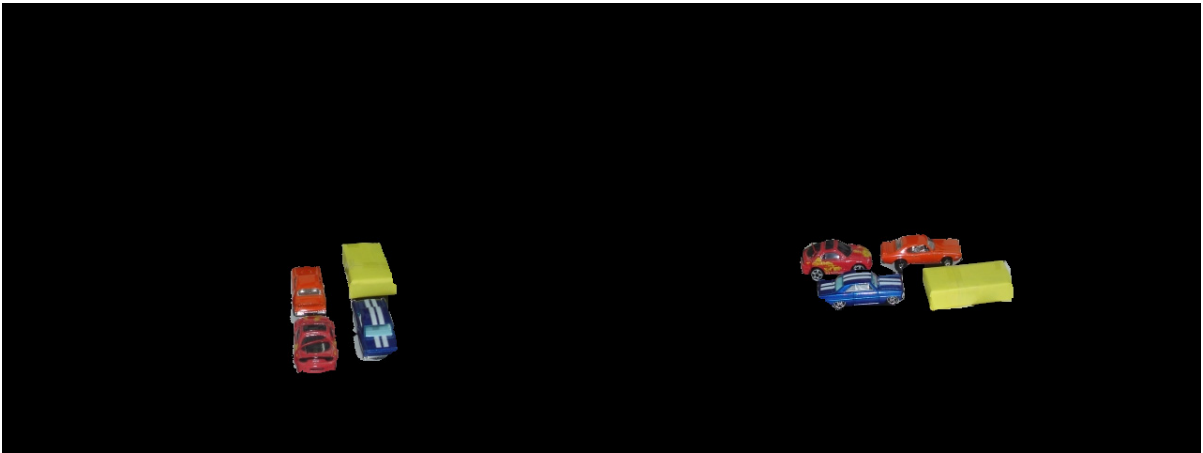
4.3.4 Deney sonuçları

Deneylerde kullanılan görüntülere ait tipik örnekler şu şekildedir (Şekil 80).:



Şekil 80: Kamera-1 ve Kamera-2 görüntüleri

Önerilen algoritma sonucu hareketli cisim ve arka plan bölütlemesi sonunda bulunan maskeler aşağıda gösterilmiştir (Şekil 81).



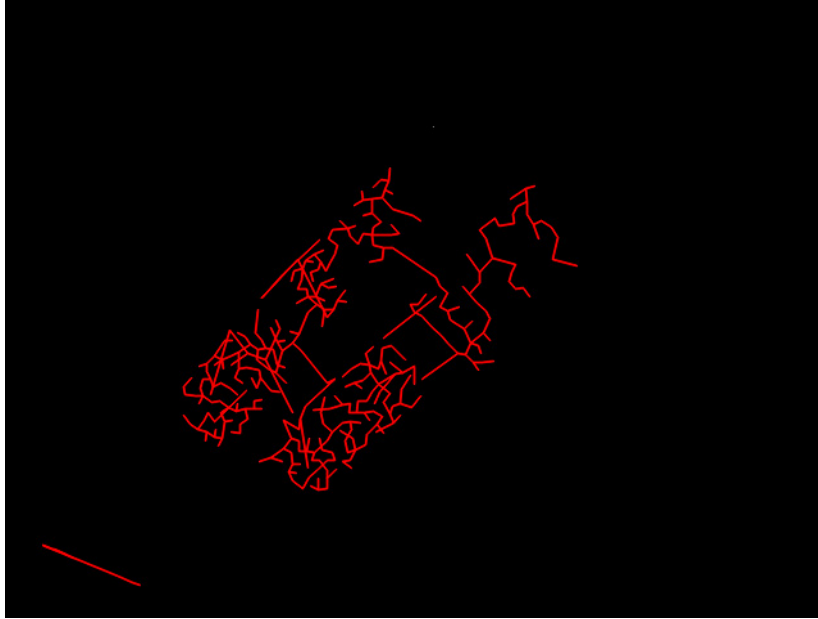
Şekil 81: Hareketli cisimlerin bölütleme sonuçları

Her bölüt çiftinin orta noktaları kullanılarak bulunan üst görüntü Şekil 82'de gösterilmiştir.



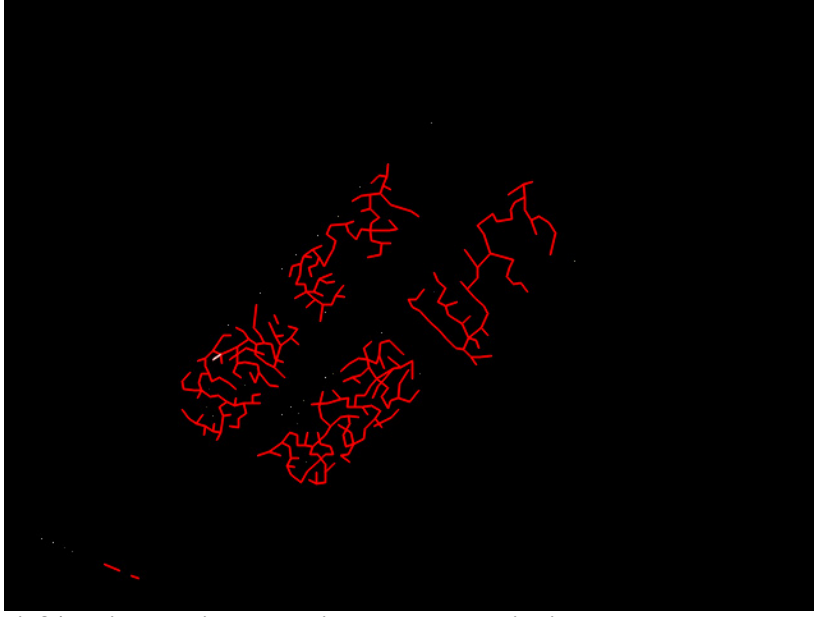
Şekil 82: Kuş bakışı görünüş

Kuş bakışı görüntüye ait oluşturulan en küçük kapsayan ağaç Şekil 83'de sunulmaktadır.

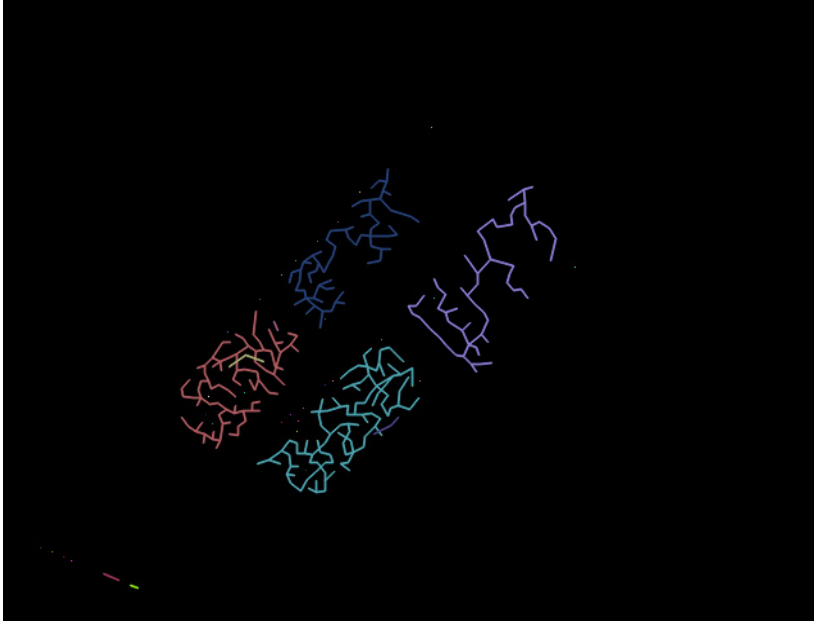


Şekil 83: En küçük kapsayan ağaç

Belirli bir eşik değeri kullanılarak en uzak bağlantıları kesilmiş ağacın örneği Şekil 84'de verilmiştir.



Şekil 84: Eşik değeri üstündeki bağlantıları kesilmiş en küçük kapsayan ağaç



Şekil 85: Gruplanmış noktalar

Eşik değeri ile bölünen ağaçlar içinde alt ağaçların oluşmasıyla bölütleme tamamlanmış olmaktadır (Şekil 85).

4.3.6. Sonuçlar

Örtüşen cisimler nesne takip sistemlerinin performanslarını düşüren başlıca nedenlerden biridir. Şu ana kadar yaptığımız çalışmalar örtüşmenin gerçekleştiği görüntülerde nesnelerin doğru şekilde ayrıştırılmasına yöneliktir. Bu sayede, örneğin, trafikteki araçların doğru ayrıştırılması veya izlenen sahnede yer alan örtüşen insanların yerlerinin bulunmasına kadar birçok alanda kullanılacak örtüşen cisimlerin ayrıştırılması algoritması geliştirilmiştir.

Gelecek çalışmalarımızda, yukarıda anlatılan algoritma ile yerleri doğru olarak bulunmuş örtüşen cisimlerin çoklu kamera sistemleri kullanılarak takip edilmesi planlanmaktadır.

4.3.7. Referanslar

[1] Anurag Mittal and Larry Davis: "Unified Multi-Camera Detection and Tracking Using Region Matching," Proceedings of the IEEE Workshop on Multi-Object Tracking, Vancouver, BC, Canada, July 2001

[2] I. Haritaoglu, D. Harwood, L. Davis. "W4:Who, When, Where, What: A Real Time System for Detecting and Tracking People." Third International Conference on Automatic Face and Gesture, Nara, April 1998.

[3] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4S: A realtime system for detecting and tracking people in 2 1/2D," 5th European Conference on Computer Vision, Freiburg, Germany, 1998.

[4] Q. Cai and J.K. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized video Streams," 6th International conference on Computer Vision, Bombay, India, 1998, pp. 356-362.

[5] T. Darrel, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, color, and Pattern Detection," IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA 1998, pp. 601-608.

[6] "Multi-camera multi-person tracking for EasyLiving," J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer. 3rd IEEE International Workshop on Visual Surveillance, 2000.

