**BAŞKENT ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

# SAAT KADRANININ OPTİK GÖRÜNTÜSÜNDEN SAATİN KAÇ OLDUĞUNU ANLAYAN GÖRÜNTÜ İŞLEM SİSTEMİ

# AN IMAGE PROCESSING SYSTEM FOR READING TIME FROM THE OPTICAL IMAGE OF HOUR PLATE

**İ. UMUT KOCATÜRK**

**YÜKSEK LİSANS TEZİ**

**ANKARA**

**ARALIK, 2007**

# SAAT KADRANININ OPTİK GÖRÜNTÜSÜNDEN SAATİN KAÇ OLDUĞUNU ANLAYAN GÖRÜNTÜ İŞLEM SİSTEMİ

# AN IMAGE PROCESSING SYSTEM FOR READING TIME FROM THE OPTICAL IMAGE OF HOUR PLATE

**İ. UMUT KOCATÜRK**

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

**ARALIK, 2007**

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma, jürimiz tarafından **ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan (Danışman) ….....

Prof. Dr. Emin AKATA

Üye ….....

Yrd. Doç. Dr. İsmail Avcıbaş

Üye ….....

Yrd. Doç. Dr. Hasan Oğul

ONAY

Bu tez ...../...../2007 tarihinde Enstitü Yönetim Kurulunca belirlenen yukarıdaki jüri üyeleri tarafından kabul edilmiştir.

...../...../2007

Prof. Dr. Emin AKATA

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

**TEŞEKKÜR**

Saygıdeğer hocam ve tez danışmanın Sayın Prof. Dr. Emin AKATA'a, değerli görüşleri ve bilgileriyle çalışmanın sonuca ulaştırılmasında ve karşılaşılan güçlüklerin aşılmasında her zaman yardımcı ve yol gösterici olduğu için,
Sayın Yrd. Doç. Dr. İsmail Avcıbaş'a, tez çalışmasında değerli görüş ve düşüncelerini belirttiği ve hataların düzeltilmesinde yardımcı olduğu için,
Sayın Yrd. Doç. Dr. Hasan Oğul'a, tez çalışmamı özenle ve titizlikle değerlendirdiği ve yaptığı katkılar için,
en içten teşekkürlerimi sunarım.

## ÖZ


**SAAT KADRANININ OPTİK GÖRÜNTÜSÜNDEN SAATİN KAÇ OLDUĞUNU ANLAYAN GÖRÜNTÜ İŞLEM SİSTEMİ**

İ. Umut Kocatürk

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Bu çalışmada, analog bir göstergesi (kadran) olan saatin, optik görüntüsünden saatin kaç olduğunu anlayan görüntü işlem sistemi olan saat okuma sistemi gerçekleştirilmiştir. Saat okuma sistemi ile akrep, yelkovan ve saniye kolu olan bir saatin görüntüsü, karşısına yerleştirilen bir video kamera aracılığı ile algılanır. Görüntü bilgisayar ortamında gerekli görüntü işlem teknikleri ile işlenerek çözümlenir ve saatin kaç olduğu anlaşılır. Gerçekleştirilen sistemin çözümleme sürecinin hızı, gerçek zamanlı çalışmaya uygundur.

Sistem, analog göstergesi olan saati otomatik olarak okumaktadır. Okuma işlemi, analog göstergenin görüntüsü içindeki işaretçilerin konumları algılanarak, analog göstergenin gösterdiği değerin bulunmasıdır. Saat görüntüsü içindeki işaretçiler olan akrep, yelkovan ve saniye kollarının algılanmasında, muhtelif görüntü işleme teknikleri kullanılmıştır. Saniye kolu harekete dayalı bölütleme ile görüntüden bölütlenip tanınmaktadır. Saniye kolunun inceltilmiş halinden çıkarılan öznitelikler kullanılarak saatin görüntü içindeki konumu bulunmaktadır. Akrep ve saniye kolları tohumlu bölge büyütme yöntemi ile görüntüden bölütlenmektedir. Akrep ve yelkovan kollarının kendisinden ve inceltilmiş hallerinden çıkarılan öznitelikler kullanılarak, iki kol birbirinden ayırt edilmekte ve tanınmaktadır. Kolların inceltilmiş hallerinden çıkarılan öznitelikler kullanılarak saatin kaç olduğu anlaşılmaktadır.



**Anahtar Sözcükler:** Görüntü İşleme, Nesne Tanıma, Görüntü Bölütleme, İnceltme, Tohumlu Bölge Büyütme

**Danışman:** Prof. Dr. Emin AKATA, Başkent Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü.

## ABSTRACT

### AN IMAGE PROCESSING SYSTEM FOR READING TIME FROM THE OPTICAL IMAGE OF HOUR PLATE

İ. Umut Kocatürk

Başkent University Institute of Science

The Department of Electrical and Electronics Engineering

In this study, a time reading image processing system, conceiving time from the optical image of a watch with an analog indicator (hour plate) was implemented. By time reading system, the image of a watch with hour hand, minute hand and second hand is captured with a video camera that is positioned against the watch. The time is read by analyzing the image with the selected image processing techniques. The speed of analysis is acceptable for real time operation.

The system, automatically reads the time from a watch with analog indicator. That process basically consists of finding the value pointed by analog indicator by sensing the positions of pointers which are inside the image of the analog watch. Various image processing techniques are used to recognize hour hand, minute hand and second hand that are the parts of watch indicator. Second hand is segmented and recognized from the image by motion based segmentation. The position of watch in the image frame is found by using features extracted from the skeleton of second hand. Hour hand and second hand are segmented from the image by seeded region growing technique. By using features extracted from both hour hand and minute hand and their skeletons, two hands are differentiated and recognized. Finally by using features extracted from the skeleton of hands, the time is conceived.

**Keywords:** Image Processing, Object Recognition, Image Segmentation, Thining, Seeded Region Growing

**Advisor:** Prof. Dr. Emin AKATA, Başkent University, The Department of Electrical and Electronics Engineering.

## İÇİNDEKİLER LİSTESİ

Sayfa

TEŞEKKÜR ..... .