[7]Eppstein, David (1999). "Spanning trees and spanners". Handbook of Computational Geometry: 425–461, Elsevier.

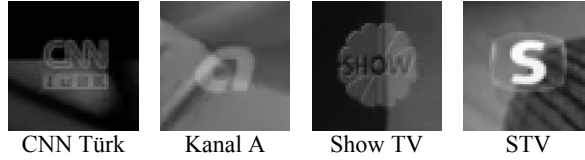
[8]Garey, Michael R.; Johnson, David S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman. ISBN 0-7167-1045-5. A2.1: ND2, pg.206.

[9]Wu, Bang Ye; Chao, Kun-Mao (2004). Spanning Trees and Optimization Problems. CRC Press. ISBN 1584884363.

4.4 TV YAYINLARINDAN LOGO TESPİTİ

4.4.1 Giriş

Televizyon (TV) yayın kuruluşları yayınladıkları görüntüler üzerinde sahiplik belirtmek ve yayını reklam, sinema, haberler gibi türlerle ifade etmek için TV logoları kullanırlar. Bu nedenle yayın üzerindeki logoların otomatik olarak tespiti reklam atlama [1], logo silme [2] gibi uygulamalar için temel bir aşama olarak ortaya çıkmaktadır. TV kanallarında, arka planın görünmesine izin veren ve yayın üzerinde daha az dikkat çeken yarı-saydam logoların kullanımına yönelim gözlenmektedir. TV kanallarının yarı-saydam logo kullanımındaki belirgin artış, ancak yakın zamanda üzerinde çalışılmaya başlanmış olan yarı-saydam logo tespiti sorununu önemli kılmaktadır. Çeşitli kanalların yayınlarında gözlenmiş örnek yarı-saydam logolar Şekil 86'da görülmektedirler.



Şekil 86: TV kanallarından örnek yarı-saydam logolar

Renk bilgilerini sürekli koruyan opak logoların aksine, yarı-saydam logolar görüntülerini arkaplan içeriğine bağlı olarak değiştirirler. Sonuç olarak, opak logolar için tasarlanmış, renk bilgisinin tutarlılığına dayalı tespit ve takip algoritmaları, yarı-saydam logolar için uygulanabilir değildir. Yarı-saydam logoların otomatik tespitine yönelik girişimler yeni olmakla beraber, çoğunluğu öntanımlı logo çevritinin varlığı ve yokluğunun algılanmasına dayanır.

Wang [3], birçok video karesinde uzam-zamansal gradyan olan genellenmiş gradyan üzerinden şablon eşleme önerir. Bu yöntem arkaplan içeriğine çok hassastır ve kabul edilir başarıma ulaşabilmek için uzun eşzamanlama aralıkları gerektirmektedir. Bu çalışmada [3] üç saniyelik zamansal biriktirme önerilmektedir. Diğer yandan Santos [4], zamansal olarak biriktirilmiş karelerde sadece uzamsal gradyan üzerinden şablon eşleme önermektedir. Bu yöntem ise yarı-saydam logoların görünmesinden beş saniye sonra tespit gerçekleştirmektedir. Bu iki yöntem dışında yarı-saydam logoların tespitini opak ve yarı-saydam logo ayrımı gözetmeden yapan olasılıksal yöntemler bulunmaktadır. Ekin'in [5] önerdiği yöntemde logolar görüntüde aykırı değerler olarak modellenmekte ve tespitlerinde olasılıksal yollar kullanılmaktadır. Duffner ve Garcia [6] yarı-saydam logoların tespiti için evrimsel sinir ağları tabanlı çoklu çözünürlük yaklaşımı önermektedir. Eğitim aşamasındaki yetersizlik nedeniyle, bu yöntem yüksek başarımları elde edememektedir.

Bu bildiriye yarı-saydam TV logolarının gerçek zamanlı tespiti için logo sınırları üzerinden eşleme yapan ve bu sınırların dolaylarındaki değişimleri kullanan bir yöntem önerilmektedir. Video karesinin ışıklılığı üzerinden Laplas işleci ile elde edilmiş kutuplu ayrıt haritası logonun çevritlerini belirlemek için kullanılmaktadır. Ayrıt haritalarının uzamsal bağıntılarının karşılaştırılması yerine, aranan logonun önceden bilinen çevrit bilgisi ile karenin ayrıt haritasından çıkarılan sınır bilgisi arasında küme karşılaştırma tabanlı karar verme kuralları önerilmektedir. Ek olarak, arkaplan gürültüsünün etkisini azaltmak için zamansal biriktirme yapılmaktadır. Ana algoritma logo giriş/çıkış sürelerini yüksek kesinlikte anlamaya yarayan ve doygunluk meydana geldiğinde logo takibinin doğru olarak devam etmesini sağlayan yollarla desteklenmektedir.

4.4.2 Önerilen Yarı Saydam Logo Tespit Yöntemi

4.4.3 Yarı-Saydam Logo Modeli

İlk olarak bu bölümde, yarı-saydam logoların görüntülere eklenmesinin matematiksel modelini sunuyoruz. Bu model aracılığı ile doymuluk ve evrilme olarak adlandırdığımız özel durumları tanımlıyoruz. L eklenecek logonun ışıklılık haritası, SL logonun dayanağı ve I üzerine logo eklenecek karenin ışıklılık kanalı olsun. Yarı-saydam logo içeren yeni ışıklılık kanalı, bu iki sinyalin basit birleşimi (1) ile ifade edilebilir [7].

$$\hat{I} = \alpha(t, x, y)L + (1 - \alpha(t, x, y))I \quad (1)$$

Burada α ;

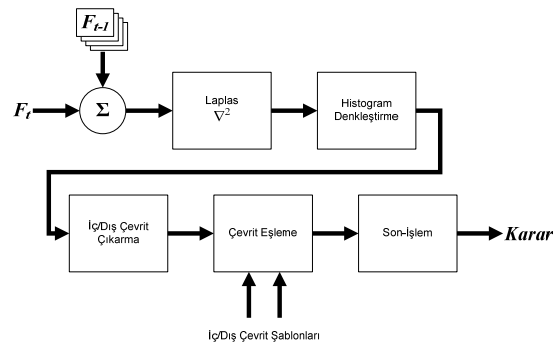
$$\alpha(\bullet, x, y) = 0, \quad (x, y) \notin S_L \text{ ise}$$

$$0 < \alpha(\bullet, x, y) \leq 1, \quad (x, y) \in S_L \text{ ise}$$

özelliklerini sağlayan geçiş fonksiyonudur.

Bu modelden görüldüğü gibi logonun saydamlık derecesi α parametresi ile denetlenir. Tamamen opaklık $\alpha=1$ durumunda elde edilir; $\alpha=0$ durumunda logo tamamen gözden kaybolur. Arkaplanı türdeş değiştirebilmek için ise α ve L genellikle sabit alınır. Bir başka deyişle α ve L uzam-zamansal boyutlarda değişmezler.

Yarı-saydam bir logonun video karesine bindirilmesi sonucunda SL sınırlarında ayrıtlar oluşur. Bu ayrıtların kuvveti $I(x,y)$ ve L arasındaki farka bağlıdır. İkisi eşit olduğunda ise ayrıtların görünmez hale geldiği doymuluk durumu oluşur. Bu yüzden ayrıt eşleme doymuluk durumları için yeterli olamaz. Ancak tamamlayıcı buluşsal yöntemlerle bunun üstesinden kolaylıkla gelinebilir. $I(x,y)$ ve L'nin değerlerine bağlı olarak \hat{I} , I'dan büyük ya da küçük olabilir. $\hat{I} < I$ iken yarı-saydam logo civarından daha koyu görünür. Evrilme adı verilen bu durum, ayrıt kutuplarının hesaba katıldığı yöntemlerde özel çözümler gerektirir.



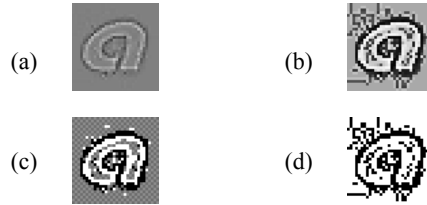
Şekil 87: Tespit algoritmasına genel bakış.

4.4.4 Önerilen Algoritma

Önerilen yöntem video dizisi üzerinde uygulanır ve yarı-saydam logo tespiti her video karesinde önsel logo sınır bilgisi kullanılarak yapılır. Bu bilginin çıkarılması, logonun video üzerinde bulunduğu ilgi bölgesinden EL ve IL isimli iki küme piksel konaçlarının elle belirlenmesini içerir. Bu kümelerden EL, logonu arkaplanla buluştuğu dış çevrit üzerindeki piksellerin konaçlarını kapsar. IL ise logonun iç çevritindeki piksellerden oluşur. Bu iki farklı

çevritin kullanılması ile logo sınırlarının ayırmsama gücünün artırılmasını hedeflemektedir. Çevrit yerelliklerini hesaba katmayan geleneksel bir ayrıt tanıma yöntemi, L ve I birbirlerine yakın değerler aldığıında başarısız olacaktır. Önerilen yarı-saydam logo tanıma yöntemi Şekil 87'de özetlenmektedir.

Logo sınırları, yarı-saydam logolar için tek dayanıklı bilgi kaynağı olduğu için, logo sınırlarının bulunmasını sağlayan güvenilir algoritmalar kullanılmalıdır. Video içeriği zamanla değiştiğinden logo çevritlerinin dayanıklı kalmaları beklenir. Bu yüzden tespit algoritmasında, çevritlerin ayırmsama gücü öncelikle zamansal biriktirme yapılarak artırılır. Hatırlatılması gereken nokta, biriktirmede kullanılan karelerin sayısı arttıkça logoların görünme ve kaybolma anlarını algılamadaki keskinliğin düşmesidir. Bu da algoritmanın verdiği geçiş kararlarının doğruluktan sapmalarına neden olur. Laplas işlecinin öntanımlı sayıda biriktirilmiş bu karelerdeki sonucu üzerinde histogram denkleştirme yapılırsa aday çevrit bölgeleri güçlendirilmiş olur. Elde edilen bu imgeye G_t diyelim. G_t 'deki yüksek değerler, etraftan yüksek ışıklılığa sahip pikselleri belirtir. Benzer şekilde düşük G_t değerleri, video karesinde komşularından daha az ışıklılığa sahip piksellere denk gelir. Şekil 88.a'da örnek bir kareden elde edilen ayrıtlar gösterilmektedir. Şekil 88.b'den de görüldüğü gibi histogram denkleştirme bu ayrıtları daha belirgin hale getirmektedir.



Şekil 88(a) Videodan elde edilen ayrıtlar ve (b) histogram denkleştirme sonucu ortaya çıkan çevritler. (c) ve (d) eşikleme sonrası önerilen çevrit konaçlarını göstermekte. Beyaz pikseller iç çevriti, siyahlar ise dış çevriti temsil etmektedirler.

Video karesinden dış ve iç çevritleri elde edebilmek için G_t eşik değerleri ile karşılaştırılır. Aday dış çevrit konaçları için $G_t(x,y) \geq T_1$ ve aday iç çevrit konaçları için $G_t(x,y) \leq -T_1$ olacak şekilde bir T_1 değeri öntanımlanır. Tek bir eşik değeri kullanmanın sebebi dış ve iç çevrit kuvvetlerinin sıfır etrafında simetrik oldukları varsayımıdır. Bu iki konaç kümesi sırasıyla E_{ft} ve I_{ft} ile ifade edilir. T_1 'in gerçek değeri, bir sonraki bölümde bahsedilen çevrit eşleme yöntemlerinden hangisinin kullanıldığına bağlıdır. Şekil 88.c ve Şekil 88.d iki farklı T_1 seçimi sonucunda elde edilmiş çevritleri göstermektedirler. T_1 eşik değeri olarak sıfır kullanmak, ilgi bölgesindeki tüm pikselleri ya iç ya da dış çevritin bir parçası olarak seçmeyle özdeşir.

4.4.5 Çevrit Eşleme

Çevrit eşleme basamağında, E_{ft} ve I_{ft} , E_L ve I_L ile karşılaştırarak temsil ettikleri logo sınırlarının benzerliği ölçülür. Bu benzerlik ölçütü, ilgi bölgesindeki tüm olası logo konumları üzerinde basit küme karşılaştırması yapılarak hesaplanır. Kareden elde edilen iki küme ve öntanımlı logo çevritini kullanan şu ölçütler tanımlanmıştır;

$$S_1(p) = \frac{n((E_{ft} \cup I_{ft}) \cap (E_L \cup I_L))}{n(E_L \cup I_L)} \quad (2)$$

$$S_2(p) = \frac{n(E_{ft} \cap E_L) + n(I_{ft} \cap I_L)}{n(E_L \cup I_L)} - \frac{n(E_{ft} \cap I_L) + n(I_{ft} \cap E_L)}{n(E_L \cup I_L)} \quad (3)$$

Burada p ilgi bölgesi üzerinde karşılaştırma yapılan konumu ve $n(A)$ da A kümesinin eleman sayısını ifade eder. S_1 , (2)'den görüldüğü gibi, karşılıklı tüm kümeler arasındaki eşlemleri tercih eder. Bununla birlikte, S_2 ise farklı çevrit sınıfları arasındaki eşlemleri olumsuz katkı olarak kabul ederken sadece aynı çevrit sınıfları arasındaki eşlemleri tercih eder. İki benzerlik ölçütü için de daha yüksek değerler, karede yarı-saydam logonun bulunduğu daha kuvvetli bir göstergedir. Herhangi bir p konumu için kullanılan ölçütün değeri öntanımlı bir eşik olan T_2 'yi geçerse o karede logonun bulunduğu karar verilir, aksi durumda ise logonun bulunmadığı sonucu elde edilir. Logonun konumu ise, p 'nin bütün değerleri arasında en yüksek benzerlik değerini elde edendir.

Deneyler sırasında, yarı-saydam logonun bazı karelerde arkaplandan daha koyu olduğu gözlenmiştir. Logonun iç ve dış çevrit tanımında bir evrilme meydana getiren bu durumu kotarabilmek için S_2 üzerinde şu değişiklik yapılır;

$$S'_2(p) = |S_2(p)| \quad (4)$$

Bu basit değişiklik, çevrim durumunda iç çevritin dış, dış çevritin de iç olarak çıkarılmasını sağlar. Evrilme durumuna örnekler Şekil 89'da gösterilmektedirler.

Bunlarla birlikte, doygunluk oluştuğunda logo sınırları çıkarılamaz; bu yüzden logo varlığı hakkında karar verilemez. Bu sorunu kotarmak için öncelikle doygunluğun meydana geldiği tespit edilir. İlgili bölgesinde görel olarak düşük miktarda ayrıt gözlenmişse, L ve ilgi bölgesinin ortalama ışıklılığı arasındaki fark eşiklenir. Eğer bu fark öntanımlı bir eşik altındaysa doygunluk oluştuğu kararı verilir. Bundan sonra, doygunluğun devam ettiği tüm aralıkta logonun varlığı hakkındaki kararlar doygunluk başlamadan önce verilen kararlar değiştirilir.

Tespit başarımını artırmak için çoklu kare ile karar yöntemi kullanılır. Logonun varlığı ya da yokluğu bir müddet boyunca art arda değişmeyeceği için, algoritmanın kısa aralıklar için elde ettiği kararsız sonuçlar pencereleme yöntemi ile kararlı hale getirilebilir. Eğer logo var ya da logo yok kararı öntanımlı bir süre penceresi boyunca kararlı kalmayı başaramamışsa hemen öncesinde verilmiş kararlar kullanılır. Pencereleme yöntemi karar verme aşamasında gecikmelere neden olurken, logo giriş ve çıkış sürelerinin keskin olarak belirlenmesine engel olmamaktadır.

TV kanalları yaygın olarak, yarı-saydam logo ile aynı sınırlara sahip opak bir logoyu başka yayın tipleri için kullanmaktadır. Bu yüzden algoritmaya aynı sınırlara sahip biri opak diğeri yarı-saydam iki logonun ayırt edilebilmesi için ek soniştirme kuralları eklenmiştir. Bu amaç doğrultusunda yarı-saydam logoların, opak logoların aksine, üzerine eklendikleri karelerin renklilik kanallarını önemli ölçüde değiştirmedikleri gözlemi kullanılmıştır. Dolayısıyla karenin önce Y kanalını kullanarak bir logonun varlığı ya da yokluğu tespit edildikten sonra U ve V kanalları da benzer şekilde işlenmektedirler. Eğer renklilik kanallarında kayda değer ölçülerde çevrit eşlemi gözlenirse tespit edilen logo opak olarak işaretlenir, aksi takdirde yarı-saydam logonun bulunduğu sonucuna ulaşılır.



Şekil 89: CNN Türk logosu üzerinde (a) tüm evrilme ve (b) kısmi evrilme.

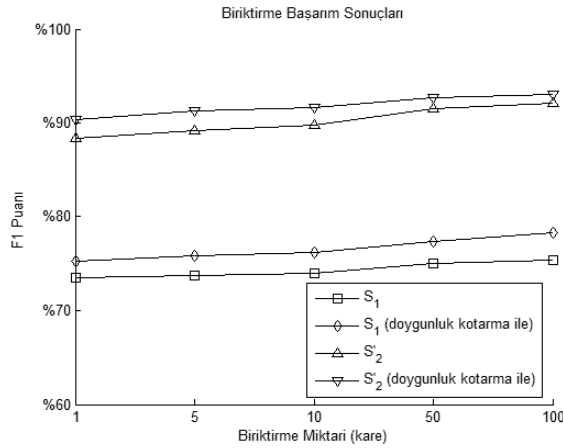
4.4.6 Deneyler

Bütün deneyler için *CNN Türk*, *Kanal A*, *Show TV* ve *STV* isimli kanalların yarı-saydam reklam logoları kullanılmıştır. Bütün test videoları MPEG-4 H.264 kodlu olup CIF çözünürlüğe, 25fps kare hızına ve saniyede 300kbit veri yoğunluğuna sahiptir [8].

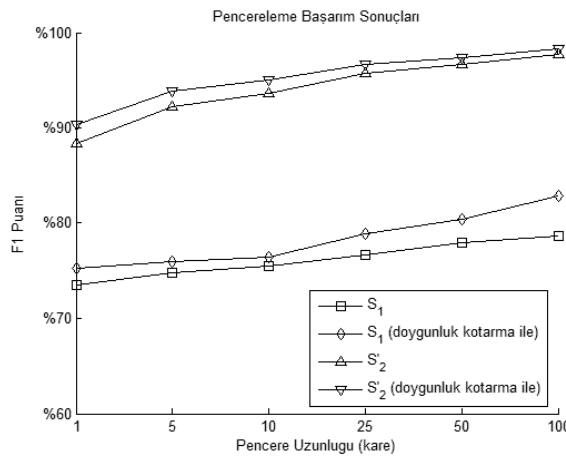
İki farklı deney grubu gerçekleştirilmiştir. İlk kısım, dört kanal için de eşit olarak dağıtılmış yaklaşık 18 saatlik bir deney verisi üzerinde gerçekleştirilmiştir. Deneylerin bu aşamasında ayrı olarak zamansal birikim ve pencerelemenin başarımlar üzerindeki etkisi gözlenmiştir. Her deney, S1 ve S'2 benzerlik ölçütleri ile birlikte, hem doyma kotarması kullanılarak hem de kullanılmadan yürütülmüştür. Doğruluk ölçütü için verilen F1 puanlama yöntemi kullanılmıştır;

$$F1 = \frac{2 \times \text{Kesinlik} \times \text{Hatırlama}}{\text{Kesinlik} + \text{Hatırlama}} \quad (6)$$

Şekil 90'da görüldüğü gibi zamansal biriktirme iki benzerlik ölçütü için de başarımlarını artırmaktadır. Pencerelemenin başarımlarına etkisi ise Şekil 91'den rahatlıkla görülmektedir. Bunların dışında, doymuluk kotarmanın algoritmanın başarımlarını yükselttiği kanıtlanmıştır. Benzerlik ölçütü olarak S'2'nin kullanılması tüm sistemin ayımsama gücünü S1'in kullanıldığı durumlara kıyasla çok daha fazla artırmaktadır. Bu sonuç da arkaplanın ilgi bölgesinde uzamsal olarak türdeş özellikler gösterdiği ve kısmi evrilmeden çok tüm evrilme gerçekleştiği yargılarını desteklemektedir.