ÖZ ..... i

ABSTRACT ..... ii

İÇİNDEKİLER LİSTESİ ..... iii

ŞEKİLLER LİSTESİ ..... v

ÇİZELGELER LİSTESİ ..... ix

SİMGELER VE KISALTMALAR LİSTESİ ..... x

KULLANILAN TERİMLER SÖZLÜĞÜ ..... xi

**1. GİRİŞ ..... 1**

**2. SAYISAL GÖRÜNTÜNÜN TEMELLERİ ..... 5**

2.1. Siyah Beyaz Görüntü Modeli ..... 5

2.2. Sayısal Görüntü ..... 6

2.3. Pikseller Arasındaki İlişkiler ..... 7

2.3.1. Komşuluk ..... 7

2.3.2. Bağlılık ve bitişiklik ..... 8

2.3.3. Yol ve bölge ..... 8

2.3.4. Uzaklık ölçümü ..... 9

2.4. Görüntü Histogramı ..... 10

2.5. Karşıtlık Artırma ..... 12

**3. İKİ BOYUTLU NESNE TANIMA ..... 13**

3.1. Görüntü Bölütleme ..... 14

3.1.1. Eşikleme yöntemi ile bölütleme ..... 16

3.1.1.1. Görüntü yumuşatma ..... 20

3.1.2. Bölgesel tabanlı bölütleme ..... 22

3.1.2.1. Bölge büyütme ..... 23
3.1.3. Harekete dayalı bölütleme ..... 27
3.2. Öznitelik Çıkarma ..... 30
3.2.1. Bölge karakteristiklerine göre çıkarılabilecek öznitelikler ..... 32
3.2.2. Sınır karakteristiklerine göre çıkarılabilecek öznitelikler ..... 34
3.2.2.1. İmza ..... 35
3.2.3. İnceltme ..... 36
3.3. Örüntü Tanıma ..... 41
**4. SAATİN KAÇ OLDUĞUNUN ANLAŞILMASI ..... 44**
4.1. Program Arayüzü ve Genel Algoritması ..... 48
4.2. Yakalanan Görüntüler ve Gri Seviyeye Dönüşümleri ..... 53
4.3. Yakalanan Görüntülerin Doğruluğu ..... 57
4.4. Saatin Konumunun Bulunması ..... 65
4.5. Saatin Konumunun Kontrolü ..... 75
4.6. Saati Okuma ..... 78
4.6.1. Kolların (işaretçilerin) bölütlenmesi ..... 79
4.6.1.1. Saniye kolunun bölütlenmesi ..... 80
4.6.1.2. Akrep ve yelkovan kollarının bölütlenmesi ..... 84
4.6.2. Kol sayısının ve açılarının bulunması: ..... 89
4.6.3. Kolların tanınması ..... 94
4.6.3.1. Kollardan öznitelik çıkarma ..... 94
4.6.3.2. Çıkarılan özniteliklerin karşılaştırılması ..... 101
4.6.4. Saat, dakika ve saniye'nin hesaplanması ..... 107
**5. SONUÇ ..... 109**
KAYNAKLAR LİSTESİ ..... 112
EKLER ..... 115

## ŞEKİLLER LİSTESİ

Sayfa

Şekil 2-1 Piksel komşulukları ..... 7

Şekil 2-2 Pikseller arasındaki yollar ..... 9

Şekil 2-3 Örnek bölgeler ..... 9

Şekil 2-4 (a) Örnek bir görüntü (b) Örnek görüntünün histogramı ..... 11

Şekil 2-5 Karşıtlık artırma dönüşüm fonksiyonu [9] ..... 12

Şekil 3-1 Görüntü bölütleme sistemi ..... 14

Şekil 3-2 İki doruklu histogram ..... 17

Şekil 3-3 Birden fazla nesne içeren görüntünün histogramı ..... 17

Şekil 3-4 Örnek bir histogram ..... 19

Şekil 3-5 3x3 boyutunda süzgeç matrisi ..... 21

Şekil 3-6 (a) Ortalama süzgeci (b) Ağırlıklı ortalama süzgeci ..... 22

Şekil 3-7 Örnek bir histogram ..... 26

Şekil 3-8 (a) $t_i$ anında kamera ile alınan analog saat görüntüsü $f(x,y,t_i)$ ..... 30

(b) $t_j$ anında kamera ile alınan analog saat görüntüsü $f(x,y,t)$.....30

(c) $g_{ij}(x,y) = f(x,y,t_i) - f(x,y,t_j)$ görüntüsünün ölçeklenmiş hali.....30

(d) Şekil 1-8(c)'deki görüntünün eşikleme uygulanmış hali.....30

Şekil 3-9 Öznitelik çıkarma sistemi ..... 31

Şekil 3-10 Nesne bölgesi ..... 33

Şekil 3-11 (a) Düzgün dokulu nesne [9] ..... 33

(b) Kaba dokulu nesne [9].....33

(c) Düzenli dokulu nesne [9].....33

Şekil 3-12 İki delikli bölge ..... 34

Şekil 3-13 Üç bağlı bileşenli bölge ..... 34

Şekil 3-14 Nesne bölgesi ve sınırları ..... 35

Şekil 3-15 (a) Örnek bir nesne sınırı (b) Nesne sınırının imza gösterimi ..... 36

Şekil 3-16 İnceltme algoritması için 8-komşuluktaki piksellerin sıralanışı [9] ..... 37

Şekil 3-17 İnceltme algoritmasındaki (a) ve (b) koşullarının gösterimi..... 38

Şekil 3-18 (a) U harfi görüntüsü (b) U harfinin iskelet görüntüsü ..... 41

Şekil 3-19 Örüntü Tanıma Sistemi..... 42

Şekil 4-1 Saat okuma sistemi ..... 44

Şekil 4-2 (a) Arka planı kollara göre daha açık olan örnek saat görüntüsü ..... 46
(b) Arka planı kollara göre daha koyu olan örnek saat görüntüsü.....46

Şekil 4-3 (a) Aydınlatmanın düşük olduğu saat görüntüsü ..... 47
(b) Aydınlatmanın düzgün olmadığı saat görüntüsü.....47

Şekil 4-4 Aydınlatmanın türdeş ve yeterli olduğu saat görüntüsü ..... 48

Şekil 4-5 Programın arayüzü ..... 49

Şekil 4-6 Programın veri akış diyagramı.....50

Şekil 4-7 Saat konumunun bileşenleri ..... 52

Şekil 4-8 1000 milisaniye farklarla yakalanmış, dört renkli saat görüntüsü..... 54
(a) Birinci görüntü.....54
(b) İkinci görüntü.....54
(c) Üçüncü görüntü.....54
(d) Dördüncü görüntü.....54

Şekil 4-9 RGB renk küpü [9] ..... 55

Şekil 4-10 Yakalanan görüntülerin gri seviyeye dönüştürülmüş halleri ..... 57
(a) Birinci görüntü.....57
(b) İkinci görüntü.....57
(c) Üçüncü görüntü.....57
(d) Dördüncü görüntü.....57

Şekil 4-11 Fark görüntüleri ve histogramları ..... 59
(a) 1. ve 2. görüntülerden elde edilen fark görüntüsü ve histogramı.....59
(b) 2. ve 3. görüntülerden elde edilen fark görüntüsü ve histogramı.....59
(c) 3. ve 4. görüntülerden elde edilen fark görüntüsü ve histogramı.....59