Şekil 90: Biriktirme başarımlar sonuçları.



Şekil 91: Pencereleme başarımlar sonuçları.

Algoritma, geciktirme kullanılmadığı durumlarda %90.38 F1 puanı elde etmektedir. Bu başarımların, Wang'ın [3] yönteminin 4 saatlik veri ve bir kısmı opak olan logolar ile elde ettiği sonuçların oldukça üzerindedir. Bu açıkça uzamsal bölgede sınır eşleme tabanlı yarı-saydam logo tespiti için uygulanabilirliğini göstermektedir. Tespit başarımlarını, 4 saniyelik pencereleme kullanıldığında %98.25'e ulaşmaktadır.

Deneylerin ikinci ayağında aynı sınırlara sahip opak ve yarı-saydam logoların ayırt edilmesi incelenmiştir. Kullanılan veri kümesi, opak ve yarı-saydam logoları aynı sınırlara sahip olan Kanal A, Show TV ve STV'den 500,000'er karelik yayını kapsamaktadır. Opak logoların sınırlarında değişken ışık ve renk değerleri bulunduğu için oluşturdukları çevritler kutupsal olarak tutarlı değildir. Bu durum, S'2 benzerlik ölçütünü bu ayırım için kullanılamaz kılar, bu yüzden sadece S1 kullanılmıştır. Her bir kanal için elde edilen sonuçlar Tablo13'de gösterilmektedir. Kanal A için elde edilen düşük kesinlik değeri kanalın tamamen beyaz olan opak logosunun renklilik bileşeni içermemesinden kaynaklanmaktadır. Önerilen ayırımsama yöntemi ise opak logoların U ve V kanallarında bileşenler gerektirmektedir.

Tablo 13: Opak ve yarı-saydam logo ayırımı..

	Kesinlik	Hatırlama	F1
Kanal A	%16.81	%96.05	%28.61
Show TV	%100.00	%99.87	%99.94
STV	%93.33	%86.47	%89.77

4.4.7 Sonuçlar

Uzamsal bölgede çalışan gerçek zamanlı bir yarı-saydam TV logosu tespit etme yöntemi sunulmuştur. Logo sınırları yarı-saydam logolar için en ayırıştırıcı özellik olduğu için bu yöntem sınır eşleme tabanlıdır. Sorunlu durumların üstesinden gelebilecek yollar önerilmiş ve bu yolların başarımlar üzerindeki etkileri incelenmiştir. Ek olarak, bu yöntemle birlikte yarı-saydam logolar ile aynı sınırlara sahip renkli opak logolar kolayca ayırıştırılabilmektedir. Bu da bu yöntemin sadece yarı-saydam değil, opak logoların tespiti için de kullanılabileceğini göstermektedir.

4.4.8 Referanslar

- [1] Albiol, A.; Ch, M. J.; Albiol, F. A.; Torres, L.; "Detection of TV Commercials", IEEE ICASSP 2004.
- [2] Meisinger, K.; Troeger, T.; Zeller, M.; Kaup, A.; "Automatic TV Logo Removal Using Statistical Based Logo Detection and Frequency Selective Inpainting", EUSIPCO, 2005.
- [3] Wang, J.; Duan, L.; Li, Z.; Liu, J.; Lu, H.; Jin, J. S.; "A robust method for TV logo tracking in video streams," IEEE ICME 2006.
- [4] Santos, A. R.; Kim, H. Y.; "Real-time opaque and semi transparent TV logos detection," I2TS, 2006.
- [5] Ekin, A.; Braspenning, E.; "Spatial detection of TV channel logos as outliers from the content," Proc. Visual Communications and Image Processing. SPIE, 2006.
- [6] Duffner, S. and Garcia, C. "A neural scheme for robust detection of transparent logos in TV programs," ENNS International Conference on Artificial Neural Networks, 2006.
- [7] Porter, T. and Duff, T., "Compositing digital images." SIGGRAPH '84. ACM, 1984.
- [8] Deney Verisi için İnternet Adresi <http://vms.bilten.metu.edu.tr/Data/TranslucentLogo/>

4.5 VİDEO KOPYASI BULMA

Son yıllardaki çoğulortam teknolojileri ve teleiletişim teknolojileri gibi alanlardaki hızlı gelişimin sonucu olarak sayısal video içeriğine ulaşımın kolaylığı artmıştır. Veri transfer hızındaki inanılmaz artış, daha verimli çoğulortam kodlama tekniklerinin bulunması ve daha birçoğu sayısal video içeriğinin artışında önemli bir katalizör olarak ortaya çıkan “video ürün konakçılarının” (video hosting services) doğuşuna sebep olmuştur. Bahsedilen hizmet sağlayıcılarına örnek olarak YouTube, Google Video ve Metacafe verilebilir. Sayısal verilerin miktarındaki üstel artış – Ağustos 2006’da YouTube’de 6.1 milyon video hizmete sunulurken Nisan 2008 tarihinde bu sayı 83.4 milyona ulaşmıştır [1] – iki önemli problem ortaya çıkmaktadır; *teelif haklarının korunması ve birçok suret video*.

İlk problem, video ürün konakçılarının popülaritesini arttıran temel nedenlerden biri olan içeriğin kullanıcı tarafından yaratılmasının bir yan ürünü olarak ortaya çıkmaktadır. Kullanıcılar, konser görüntüleri, müzik videoları, televizyon şovları ve filmlerden alıntılar gibi potansiyel olarak telif hakkı çiğneyebilecek video verilerini yükleyebilmektedir. Kullanıcı ve video sayısındaki artış ise birbirinin sureti/kopyası olan video sayısını ve bunun sonucu olarak da ürün konakçılarının depolama ve bant genişliği gereksinimlerini arttırmaktadır bu da doğal olarak şirketlerin masraflarını yükseltmektedir.

Bahsedilen sorunların çözümü için aktif ve pasif metotlar olmak üzere iki farklı çözüm yolu mevcuttur. Gizli damgalama temelli yöntemler en sık başvurulan aktif sezim yöntemidir. Ancak damgalama yöntemlerinin kullanılmasından doğan birçok kısıtlama ortaya çıkmaktadır. Bunlardan özellikle iki kısıt bahsedilen sorunlar için önemli bulunmaktadır. Öncelikli olarak, damgaların orijinal görüntüye kopyaların yaratılmasından önce konması gerektiğinden, bu yöntemle halihazırda dolaşımda olan içeriğin takip edilmesi imkansızlaşmaktadır. İkincil olarak, belirli bir videodaki damganın varlığının çeşitli ataklar ile tehlikeye düşmesi veya silinmesi durumunda herhangi bir alternatif sezim imkanı kalmamaktadır. Pasif sezim yöntemleri ise, telif hakkı ihlallerinin ya da suret videolarının bulunmasını, sorgulanan veri ile bir veritabanının karşılaştırılması ile gerçekleştirmektedir. Bu yaklaşım damgalama yöntemlerini tamamlayıcı bir teknoloji olarak düşünülebilir. Yaklaşımın temel aldığı düşünce verinin kendisinin damga özelliği taşıdığıdır. Bir diğer deyişle, verinin (imge, ses ve video) suretlerinin sezimi için kendi içerisinde yeterince özgün bilgi içerdiğini kısaca bir *parmak izine* sahip olduğudur. Pasif sezim yöntemlerinin karşılaşılabileceği temel zorluk suret videoların tamamıyla aynı ol(a)mayacağıdır. Suret videolar üzerinde parlaklık ya da karışıklık artırma, sıkıştırma, gürültü, sonradan eklenen yazı veya geometrik dönüşümler gözlenebilir bu da birbirine daha da az benzeyen görüntülerin karşılaştırılmasını doğurmaktadır.

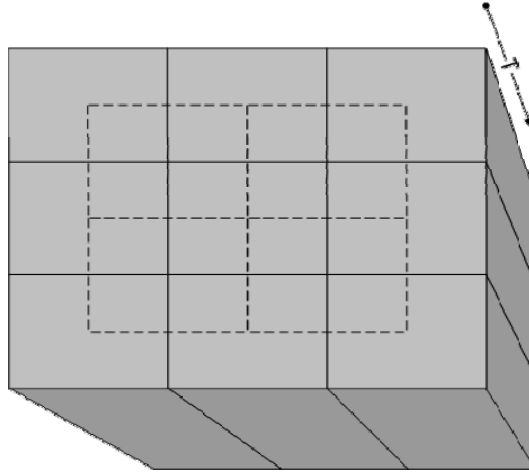
Amerikan Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından her sene düzenlenen TRECVID organizasyonu kapsamına alınan İçerik-tabanlı Kopya Sezimi (Content-Based Copy Detection) (CBCD) işinde yukarıda bahsedilen problemin çözümü için önerdiğimiz sistem ile katılmıştır. TRECVID 2008’de kopya/suret ses ya da video verisi, referans veriden çeşitli dönüşümler/saldırıları ile türetilmiş veri olarak tanımlanmıştır. Video için bu dönüşümler ekleme, çıkarma ve video kaydı gibi birçok saldırıyı içermektedir. Ses verisi için ise dönüşümler, bant kısıtlaması kodlama bozulumu ve çeşitli harmanlamalar olarak belirlenmiştir. TRECVID 2008 – CBCD’nin amacı da kopyaların, eğer gerçekten varsa, referans verilerdeki yerlerinin bulunması olarak tanımlanmıştır. Referans veri 200 saatlik video ve senkronize ses verisinden oluşmaktadır. Ana başlık olarak yalnızca görüntü verisi kullanılarak katılımcılara sunulan yalnızca video dönüşümlerine maruz kalmış 2000 sorgu (kopya) verisinin bulunması; alt başlık olarak da ses+video kullanılarak, ses ve video

dönüşümleri ile elde edilmiş 16000 sorgu verisinin bulunması hedeflenmiştir. TRECVID 2008 ve CBCD hakkında daha detaylı bilgi için [5] başvurulabilir. Kullanılan sistemin ana hatları ve TRECVID 2008'de elde edilen sonuçlar aşağıdaki bölümlerde açıklanmıştır.

4.5.1. Video Öznitelik Çıkarma Modülü:

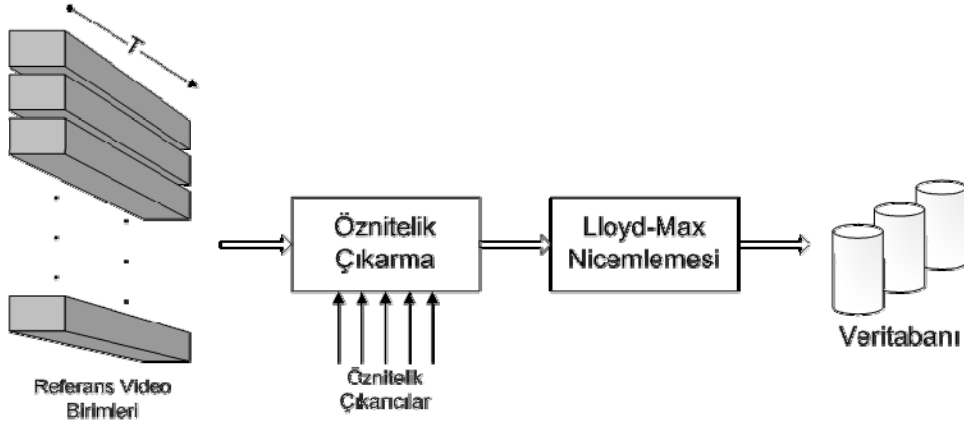
Video endeksleme, büyük miktarlardaki içeriğin taşıdığı bilginin daha etkin gösterimi için düşük boyutlulukta uzaya eşlenmesi olarak görülebilir. Bu bağlamda, anahtar değerleri kısa ve tanımlayıcı olarak tanımlanabilir. Ancak yöntemimizde gen veritabanlarında [2] kullanılan benzer uzun ve kaba anahtar değerleri ile karakterize edilebilecek bir yaklaşımdan faydalanılmıştır. Anahtar değerlerinin, ayırt etme gücü uzunluğundan, gürbüzlüğü ise kaba yapısından kaynaklanmaktadır bununla birlikte hızlı aramalar hiyerarşik ve çoklu-çözünürlük yapılar içeren özel endeksleme yöntemlerinin kullanımı ile gerçekleştirilmektedir.

Kısaca yaklaşımımız, sorgu videosu ile referans videoları arasında bir öznitelik eşleme algoritmasıdır. Öznitelikler ise video içerisindeki uzamsal ve zamansal birimlerden çıkarılmaktadır. Belirtilen bu birimler, video üzerinde oluşturulan bir örnek ve birimler arası kesişime izin veren ızgara yapısı ile elde edilmektedir. Ayrıca, çoklu-çözünürlük özelliği de düşük boyutlara indirgenmiş kareler ile sağlanmaktadır. Buna ek olarak da, zamansallık her ızgara alanının belirli bir periyotta video zamanı içerisinde genişletilmesi ile gerçekleştirilmektedir.



Şekil 92: Öznitelik vektörlerinin çıkarıldığı, örnek ikinci seviye video elemanları. Bu seviyede Toplam 13 eleman bulunmaktadır.

Elde edilen her bir *prizma* üzerinde öznitelik vektörü video içeriğini tanımlayan renk, doku ve hareket uzaylarını kapsayacak şekilde çeşitli yöntemler kullanılarak çıkarılmaktadır. Sistemde, MPEG-7 görsel tanımlayıcılarından [3] elde edilen öznitelik çıkarma yöntemleri kullanılmıştır. Tanımlayıcılar üzerinde gerçekleştirilen değişiklikler temel olarak hesaplama karmaşıklığını azaltmak ve kaba tanımlama yeteneğini kazandırmak için yapılmıştır. Dikkat edilmesi gereken bir nokta da tek bir prizmadan çıkarılan çok çeşitli vektörlerin sıralanarak tek bir *uzun ve kaba* öznitelik vektörü elde edilmiştir. Kaba gösterim, öznitelik vektörlerinin nicemlemesi ile daha da vurgulanmıştır. Referans videolarının öznitelik vektörleri ile oluşturulan veritabanı üzerinde sorgular yapılarak eşlenen bölütler bulunmaktadır. Eşleme, sorgu ve referans öznitelik vektörlerinin iç çarpımlarının sorgu boyunca alınan ortalamasının en büyük olduğu yerde yapılmaktadır.



Şekil 93: Referans videolardan veritabanı oluşturulması.

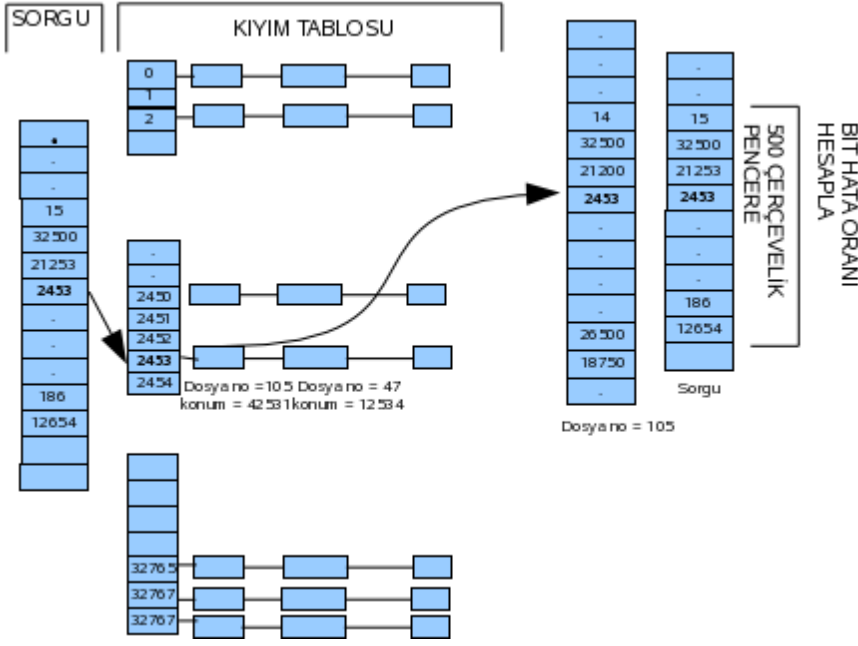
4.5.2. Ses Öznitelik Çıkarma Modülü:

İşitsel verilerin kopyalarının sezimi için kullanılan sistemin, sıkıştırma, bant kısıtlaması ve çeşitli veriler ile karıştırılma gibi birçok атаğa karşı dayanıklı olması gerekmektedir. Kullanılan ses analiz metodu Haitsma [4] tarafından tanımlanan kıyım yöntemini temel almaktadır. Fakat kullandığımız yöntem ses bilgisinin nitelenmesi için 32 bit yerine 15 bitlik kıyım kodları kullanılmaktadır. Her ses çerçevesine dönüşüm uygulanmadan önce Hamming çerçevelemesi uygulanmıştır. Spektrum ise 300Hz – 3000Hz arasında Bark frekanslarına göre 16 kesilmeyen banda bölünmüştür. Bu 16 banttaki enerji değerlerine göre çerçevenin gösterimi için 15 bitlik kıyım değeri hesaplanmaktadır. Ve de, ses kıyımının üretilmesi için aşağıdaki fonksiyon kullanılmaktadır;

$$H(n,m) = 1, \quad EB(n,m) - EB(n,m+1) > 0$$

$$H(n,m) = 0, \quad EB(n,m) - EB(n,m+1) \leq 0$$

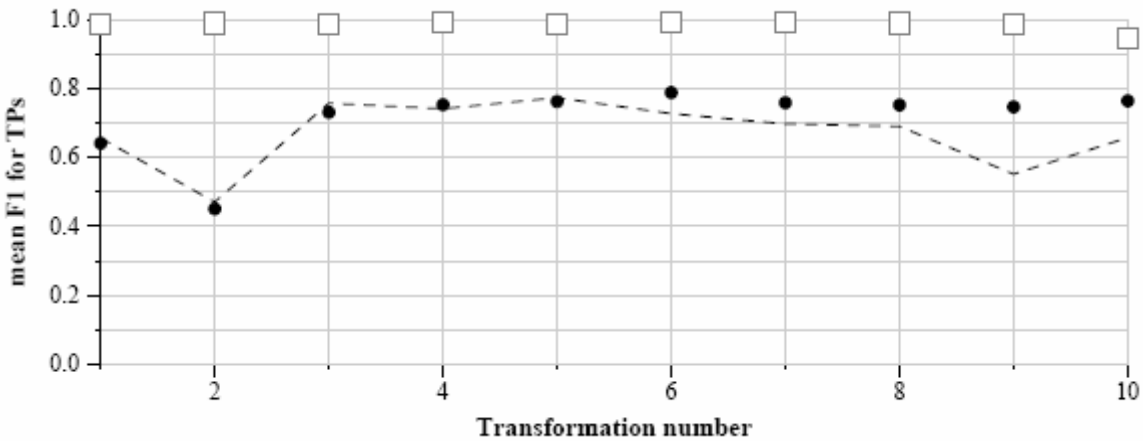
burada $EB(n,m)$ n ve m bantlarındaki enerjiyi tanımlamaktadır. Veritabanı üzerindeki arama işlemi bu kıyım değerleri üzerinden gerçekleştirilerek daha verimli hale aramanın daha verimli hale gelmesi sağlanmıştır. İlk olarak, sorgu verisindeki her bir kıyım değeri için eşleyen veritabanındaki kıyım noktaları bulunmaktadır. Her eşleyen nokta için işe 500 çerçvelik alandaki sorgu ve referans verisi arasındaki bit hata oranı (BER) hesaplanmaktadır. BER'in %30'dan az olduğu çerçeve doğru eşleşme alanı olarak sunulmaktadır. Son-işleme safhasında işe sorgunun referans verideki başlangıç ve bitiş alanları belirlenmektedir. 32 bitlik kıyım yerine 15 bitlik kıyımın kullanılması ile verinin maruz kaldığı ciddi ataklara karşı sistemin daha gürbüz olması sağlanmıştır. Ayrıca 15 bitlik gösterim ile daha düşük bellek ihtiyacı gerekmiştir.