Şekil 4-12 Yumuşatılmış fark görüntüleri ve histogramları..... 61
(a) 1. Fark görüntüsü ve histogramı.....61
(b) 2. Fark görüntüsü ve histogramı.....61
(c) 3. Fark görüntüsü ve histogramı.....61

Şekil 4-13 Yumuşatılmış ve eşiklenmiş fark görüntüleri ..... 63
(a) 1. Fark görüntüsü ve histogramı.....63
(b) 2. Fark görüntüsü ve histogramı.....63
(c) 3. Fark görüntüsü ve histogramı.....63

Şekil 4-14 Yumuşatılmış ve eşiklenmiş fark görüntüleri ..... 66
(a) 1. Fark görüntüsü ve histogramı .....66
(b) 3. Fark görüntüsü ve histogramı .....66

Şekil 4-15 (a) 1. ile 3. saniye kolları ve iskeletleri ..... 68
(b) 2. ile 4. saniye kolları ve iskeletleri .....68

Şekil 4-16 İki doğru parçasının kesişimi [28] ..... 69

Şekil 4-17 (a) 1. ve 3. doğru parçalarının başlangıç, bitiş ve kesişim noktaları .... 72
(b) 2. ve 4. doğru parçalarının başlangıç, bitiş ve kesişim noktaları.....72

Şekil 4-18 (a) 1. durumda yakalan 4 görüntüden sonuncusu ..... 76
(b) 2. durumda yakalan 2 görüntüden birincisi.....76

Şekil 4-19 (a) Şekil 4-18(a)'daki görüntünün üzerine çember oturtulmuş hali..... 77
(b) Fark görüntüsünün üzerine çember oturtulmuş hali.....77

Şekil 4-20 İskeletler üzerindeki en uzak noktalar ve iskeletlerin açıları ..... 83

Şekil 4-21 Şekil 4-10(d)'deki görüntünün karşıtlık artırma uygulanmış hali ..... 85

Şekil 4-22 (a) Görüntü parçası (b) Görüntü parçasının histogramı..... 86

Şekil 4-23 S.R.G. algoritmasının başlangıç görüntüsü ..... 87

Şekil 4-24 (a) S.R.G. algoritmasının uygulandığı görüntü ..... 88
(b) S.R.G. algoritmasının çıkış görüntüsü.....88

Şekil 4-25 Bölütleme işlemi sonucunda elde edilen bölgeler..... 89

Şekil 4-26 Şekil 4-25'daki görüntünün iskeleti ..... 90

Şekil 4-27 Şekil 4-26'daki görüntünün, üzerine çember oturtulmuş hali ..... 91

Şekil 4-28 Açı değerlerindeki nokta sayılarını veren grafik (imza gösterimi)..... 91

Şekil 4-29 Kolların açıları..... 93

Şekil 4-30 En uzak iskelet noktalarının aranacağı bölgeler ..... 96

Şekil 4-31 Kolu içine alan doğrular ..... 97

Şekil 4-32 Doğruların oluşturduğu bölgeler ..... 99

Şekil 4-33 Kol noktaları ..... 100

Şekil 4-34 Alan değerlerinin bulunduğu bölgeler ..... 101

Şekil 4-35 (a) Saniye ve yelkovan kollarının çakıştığı görüntü ..... 104

(b) Akrep ve yelkovan kollarının çakıştığı görüntü……..104

(c) Saniye ve akrep kollarının çakıştığı görüntü……..104

Şekil 4-36 Üç kolun kesiştiği saat görüntüsü ..... 106

Şekil 4-37 Akrep, yelkovan ve saniye kollarının açıları..... 107

## ÇİZELGELER LİSTESİ

Sayfa

Çizelge 4-1 Eşik değerleri..... 62

Çizelge 4-2 Değişim noktaları alan değerleri..... 64

Çizelge 4-3 Doğru parçalarının başlangıç ve bitiş noktalarının koordinatları..... 71

Çizelge 4-4 1. ve 2. kesişim noktalarının koordinatları..... 71

Çizelge 4-5 $(x_n, y_n)$ ve $(x_m, y_m)$ noktalarının koordinatları ve noktalar arasındaki uzaklılar..... 74

Çizelge 4-6 Orta noktaya en uzak olan iskelet noktalarının orta noktaya olan uzaklıkları..... 75

Çizelge 4-7 En uzak nokta koordinatları ve açı değerleri..... 82

Çizelge 4-8 Eşik değerleri..... 87

Çizelge 4-9 Açı değerleri ve açılardaki nokta sayıları..... 92

Çizelge 4-10 Toplam nokta sayıları..... 92

Çizelge 4-11 En uzak iskelet noktası koordinatları ve kolların uzunlukları..... 96

Çizelge 4-12 Nokta koordinatları..... 98

Çizelge 4-13 Kolların alan değerleri..... 101

Çizelge 5-1 Saatin okuma sisteminin çözümleme süreleri..... 111

## SİMGELER VE KISALTMALAR LİSTESİ

| | | |
|---|---|---|
| $N_4(p)$ | : | $p$ pikselinin 4-komşuluğundaki piksellerin kümesi |
| $N_8(p)$ | : | $p$ pikselinin 8-komşuluğundaki piksellerin kümesi |
| $N_D(p)$ | : | $p$ pikselinin D-komşuluğundaki piksellerin kümesi |
| $V$ | : | Benzerlik kümesi |
| $D(p,q)$ | : | $p$ ve $q$ pikselleri arasındaki uzaklık fonksiyonu |
| $T$ | : | Eşik değeri |
| $\delta(x)$ | : | Benzerlik metriği |
| $D$ | : | Delik sayısı |
| $E$ | : | Euler numarası |
| $B$ | : | Bağlı bileşen sayısı |
| $(x_o, y_o)$ | : | Saatin orta noktası (saat konumun birinci bileşeni) |
| $R$ | : | Saniye kolunun, saatin orta noktasına en uzak noktasının orta noktaya uzaklığı (saat konumun ikinci bileşeni) |
| $m$ | : | Ortalama değer |
| $\sigma$ | : | Standart Sapma |
| $\theta_S$ | : | Saniye kolunun açısı |
| $\theta_A$ | : | Akrep kolunun açısı |
| $\theta_Y$ | : | Yelkovan kolunun açısı |

| | | |
|---|---|---|
| S.R.G | : | Tohumlu Bölge Büyütme |
| NTSC | : | National Television System Committee |

## KULLANILAN TERİMLER SÖZLÜĞÜ

| | | |
|---|---|---|
| Ağırlıklı Ortalama Süzgeç | : | Weighted Averaging Filter |
| Arayüz | : | Interface |
| Aydınlatma | : | Illumination |
| Bağlılık | : | Connectivity |
| Bayırgı | : | Gradient |
| Bitişiklik | : | Adjacency |
| Bölge | : | Region |
| Bölge Yarma Kaynaştırma | : | Region Spliting and Merging |
| Bölütleme | : | Segmentation |
| Dürüm | : | Iteration |
| Eşikleme | : | Thresholding |
| Hava Görüntüleri | : | Aerial Images |
| Izgara | : | Grid |
| İmza | : | Signature |
| İnceltme | : | Thining |
| İskelet | : | Skeleton |
| İşaretçi | : | Pointer |
| Görüntü | : | Image |
| Gösterge | : | Display |
| Gri Seviye | : | Gray Level |
| Gri Ton Ölçeği | : | Gray Scale |
| Karşıtlık | : | Contrast |
| Karşıtlık Artırma | : | Contrast Strecthing |
| Kenar | : | Edge |

| | | |
|---|---|---|
| Komşuluk | : | Neighborhood |
| Macenta | : | Magenta |
| Morfolojik Su Bölümü Çizgisi | : | Morphological Watershed |
| Onarma | : | Restoration |
| Orijin | : | Origin |
| Ortalama Süzgeç | : | Averaging Filter |
| Ortanca Süzgeç | : | Median Filter |
| Öklidyen | : | Euclidean |
| Öznitelik Çıkarma | : | Feature Extraction |
| Saat Kadranı | : | Hour Plate |
| Siyan | : | Cyan |
| Süzgeç | : | Filter |
| Tohum | : | Seed |
| Tohumlu Bölge Büyütme | : | Seeded Region Growing |
| Türdeş | : | Homogeneous |
| Uzamsal | : | Spatial |
| Yansıtma | : | Reflectance |
| Yol | : | Path |

# 1. GİRİŞ

Günümüzde, sayısal görüntü işleme yaygın olarak tıbbi tanı, uydu ile uzaktan algılama, endüstriyel ürün denetimi, parmak izi okuma, plaka okuma gibi birçok değişik alanda uygulanmaktadır. Sayısal görüntü işlemenin kullanıldığı alanlardaki birçok birim amaçlarını gerçekleştirebilmek için, çevrelerindeki görsel bilgiye ihtiyaç duyar. Bu tür birimler için, çevrelerindeki görsel bilgiyi elde etmenin en iyi yolu sahne analizi yapmaktır. Bu birimler için gerçekleştirilen sahne analiz sistemleri görsel bilgiye ihtiyaç duyulan çevreden kamera ile sayısal görüntüler elde ederler. Sahne analizi sistemleri ilk adım olarak, çevrelerinden elde ettikleri sayısal görüntünün bileşenlerini (nesneleri) görüntü işleme teknikleri kullanarak tanımalıdır. Sahne analizi gerçekleştirilirken en yaygın olarak kullanılan yöntemlerden bir tanesi nesne tanıma analizidir. Nesne tanıma analizi, ortak özellikler gösteren ve aralarında bir ilişki kurulabilen nesneleri veya şekilleri, belirlenen özellikleri kullanarak tanımlama veya sınıflandırma işlemidir. Tipik bir nesne tanıma analizi görüntü bölütleme, öznitelik çıkarma ve örüntü tanıma gibi üç adımdan oluşur. Daha sonraki adımlarda, gerçekleştirilen sistemler, çevrelerinden anlamlar çıkarmak için çevrelerinden elde ettikleri sayısal görüntüdeki farklı nesneler arasındaki ilişkileri belirleyebilmelidir.

Son 20 yıl içinde, değişik tiplerdeki sahneleri yorumlamak ve sahnelerden anlam çıkarmak için çeşitli çalışmalar yapılmıştır. Bu çalışmalar arasında, A.T.S. Ho, X. Zhu ve B. Vrusias'ın [1] kapalı ve açık alanlardaki sahnelerden delil elde etmek için yaptığı çalışma, Walter F. Bischof ve Terry Caelli'nin [2] sahneden 3 boyutlu nesne tanıma çalışması, Jyh-Yeong Chang ve Chien-Wen Cho'nun [3] açık alan sahnesi olan trafik sahnesinde öndeki aracı algılama çalışması ve Matsuyama ve Hwang'ın [4] ve Mckeown'un [5] hava görüntüleri üzerinde yaptığı çalışmalar verilebilir. Bahsedilen çalışmalarda birtakım görevler başarı ile gerçekleştirilmiş ise de hiçbirinde uygulamadan bağımsız genel amaçlı bir sahne analizi sunulmamıştır. Bu durum, her yeni uygulamada yeni bir sahne analizi yaklaşımı yapılmasının gerekliliğini göstermektedir.

Bu çalışmada bir sahne analizi uygulaması olarak saatin optik görüntüsünden saatin kaç olduğunun anlaşılması amaçlanmıştır. Bu amaçla, analog bir göstergesi (kadran) olan saatin optik görüntüsünden saati okuyan bir görüntü işlem sistemi gerçekleştirilmiştir. Okuma işlemi, analog göstergenin görüntüsü içindeki işaretçilerin ve ölçeklerin konumları algılanarak, analog göstergenin gösterdiği değerin bulunmasıdır. Gerçekleştirilen sistem ile, saatin işaretçileri olan akrep, yelkovan ve saniye kolu olan bir saatin görüntüsü, karşısına yerleştirilen video kamera ile algılanmıştır. Algılanan görüntü bilgisayar ortamında gerekli görüntü işleme teknikleri işlenerek çözümlenmiştir ve saatin kaç olduğu anlaşılmıştır.

Saatin kaç olduğunun anlaşılması işlemi iki aşamada gerçekleştirilmektedir. Birinci aşama saatin görüntü içindeki konumunun bulunmasıdır. İkinci aşama saatin okunmasıdır. Saatin konumunun bulunması, saatin kadranındaki işaretçilerden biri olan saniye kolunun farklı konumlarda algılanması ve farklı konumlardaki saniye kolu kullanılarak saatin orta noktasının ve saniye kolunun uzunluğunun bulunmasıdır. Saatin okunma işlemi, saat kadranındaki işaretçilerin algılanması ve işaretçilerin konumlarına göre yapılan hesaplamalarla gerçekleştirilmektedir.