Şekil 94: Ses modülü sorgu süreci.

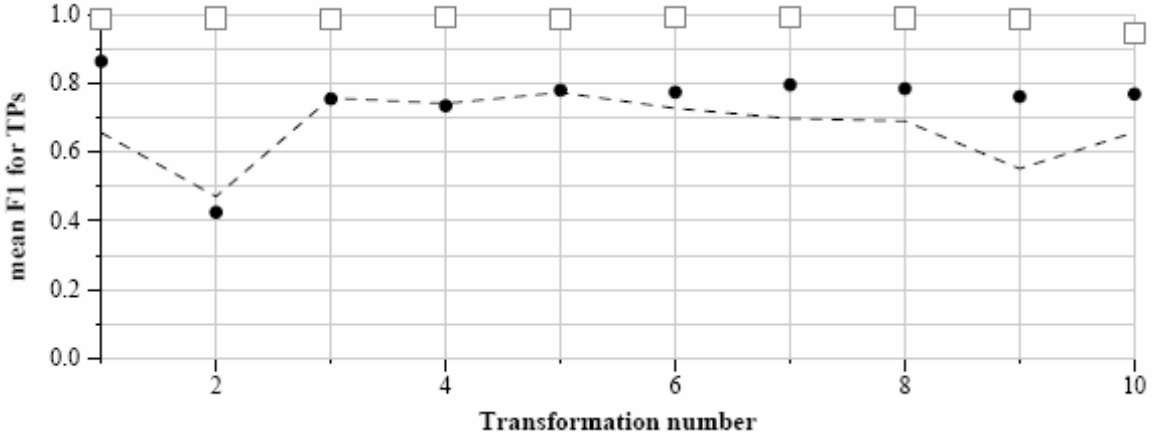
4.5.3. Sonuçlar:

Yalnızca video dönüşümlerinin kullanıldığı alt-başlıkta, yukarıda bahsedilen sistem kullanılmıştır. Sistemin içerisinde öznitelik vektörü oluşturulurken, izleyen nitelikler çıkarılmıştır; her seviyedeki bütün renk kanallarındaki baskın renk değeri, ilk iki seviyedeki yapısal baskın renk videonun renk bilgisini taşımak için değerlendirilmiştir. Dokusal özellikleri için ise, 3D DCT katsayılarından elde edilen 3 temel değer ve kenar enerjisi kullanılmıştır. Hareket bilgisi ise, videonun hareket etkinliği ile gösterilmiştir. NIST tarafından değerlendirilen dönüşüm çeşitlerine göre ortalama F1 performansı Şekil 95’de sunulmuştur. İkinci dönüşüm olarak nitelendirilen “Resim içinde Resim” (PIP) atağı dışında sistem, diğer grupların performansından elde edilen medyan eğrisinin üstünde bir performans göstermiştir.



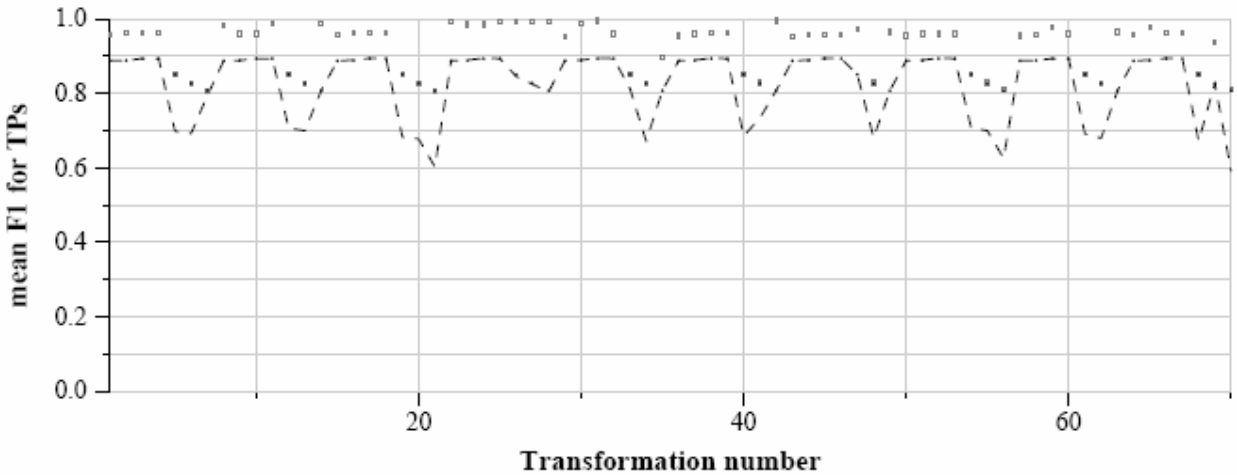
Şekil 95: COST292.v.metuq1 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile önerilen sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.

İkinci çalışmada bir önceki çalışmadakinden farklı olarak videonun renk özellikleri renk topakları olarak gösterilmiştir. Her prizmanın içerisindeki renk histogram ile belirli sayıda topak oluşturularak bunların yüzdeleri değerlendirilmiştir. Bu değişiklikle önceki sonuçtan biraz daha iyi performans elde edilmiştir. En önemli artış birinci dönüşümde (Şekil 96) olmuştur.



Şekil 96: COST292.v.metuq1 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.

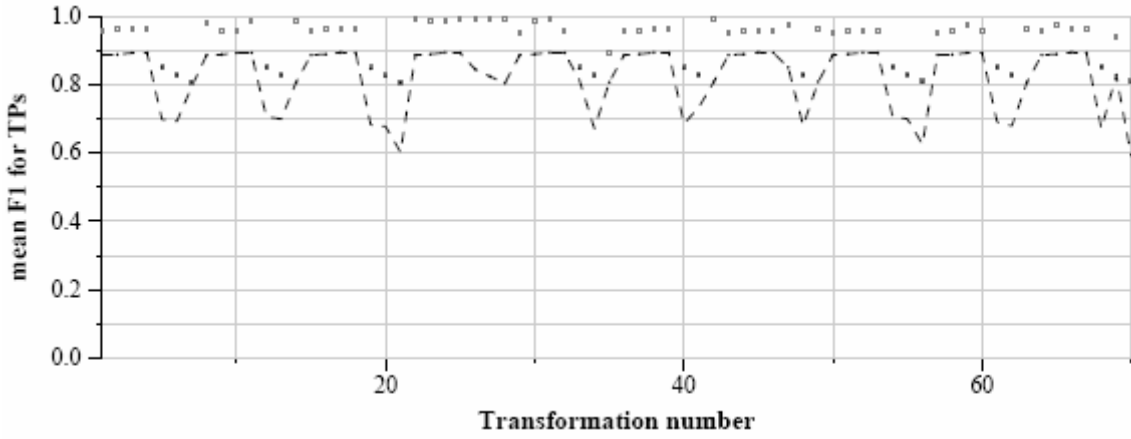
Video ve ses bilgisinin birlikte kullanıldığı ve her iki alan üzerinde yapılan dönüşümler ile elde edilen sorguların bulunmasında daha önce açıklanan modüller birlikte kullanılmaya çalışılmıştır. İlk çalışmada, çalışmanın denektaşı olabilmesi için sorgular yalnızca ses verisi kullanılarak yapılmıştır. Elde edilen sonuçlar Şekil 97'de gösterilmiştir. F1 skorları genel olarak 0.8-0.9 değerleri arasında yer almıştır ve ayrıca konuşma karıştırma atağına maruz kalmış sorguların sonuçlarının diğer dönüşümlerde elde edilen performansdan daha kötü olduğu görülmüştür.



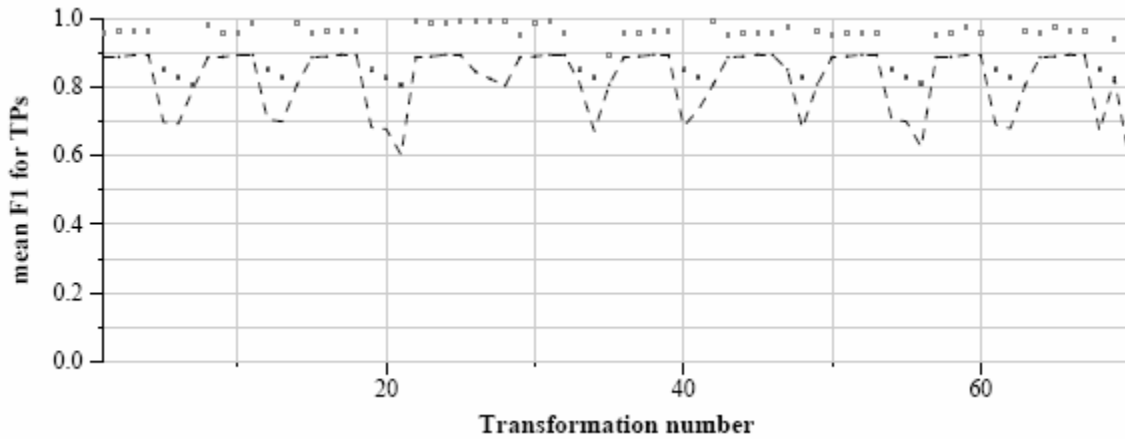
Şekil 97: COST292.m.A0metuq1 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.