Saat kadranın optik görüntüsünden saatin okunması işlemi Sablatnik ve Kropatsch [6] tarafından da yapılmıştır. Sablatnik ve Kropatsch'ın çalışmasında analog göstergelerin okunması için geliştirilen genel bir algoritma, analog göstergeli bir saati okumak için uygulanmıştır. Sablatnik ve Kropatsch'ın çalışmasında, işaretçilerin çakıştığı durumlar olmayan, örnek saat görüntüleri kullanılmıştır. Kullanılan örnek görüntüler tek bir tip saatten alınmıştır ve görüntüsü alınan saatin kameraya uzaklığı sabit tutulmuştur. Dairesel ölçeklerin algılanmasında, Hough dönüşümüne dayanan dairesel yay algılama yöntemleri kullanılmıştır [7,8]. İşaretçiler ise, dairesel ölçeklerin içinde kalan görüntü düzlemindeki gri seviye profiller kullanılarak algılanmıştır [6]. Sablatnik ve Kropatsch'ın çalışmasında saat okuma işleminin çözümleme sürecinin üzerinde durulmamıştır.

Bu çalışmadaki saat okuma sisteminde Sablatnik ve Kropatsch'ın çalışmasından farklı olarak, saatin kaç olduğunun anlaşılması işleminde işaretçilerin algılanması için, karşıtlık arttırma ve yumuşatma gibi düşük seviyeli görüntü işleme uygulamalarının yanında, yüksek seviyeli görüntü işleme adımları içeren nesne tanıma analizi kullanılmıştır. Nesne tanıma analizinin kullanılmasıyla, Sablatnik ve Kropatsch'ın çalışmasında gözardı edilen kolların çakışma durumlarında da saatin başarı ile okunması amaçlanmıştır. Gerçekleştirilen sistem ile önceden tanımlanan koşullara uyan analog göstergeli saatlerin, kollar görüntü içinde kaldığı sürece kameraya farklı uzaklıklarda olduğu durumlarda da okunması amaçlanmıştır. Bu çalışmada yine Sablatnik ve Kropatsch'ın çalışmasında üzerinde durulmayan saatin okunma işleminin çözümleme sürecinin, gerçek zamanlı olması amaçlanmıştır.

Bu çalışmadaki nesne tanıma analizinde, saniye kolunun görüntüden ayrılması, harekete dayalı bölütleme yöntemi ile gerçekleştirilmiştir. Saniye kolunun farklı konumlarda bulunduğu görüntüler kullanılarak fark görüntüleri elde edilmekte ve eşikleme yapılarak saniye kolu görüntüden ayırt edilmektedir. Saniye kolunun farklı konumlarda bulunduğu görüntülerde akrep ve yelkovan kolların konumları değişmediği için harekete dayalı bölütleme işlemi sonunda sadece saniye kolu görüntüden ayrılmış olur. Bu durumda, harekete dayalı bölütleme işlemi sonunda saniye kolu tanınmış da olmaktadır.

Akrep ve yelkovan kolları da tohumlu bölge büyütme yöntemi ile görüntüden ayrılmıştır. Kolların inceltilmiş hallerinden çıkarılan öznitelikler kullanılarak kolların (işaretçilerin) çakışma durumları algılanmıştır. Çakışma durumları belirlenen kollarının tanınmasında (birbirinden ayrılmasında), kollardan çıkarılan alan, kolların inceltilmiş hallerinden çıkarılan uzunluk ve açı öznitelikleri karşılaştırılarak, akrep ve yelkovan kollarının tanınması sağlanmaktadır.

Bu çalışmadaki saat okuma sisteminin çözümleme sürecinin gerçek zamanlı olması amaçlanmıştır. Saat okuma sisteminin gerçek zamanlı olabilmesi için, saat görüntüleri (sistemin girdileri) alındıktan sonra zamanın (sistemin çıktısı) anlaşılmasına ve görüntülenmesine kadar ki gecen sürenin bir saniyeden az olması gerekir. Bu çalışmada, uygulanan görüntü işleme algoritmalarında, gerçek

zamanlı sistemlerin gereksinimleri göz önüne alınmıştır ve bu konu ile ilgili noktalar, tezin bölümlerinde yeri geldikçe açıklanmıştır.

Çalışmanın ikinci bölümünde, sayısal görüntünün temelleri hakkında bilgi verilmiştir ve çalışmada kullanılan pikseller arası ilişkilerden söz edilmiştir.

Üçüncü bölümde, nesne tanıma analizinin temel adımları olan görüntü bölütleme, öznitelik çıkarma ve örüntü tanıma ve her bir adımda kullanılabilecek yöntemler üzerinde durulmuştur.

Dördüncü bölümde, saat okuma sistemi ve sistemin çalışması genel olarak ele alındıktan sonra, sistemin temel özellikleri, ön koşulları ve aydınlatmada yaşanabilecek sorunlar ve çözümleri üzerinde durulmuş ve son olarak da saat okuma işlemi için geliştirilen, birinci aşaması saatin konumunun bulunması, ikinci aşamasında ise saatin okunması olan iki aşamalı genel algoritma açıklanmıştır. Ayrıca, yapılan bir saat okuma işleminde, algoritmanın iki aşamasında da kullanılan nesne tanıma yöntemlerinin saat okuma problemine uygulanışı ve sonuçları verilmiştir.

Son bölümde ise tasarlanan sistemin ve kullanılan yöntemlerin başarısı ve performansı elde edilen sonuçlar çerçevesinde irdelenmiştir.

## 2. SAYISAL GÖRÜNTÜNÜN TEMELLERİ

Bu tez çalışmasında kullanılan bütün görüntüler *240x320* piksel boyutundadır. Görüntülerdeki bir piksel *8* bit ile ifade edilir. Pikselin alabileceği gri seviye değerleri *[0, 255]* arasındadır. Bu bölümde bu kavramları da içerecek şekilde sayısal görüntü hakkında temel bilgiler verilmektedir.

### 2.1. Siyah Beyaz Görüntü Modeli

Görüntü, ışık yoğunluğunu gösteren iki boyutlu bir fonksiyon olarak düşünülebilir. Bu *f(x, y)* fonksiyonunda $x$ ve $y$ düzlemsel koordinatları, fonksiyonunun *(x, y)* noktasındaki değeri resmin o noktadaki parlaklığını gösterir. Siyah beyaz bir resmin *(x, y)* noktasındaki parlaklığına resmin bu noktadaki gri ton değeri adı verilir. $x, y \in R$'dir ve *f(x, y)* sıfırdan büyük, sonlu bir sayıdır. *f(x, y)*'nin iki bileşeni vardır [9] :

**1)** Ortamdaki ışığın miktarı (aydınlatma),

**2)** Nesnenin ışığı yansıtma oranı (yansıtma),

Aydınlanmayı *i(x, y),* yansıtmayı *r(x, y)* ile gösterirsek *f(x, y)*'yi bunların çarpımı şeklinde yazılabilir [9] :

$$f(x,y) = i(x,y) * r(x,y) \qquad (2.1)$$

Burada teorik olarak

$$0 < i(x,y) < \infty \text{ ve } 0 < r(x,y) < 1 \qquad (2.2)$$

kabul edilir. *i(x, y)*'yi ışık kaynağı, *r(x, y)*'yi nesnelerin yansıtma özellikleri belirler. *r(x, y)* için 0 limit değeri ışığın tamamen emilmesi, 1 limit değeri ışığın tamamen yansıması demektir.