Diğer çalışmalarda ise her bir alandan (video/ses) ayrı ayrı elde edilen sorgu sonuçları tümleştirilmiştir. Şekil 98'de gösterilen sonuçlar ses modülüne öncelik verilerek elde edilmiştir. Bundaki en önemli sebep ses modülünün video modülüne göre daha düşük yanlış sonuç dönmesi ve daha az olasılık ile sorguları kaçırmasıdır. Şekil 99'deki sonuçlar ise iki

modüle eşit önem verilerek elde edilmiştir, burada iki modülden de elde edilen güvenilirlik değerleri ortalama alınarak değerlendirilmiştir.



Şekil 98: COST292.m.A0metuq2 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.



Şekil 99: COST292.m.A0metuq3 ait Ortalama F1 ve Dönüşümler grafiği. Nokta ile sistemin performansı, noktalı çizgi ile de katılan bütün grupların sonuçlarının medyanı gösterilmektedir. Kareler ise en iyi performansı belirtmektedir.

4.5.3. Referanslar:

- [1] YouTube - Broadcast Yourself, http://www.youtube.com/results?search_query=*
- [2] Tamer Kahveci and Ambuj K. Singh. An efficient index structure for string databases. In VLDB, pages 351–360. Morgan Kaufmann, 2001.
- [3] B. Manjunath, P. Salembier, and T. Sikora. Introduction to MPEG 7: Multimedia Content Description Language. Ed. Wiley, 2002.
- [4] Jaap Haitsma and Ton Kalker. Robust audio hashing for content identification. In In Content-Based Multimedia Indexing (CBMI), 2001.
- [5] <http://www-nlpir.nist.gov/projects/trecvid/>

5. SONUÇ VE ÖNERİLER

TÜBİTAK destekli COST 292 projesinin amaçlarına tümüyle ulaşmış ve proje kapsamında yer alan bütün konularda hedeflenen bilimsel çalışmalar gerçekleştirilmiş durumdadır. Projeye başlanırken ilgili kurum mevzuatındaki o dönemde yer alan kısıtlar nedeniyle, bu proje kapsamında tam veya yarı zamanlı araştırmacıların çalıştırılmasının mümkün olmaması, projedeki bilimsel çalışmaları daha ileriye götürememe konusunda belirleyici olmuştur. Bu duruma rağmen, proje kapsamında yer alan farklı konu ve başlıklarda, özgün veya literatürdeki yeni çalışmalarla ilişkili algoritmalar ortaya konmuş ve deneyler yardımıyla başarımları değerlendirilmiştir. Ayrıca COST 292 grubu içinde, projenin teknik planının yazılmasında öncü rol oynanmasından başlanarak, teknik iş paketi liderliği yanında, CMBI 2008 çalıştayında COST 292 adına özel oturum düzenlenmesiyle ve COST 292 olarak beraberce gerçekleştirilen tüm uluslararası aktivitelere (TRECVID 2006, TRECVID 2007, TRECVID 2008) ODTÜ adına katılım sağlanarak göreceli olarak aktif bir rol oynanmıştır.

Üst düzey anlamsal bilgiye ulaşma konusundaki iddialı hedefe ulaşmak için literatürde çeşitli yollar önerilmektedir. Genel olarak, bu problemin yerel yaklaşımlar ile çözülebileceği, literatürdeki metodlara bağlı değerlendirmeler ve yapılan deneyler sonucu söylenebilir. Yerel çözüm başlığı altında, genel resmin homojen koyuluk (veya hareket veya 3B nokta yakınlığı) bölgelerine bölünmesi ve sonrasında bölütlenmiş nesnelere tanımlanması yanında, eşleştirmeye uygun yerel ilgi noktaları kullanarak nesnelere eşleştirilmesi ve anlamsal analizlerinin yapılması düşünülebilir. Proje kapsamında bölütleme problemi tüm yönleriyle incelenmiş, uzamsal (imge), zamansal (kamera çekim sınırları) ve uzamsal-zamansal (hareketli nesne) bölütleme problemleri için özgün çözümler ortaya konmuştur. Bunlar dışında, videoda gözlenen sahneye ait seyrek 3B derinlik değerlerinin de kendi içlerinde bölütlenmesi problemi de incelenmiştir. İlgi noktaları ile ilgili araştırmalar ise ön safhasında raporlanmış ve bu konuda literatürde önde gelen yöntemler sunulmuştur.

Bölütleme probleminin çeşitli durumlar için başarılması halinde, ortaya çıkacak nesne silüetlerinin üst düzey anlamsal bilgiler taşıması durumuna bağlı olarak proje kapsamında 2-B ve 3-B şekil tanımlayıcıları konusunda çalışmalar yapılarak, literatürde önde gelen yaklaşımlar deneyler yardımıyla karşılaştırılmıştır. Şekil kökenli nesne tanıma araştırmalarında 3B modellerin 2B izdüşümlerini kullanıyor ve bu izdüşümleri çeşitli ataklar yardımıyla bozarak gerçek uygulamaları simüle etmeye çalışarak gerçekçi sonuçlara karşılık gelmemektedir. Bu rağmen bölütleme sonuçlarının yaratabileceği hataların şekil kökenli tanıma algoritmaları için performansı belirleyici olacağı düşüldüğünde deneyler sonucu bölge tabanlı yaklaşımların, çeper tabanlı yaklaşımlara göre daha gülbüz olduğu ve bölütleme hatalarından daha az etkileneyeceği değerlendirilmiştir. Performansı değerlendirilmesi için gerçek bölütleme sonuçlarından oluşan bir veri seti oluşturulup, bu set ile deneyler yapılması önerilmektedir. 3-B şekil tanımlayıcıları için Fourier tabanlı yaklaşımların kabul edilebilir bir performansı vermesine rağmen, göreceli olarak daha güncel küresel harmonik kökenli yaklaşımların bu performansın üzerine çıkabileceği değerlendirilmiştir. Kamera hareketine yönelik çalışmalar ise literatürdeki önde gelen yaklaşımları dikkate alarak gerçekleştirilmiş ve gerek kamera hareket tespiti, gereken bu hareketin bulunması konularında yapılan çalışmalar doyurucu sonuçlar vermiştir.

Proje kapsamında gerçekleştirilen uygulamalar, proje önerisinde belirlenen konular dışında, proje süresince COST 292 grubunda beraberce hareket edilmesi düşünülerek seçilen alanları da dikkate alınarak revize edilmiştir. Videoyazı konusunda baştan sona bir sistem tüm alt bölümleriyle, literatürde yer alan çalışmalar dikkate alınarak ederek, tasarlanmış, çeşitli yenilikçi yaklaşımlarla zenginleştirilmiştir. Önerilen sistemin temel çıkması, video

özünürlüklerinin, sayısal olarak taranmış tipik doküman özünürlüklerine göre çok düşük kalması ve karakter tanıma algoritması için yetersiz olmasıdır. Keza, aynı düşük özünürlük problemi, karakterlerin birbirlerinden ayrıştırılması adımında da önemli sorunlar çıkarmakta, yapışık karakterlerin oluşmasına neden olmaktadır. Bu temel problemi aşmaya yönelik öneri, sistemde kullanılmayan zaman eksenindeki artık bilgiyi alt düzey piksel seviyesinde özünürlük arttırmaya yönelik kullanmak (süperözünürlük kökenli yaklaşım) veya video karelerinde arka arkaya yapılacak tanımların kararlarını üst düzeyde biraraya getirmek olarak değerlendirilmektedir.

Görsel güvenlik sistemlerinde birden çok kamerayı birarada kullanarak ortamdaki anlamsal analizleri gerçekleştirmenin, tek kameralı sistemlere göre daha başarılı olacağı aşıkardır. Proje kapsamında yapılan alışmalarda çok sayıda kameranın tek bir zaman anı için örtüşmelere olan etkisi incelenmiş ve iyileştirmeler gözlenmiştir. Bu yöntemi daha ileri götürmek için önerilebilecek bir yaklaşım üst açıdan yaratılan nesnelere ait çizgenin zaman eksenindeki deęişikliklerini de dikkate almaktır. Böyle bir yaklaşımın çok daha başarılı bir örtüşme özümü olacağı değerlendirilmektedir.

Video kopyası bulma problemi göreceli olarak yeni bir problem olarak durmaktadır. Önerilen yaklaşım uzamsal-zamansal parçalardan elde edilen standartlaştırılmış renk ve doku tanımlayıcılarının önemli oranda nicemlendirilmesinden sonra düşük işlem karmaşıklığında karşılaştırılmasına dayanmaktadır. Ancak literatürdeki ve TRECVID sonucu ortaya çıkan başarılı algoritmalarındaki temel yaklaşım anahtar karelerden ilgi noktası bulmanın daha başarılı olduğu yönündedir. Video kopyası bulma probleminde bu yaklaşımın daha tercih edilmesi önerilmektedir.