*f(x, y)* fonksiyonun *(x, y)* noktasındaki değerine $\lambda$ dersek Eşitlik 2.1 ve Eşitlik 2.2'deki denklemlerden $\lambda$'nın

$$L_{\min} < \lambda < L_{\max} \quad (2.3)$$

aralığında değiştiğini görürüz. [$L_{min}$, $L_{max}$] aralığına gri ton ölçeği adı verilir. Pratikte bu aralık *[0, L-1]* aralığındadır. Burada $\lambda = 0$ siyah ve $\lambda = L-1$ beyaz olarak kabul edilir. Aradaki değerler grinin tonlarına karşılık gelir.

## 2.2. Sayısal Görüntü

Görüntü üzerinde bilgisayarla işlem yapabilmek için *f(x, y)* görüntü fonksiyonunun koordinat sisteminin ve genliğinin sayısallaştırılması gerekir. *(x, y)* düzlemsel koordinat sisteminin sayısallaştırılmasına görüntü örnekleme, genliğinin sayısallaştırılmasına gri ton ayrıklaştırılması denir. Bu durumda sayısal görüntü hem düzlemsel koordinatların, hem de parlaklığın ayrık biçimde olduğu bir fonksiyonudur. Eşitlik 2.4'de *f(x, y)* fonksiyonu sayısal görüntüyü göstermektedir.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & \cdots & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.4)$$

Eşitlik 2.4'deki denklemin sağ tarafı, yani sayısal görüntü, satır ve sütun indislerinin görüntüdeki bir noktanın koordinatlarını, bu noktadaki elemanın gri seviye değerini gösterdiği bir matris olarak ifade edilebilir. Bu şekildeki bir matris dizisinin her elemanına piksel denir [9].

Sayısallaştırma işleminde *N, M* değerlerini ve ayrık gri seviye sayısını belirlemek gerekir. Gri seviyelerin sayısı $L$, bilgisayar uygulamalarında genellikle 2'nin pozitif tamsayı kuvvetleri olarak seçilir.

$$N = 2^n, M = 2^m, L = 2^k \quad (2.5)$$

Eşitlik 2.5'deki gri seviye değerlerinin *[0, L-1]* aralığında ve eşit aralıklı ayrık değerler olduğunu kabul ediyoruz. Görüntüler bilgisayar ortamında işlem gördüğüne ve saklandığına göre, bir resim saklamak için $N*M*k$ bir hafızaya ihtiyaç vardır. Bu parametreler büyüdükçe sayısal görüntü gerçek görüntüye daha

çok yaklaşır. Ancak resim için gereken hafıza miktarı ve görüntü işleme için gereken zaman da artacaktır. İyi bir resmin tanımını yapmak zordur çünkü çeşitli uygulama alanlarının ihtiyaçları birbirinden farklıdır.

## 2.3. Pikseller Arasındaki İlişkiler

### 2.3.1. Komşuluk

$(x, y)$ koordinatındaki bir $p$ pikselinin, koordinatları Eşitlik 2.6'da verilen dört adet dikey ve yatay komşusu vardır.

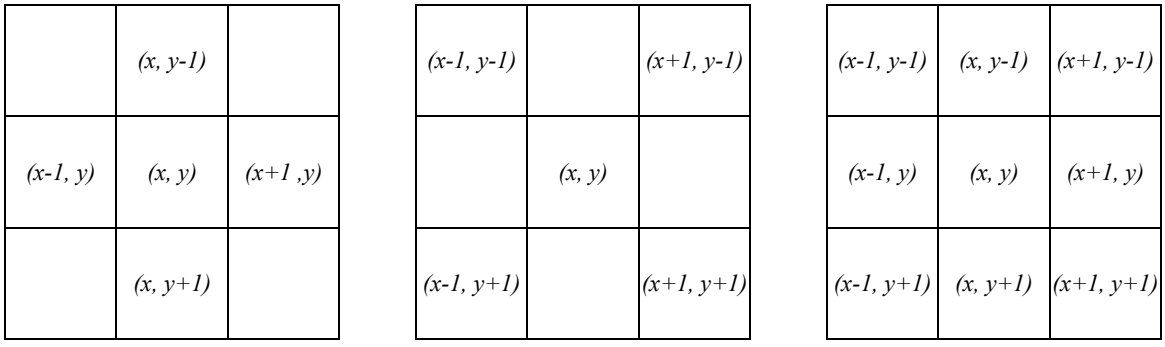
$$(x+1, y), (x-1, y), (x, y+1), (x, y-1) \quad (2.6)$$

Bu piksel kümesine $p$'nin 4-komşuluğu denir ve $N_4(p)$ ile gösterilir.

$(x, y)$ koordinatındaki $p$ pikselinin, koordinatları Eşitlik 2.7'de verilen 4 adet de köşegen komşusu vardır.

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) \quad (2.7)$$

Bu piksel kümesine de $p$'nin D-komşuluğu denir ve $N_D(p)$ ile gösterilir. $N_D(p)$ ve $N_4(p)$ piksel kümelerinin oluşturduğu piksel kümesine de $p$'nin 8-komşuluğu denir ve $N_8(p)$ ile gösterilir. Eğer $(x, y)$ görüntünün sınırında ise, $N_4(p)$, $N_8(p)$ ve $N_D(p)$ kümelerindeki bazı pikseller görüntünün dışında kalabilir. Şekil 2-1'de $(x, y)$ koordinatındaki pikselin komşulukları görülmektedir.




$N_4(p)$ $N_8(p)$ $N_D(p)$

**Şekil 2-1** Piksel komşulukları

### 2.3.2. Bağlılık ve bitişiklik

Pikseller arasındaki bağlılık, sayısal görüntü işlemede, bölgelerin ve nesnelerin bulunmasında kullanılan önemli bir kavramdır. İki pikselin bağlılığından söz edebilmek için önce bu bağlılığın temelini oluşturacak bir komşuluk kararlaştırmak gerekir. Bundan sonra, kararlaştırılan bu komşulukta bulunan iki pikselin gri seviye değerlerinin, önceden tanımlanmış bir benzerlik kriterine uygunluğu kontrol edilir. Burada benzerliğin tanımı, iki pikselin gri seviye değerlerinin de, benzerlik kriteri olarak seçilen bir $V$ kümesinin elemanı olmalarıdır. İkili görüntüde benzerlik kümesi $V = \{0, 1\}$ ile sınırlıdır.

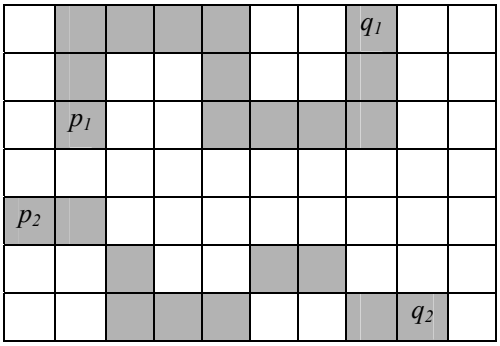
Gri seviye değerleri $V$'nin elemanı olan $p$ ve $q$ pikselleri, eğer $q$ $N_4(p)$ kümesinde ise 4-bitişiktir, eğer $q$ $N_8(p)$ kümesinde ise 8-bitişiktir. Bu iki bitişiklik çeşidinden başka, m-bitişiklik adı verilen karma bir tür de tanımlanabilir. Gri seviye değerleri $V$'nin elemanı olan $p$ ve $q$ pikselleri, eğer $q$ $N_4(p)$ kümesinde ise, veya $q$ $N_D(p)$ kümesinde ve $N_4(p) \cap N_4(q)$ kümesi boş ise (yani $p$ ve $q$'nun elemanları $V$ kümesinde değilse) m-bitişiktir.

### 2.3.3. Yol ve bölge

$(x_0, y_0)$ koordinatındaki $p$ pikselinden, $(x_n, y_n)$ koordinatındaki $q$ pikseline yol,

$$(x_0, y_0), (x_1, y_1), ...., (x_n, y_n) \qquad (2.8)$$

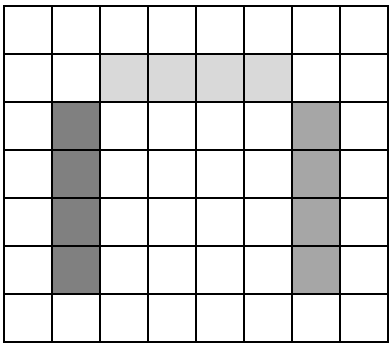
koordinatlarındaki birbirinden farklı pikseller kümesidir. $1 \leq i \geq n$ olmak üzere $(x_i, y_i)$ pikseli $(x_{i-1}, y_{i-1})$ pikseline bitişiktir ve bu durumda bu yolun uzunluğu $n$'dir. Belirlenen bitişiklik türüne göre 4-, 8- veya m-yol tanımlanabilir. Şekil 2-2'de, 4-komşuluk tanımlandığında sadece $p_1$ pikselinden $q_1$ pikseline beyaz olmayan piksellerin oluşturduğu yol vardır, 8-komşuluk tanımlandığında hem $p_1$ pikselinden $q_1$ pikseline hem de $p_2$ pikselinden $q_2$ pikseline beyaz olmayan piksellerin oluşturduğu yol vardır.




**Şekil 2-2** Pikseller arasındaki yollar

$p$ ve $q$'nun, bir görüntünün S ile ifade edilen altkümesinin iki pikseli olduğunu varsayalım. $p$'nin $q$'ye S içinde bağlı olma koşulu, $p$'den $q$'ye tamamen S içinde bulunan piksellerden oluşan bir yol bulunmasıdır.

S kümesi içinde herhangi bir $p$ pikseline bağlı olan piksellere S kümesinin bağlı bileşeni denir. Eğer tek bir tane bağlı bileşen var ise S kümesine bölge adı verilir. Bağlılığın, bölge tanımlamasındaki önemi Şekil 2-3'den anlaşılır. Şekildeki pikseller arasında 4-komşuluk tanımlanmış ise farklı gri tonlarla ayrılmış ve beyaz olmayan piksellerin oluşturduğu üç bölge, 8-komşuluk tanımlanmış ise beyaz olmayan piksellerin oluşturduğu bir bölge vardır.



**Şekil 2-3** Örnek bölgeler

### 2.3.4. Uzaklık ölçümü

Koordinatları sırasıyla *(x, y), (s, t), (v, w)* olan *p, q* ve *z* pikselleri için,

**(a)** $D(p,q) \geq 0 \quad (D(p,q) = 0 \quad eğer \quad p = q)$

**(b)** $D(p,q) = D(q,p)$

**(c)** $D(p,z) \leq D(p,q) + D(q,z)$

ise $D$ uzaklık fonksiyonudur.

$p$ ve $q$ pikselleri arasındaki öklidyen uzaklık aşağıdaki şekilde tanımlanır,

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2} \qquad (2.9)$$

Öklidyen uzaklıkta, $(x, y)$ pikselinden uzaklığı $r$ gibi bir değerden küçük veya aynı olan pikseller, merkezi $(x, y)$ koordinatında olan, yarı çapı $r$ olan bir diskin içinde kalırlar.

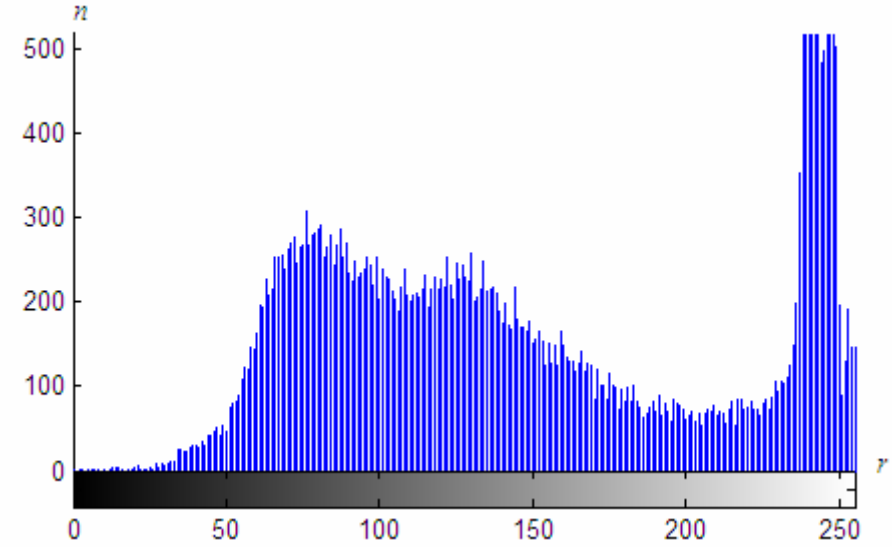
## 2.4. Görüntü Histogramı

Gri seviye değerleri $[0, L-1]$ aralığında değişen sayısal bir görüntünün histogramı,

$$h(r_k) = n_k \qquad (k = 0, 1, \ldots, L-1) \qquad (2.10)$$

şeklinde tanımlanmış ayrık fonksiyondur. $r_k$ $k.$ gri seviye değerini, $n_k$ $k.$ gri seviye değerine sahip piksellerin sayısını gösterir [9]. $h(r_k)$ fonksiyonu görüntüdeki toplam piksel sayısını gösteren $N$ değerine bölünerek $p(r_k)=n_k/N$ şeklinde normalize edilir. $p(r_k)$, gri seviye değerlerinin görüntüde bulunma olasılıklarının verir. Şekil 2-4(a)'daki örnek görüntünün histogramı Şekil 2-4(b)'de görülmektedir.