**TÜBİTAK
PROJE ÖZET BİLGİ FORMU**

Proje No: 104E019
Proje Başlığı: Sayısal Çoğulortam Verisinin Anlamsal Çokkipli Analizi
Proje Yürütücüsü ve Araştırmacılar: Doç.Dr. A. Aydın Alatan (TÜBİTAK destekli başka araştırmacı bulunmamaktadır)
Projenin Yürütüldüğü Kuruluş ve Adresi: O.D.T.Ü. Elektrik ve Elektronik Mühendliği Bölümü, Balgat 06531, Ankara
Destekleyen Kuruluş(ların) Adı ve Adresi: TÜBİTAK EEEAG, Kavaklıdere 06100 Ankara
Projenin Başlangıç ve Bitiş Tarihleri: 1 Ekim 2004 – 30 Eylül 2008
Öz (en çok 70 kelime) COST 292 projesini temel amacı çokluortam verilerinden otomatik olarak anlamsal bilgi çıkarmaya yöneliktir. Proje kapsamında yapılan çalışmalar, öznelik çıkarma, bölütleme ve uygulamalar olarak 3 ana başlık altında incelenebilir. Öznelik başlığı altında, 2-B ve 3-B şekil tanımlayıcılar, kamera hareketine yönelik zamansal öznelikler ve ilgi noktası çıkarma konuları incelenmiştir. Bölütleme başlığı altında, çizge kuramı temelli imge bölütleme, parametric bölgesel hareket modelleri kullanarak nesne bölütleme ve kamera çekim sınırlarını bulma amaçlı çizge kuramsal domine set tabanlı özgün yaklaşımlar önerilmiştir. Uygulamalar başlığında ise, 2-B silüetlerde taşıt tanıma, videoyazı bulma ve tanıma, güvenlik sistemlerinde çoklu kamera ile bölütleme, TV yayınlarından logo bulma ve çoğulortam veri tabanlarından video kopyası bulma konularında çalışmalar yapılmıştır.
Anahtar Kelimeler: İçerik tabanlı analiz, çoğulortam indekleme, şekil tanımlayıcı, kamera hareketi, imge bölütleme, nesne bölütleme, kamera çekimi bölütleme, videoyazı, logo bulma, çoklu kamera güvenlik sistemleri, video kopyası bulma
Projeden Yapılan Yayınlar: <ul style="list-style-type: none">• Cevahir Çigla, A. Aydın Alatan, “Dense Depth Map Estimation for Object Segmentation in Multi-view Video”, IEEE Transactions on Multimedia dergisine sunulmuştur• Ersin Esen, Medeni Soysal, Tugrul K. Ates, Ahmet Saracoglu and A. Aydın

Alatan,"A Fast Method For Animated TV Logo Detection", Sixth International Workshop on Content-Based Multimedia Indexing, 18-20th June, 2008, London, UK

- Oytun Akman, A. Aydın Alatan and Tolga Çiloglu, "Exploitation of Multi-camera Configuration for Visual Surveillance", Sixth International Workshop on Content-Based Multimedia Indexing, 18-20th June, 2008, London, UK
- Cevahir Çigla and A. Aydın Alatan, "Region-based Image Segmentation via Graph Cuts", ICIP 2008, San Diego CA
- Oytun Akman, A. Aydın Alatan and Tolga Çiloglu. "Multi-Camera Visual Surveillance for Motion Detection, Occlusion Handling, Tracking and Event Recognition", European Conference on Computer Vision (ECCV) Workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications, October 2008, Marseille, France
- Cevahir Çigla and A. Aydın Alatan, "Bölge Tabanlı Çizge Kesik Görüntü Bölütlemesi", IEEE 16. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı, 20-22 Nisan 2008, Didim (*Alper Atalay* en iyi 2. bildiri ödülü)
- Tugrul K. Ates, Ersin Esen, Ahmet Saracoğlu, A. Aydın Alatan, "Sinir Eslemeye Dayalı Yarı-Saydam TV Logosu Tespiti" IEEE 16. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı, 20-22 Nisan 2008, Didim
- Q. Zhang, M. Corvaglia, S. Aksoy, U. Naci, N. Adami, N. Aginako, A. Alatan, et.al., "The COST292 experimental framework for TRECVID 2007", TRECVID Workshop November 5-6 2007, Gaithersburg, MD
- Birant Örtten, A. Aydın Alatan, Tolga Çiloglu, Recognizing Events in an Automated Surveillance System. "Int. Workshop on MRCS 2006 (LNCS 4105)", (2006)
- Ahmet Saracoğlu, A. Aydın Alatan, MRF-based Videotext Detection. "Proc. of 13th Int. Conf. on SSIP", (2006)
- B. Birant Örtten, A. Aydın Alatan, Tolga Çiloglu, Otomatize Güvenlik Sistemlerinde Olay Analizi. "SIU 2006", (2006)
- Ahmet Saraçoğlu, A. Aydın Alatan, Görüntü İçindeki Yazıların Otomatik Tespiti. "Sinyal İşleme ve Uygulamaları, SIU 2006", (2006)
- Ahmet Saracoğlu, A. Aydın Alatan, et.al., COST292 experiments for TRECVID 2006. "TRECVID 2006 Workshop", (2006)
- B. Birant Örtten, Medeni Soysal and A. Aydın Alatan, Person Identification in Surveillance Video by Combining MPEG-7 Experts. "WIAMIS 2005", (2005)
- B. Birant Örtten, Medeni Soysal and A. Aydın Alatan, Güvenlik Amaçlı Videolarda MPEG-7 Uzmanlarını Birleştirerek İnsan Tanıma. "SIU 2005", (2005)

**TÜBİTAK
PROJE ÖZET BİLGİ FORMU**

Proje No: 104E019
Proje Başlığı: Sayısal Çoğulortam Verisinin Anlamsal Çokkipli Analizi
Proje Yürütücüsü ve Araştırmacılar: Doç.Dr. A. Aydın Alatan (TÜBİTAK destekli başka araştırmacı bulunmamaktadır)
Projenin Yürütüldüğü Kuruluş ve Adresi: O.D.T.Ü. Elektrik ve Elektronik Mühendliği Bölümü, Balgat 06531, Ankara
Destekleyen Kuruluş(ların) Adı ve Adresi: TÜBİTAK EEEAG, Kavaklıdere 06100 Ankara
Projenin Başlangıç ve Bitiş Tarihleri: 1 Ekim 2004 – 30 Eylül 2008
Öz (en çok 70 kelime) COST 292 projesini temel amacı çokluortam verilerinden otomatik olarak anlamsal bilgi çıkarmaya yöneliktir. Proje kapsamında yapılan çalışmalar, öznitelik çıkarma, bölütleme ve uygulamalar olarak 3 ana başlık altında incelenebilir. Öznitelik başlığı altında, 2-B ve 3-B şekil tanımlayıcılar, kamera hareketine yönelik zamansal öznitelikler ve ilgi noktası çıkarma konuları incelenmiştir. Bölütleme başlığı altında, çizge kuramı temelli imge bölütleme, parametric bölgesel hareket modelleri kullanarak nesne bölütleme ve kamera çekim sınırlarını bulma amaçlı çizge kuramsal domine set tabanlı özgün yaklaşımlar önerilmiştir. Uygulamalar başlığında ise, 2-B silüetlerde taşıt tanıma, videoyazı bulma ve tanıma, güvenlik sistemlerinde çoklu kamera ile bölütleme, TV yayınlarından logo bulma ve çoğulortam veri tabanlarından video kopyası bulma konularında çalışmalar yapılmıştır.
Anahtar Kelimeler: İçerik tabanlı analiz, çoğulortam indekleme, şekil tanımlayıcı, kamera hareketi, imge bölütleme, nesne bölütleme, kamera çekimi bölütleme, videoyazı, logo bulma, çoklu kamera güvenlik sistemleri, video kopyası bulma
Projeden Yapılan Yayınlar: <ul style="list-style-type: none">• Cevahir Çigla, A. Aydın Alatan, "Dense Depth Map Estimation for Object Segmentation in Multi-view Video", IEEE Transactions on Multimedia dergisine sunulmuştur• Ersin Esen, Medeni Soysal, Tugrul K. Ates, Ahmet Saracoglu and A. Aydın Alatan, "A Fast Method For Animated TV Logo Detection", Sixth International Workshop on Content-Based Multimedia Indexing, 18-20th June, 2008, London, UK• Oytun Akman, A. Aydın Alatan and Tolga Çiloglu, "Exploitation of Multi-camera

Configuration for Visual Surveillance", Sixth International Workshop on Content-Based Multimedia Indexing, 18-20th June, 2008, London, UK

- Cevahir Çigla and A. Aydın Alatan, "Region-based Image Segmentation via Graph Cuts", ICIP 2008, San Diego CA
- Oytun Akman, A. Aydın Alatan and Tolga Çiloğlu. "Multi-Camera Visual Surveillance for Motion Detection, Occlusion Handling, Tracking and Event Recognition", European Conference on Computer Vision (ECCV) Workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications, October 2008, Marseille, France
- Cevahir Çigla and A. Aydın Alatan, "Bölge Tabanlı Çizge Kesik Görüntü Bölütlemesi", IEEE 16. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı, 20-22 Nisan 2008, Didim (*Alper Atalay* en iyi 2. bildiri ödülü)
- Tugrul K. Ates, Ersin Esen, Ahmet Saracoğlu, A. Aydın Alatan, "Sinir Eslemeye Dayalı Yarı-Saydam TV Logosu Tespiti" IEEE 16. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı, 20-22 Nisan 2008, Didim
- Q. Zhang, M. Corvaglia, S. Aksoy, U. Naci, N. Adami, N. Aginako, A. Alatan, et.al., "The COST292 experimental framework for TRECVID 2007", TRECVID Workshop November 5-6 2007, Gaithersburg, MD
- Birant Örtten, A. Aydın Alatan, Tolga Çiloğlu, Recognizing Events in an Automated Surveillance System. "Int. Workshop on MRCS 2006 (LNCS 4105)", (2006)
- Ahmet Saracoğlu, A. Aydın Alatan, MRF-based Videotext Detection. "Proc. of 13th Int. Conf. on SSIP", (2006)
- B. Birant Örtten, A. Aydın Alatan, Tolga Çiloğlu, Otomatize Güvenlik Sistemlerinde Olay Analizi. "SIU 2006", (2006)
- Ahmet Saracoğlu, A. Aydın Alatan, Görüntü İçindeki Yazıların Otomatik Tespiti. "Sinyal İşleme ve Uygulamaları, SIU 2006", (2006)
- Ahmet Saracoğlu, A. Aydın Alatan, et.al., COST292 experiments for TRECVID 2006. "TRECVID 2006 Workshop", (2006)
- B. Birant Örtten, Medeni Soysal and A. Aydın Alatan, Person Identification in Surveillance Video by Combining MPEG-7 Experts. "WIAMIS 2005", (2005)
- B. Birant Örtten, Medeni Soysal and A. Aydın Alatan, Güvenlik Amaçlı Videolarda MPEG-7 Uzmanlarını Birleştirerek İnsan Tanıma. "SIU 2005", (2005)