**(b)**




**(b)**

**Şekil 2-4 (a)** Örnek bir görüntü **(b)** Örnek görüntünün histogramı

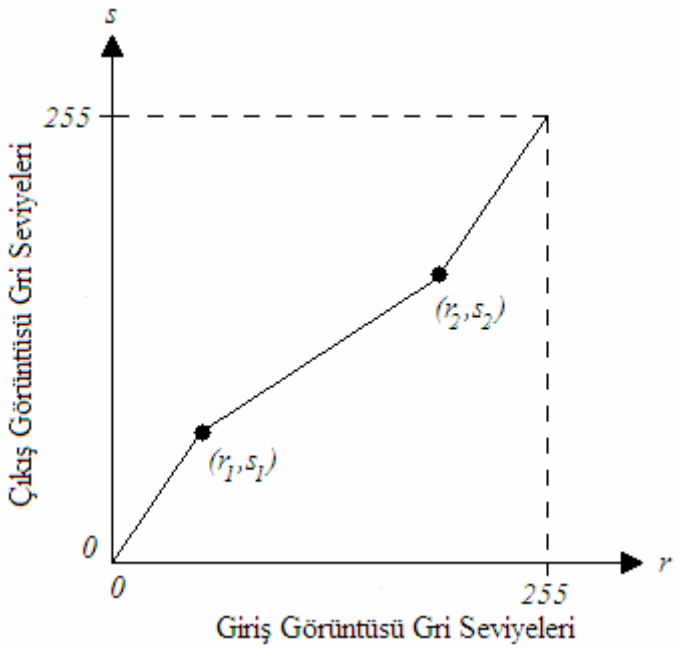
Görüntü histogramı, görüntünün genel görünümü hakkında bilgi verir. Histogramda, piksellerin yoğunlaştığı bölgelere bakılarak yorum yapılabilir. Pikseller, histogramın düşük gri seviye değerlerini gösteren bölgelerinde yoğunlaşmış ise, görüntünün genel olarak karanlık olduğu sonuçu çıkarılabilir. Pikseller yüksek gri seviye değerlerinde yoğunlaşıyorsa görüntünün aydınlık olduğu sonuçuna varılabilir. Pikseller histogramın orta bölgelerinde yoğunlaşıyorsa, görüntüde gri tonun hakim olduğu sonuçu çıkarılabilir. Pikselleri histogram üzerinde eşit dağılmış ise görüntünün karşıtlığının yüksek olduğu sonucuna varılabilir. Histogramda pikseller dar alanlarda yoğunlaşıyorsa, görüntü üzerinde uygulamalar yapmadan önce ön işlem olarak kontrast artırma işlemi ile görüntünün karşıtlığı artırılabilir.

Histogram, her bir gri seviye değerine sahip piksellerin sayısını vermesine rağmen, gri seviye değerlerinin görüntü düzlemindeki dağılımı hakkında bir bilgi vermediği için görüntü içeriği hakkında histograma bakılarak fikir edinilemez. Görüntünün belli bir parçası ile ilgilenilirse, bu görüntü parçasına ait histogramın hesaplanması gerekir.

## 2.5. Karşıtlık Artırma

Karşıtlık artırma, görüntüdeki gri seviyelerin dinamik dağılımını artıran bir dönüşüm fonksiyonudur. Karşıtlık artırma, görüntü elde edilirken, aydınlatmanın yetersiz olması veya kameranın gri seviyedeki dinamik dağılımı sağlayamadığı düşük karşıtlıklı görüntülere uygulanabilir.

Tipik, doğrusal karşıtlık artırma dönüşüm fonksiyonun biçimi Şekil 2-5'de görülmektedir. $(r_1, s_1)$ ve $(r_2, s_2)$ noktalarının konumu karşıtlık artırma fonksiyonun özelliğini belirler. Genel olarak, $r_2 \geq r_1$ ve $s_2 \geq s_1$ olarak seçilerek, dönüşüm fonksiyonun tek değerli ve tekdüzeli olarak artması sağlanır [9]. Dönüşüm fonksiyonun bu özelliklere sahip olması görüntüdeki gri seviyelerin sıralarının korunmasını sağlayarak, çıkış görüntüsünde gri seviyelerdeki bozulmaları önler.




**Şekil 2-5** Karşıtlık artırma dönüşüm fonksiyonu [9]

# 3. İKİ BOYUTLU NESNE TANIMA

Nesne tanıma analizi, gerçek dünya görüntülerinde bulunan nesneleri ayırma ve ait olduğu sınıflara atama işlemidir.

İnsanlar nesne tanıma işlemlerini farkında olmadan, fazladan güç sarf etmeden gerçekleştirir. İnsanların kolayca gerçekleştirdiği nesne tanıma işlemlerini bilgisayarlara yaptırmak için gereken algoritmaların tanımlanması her zaman güç olmuştur. Tipik bir nesne tanıma analizi üç temel adımdan oluşur:

1. Görüntü Bölütleme: Nesneyi veya nesneleri görüntüden soyutlama.

2. Öznitelik Çıkarma: Görüntüden soyutlanmış olan nesneyi dönme, öteleme ve büyüklük değişimlerinden bağımsız bir şekilde ifade eden özniteliklerini çıkarma.

3. Örüntü Tanıma: Görüntüden bölütlenmiş nesneden çıkarılan öznitelikleri kullanarak nesneyi doğru sınıfa atama.

Nesne tanıma karmaşık ve yaratıcılık isteyen bir analizdir. Her problemin kendine özgü yapısı ve çözümü vardır. Bu nedenle nesne tanıma analizi gerçekleştirilmeden önce problemin tanımı ayrıntılı şekilde yapılmalıdır.

Bir nesne tanıma analizindeki karmaşıklık aşağıdaki konulardan kaynaklanabilir.

- Ortamın Sabitliği: Aydınlatma, kamera parametreleri, ve kameranın konumunun sabit olmaması tanıma analizinin karmaşıklığını arttırır. Bu parametrelerin değişken olması nesne tanıma sisteminin bu değişimlere karşı dinamik olarak düzeltilmesini gerektirebilir veya parametreler değişmeyecek şekilde ortam sabitlenir.

- Görüntüdeki Nesne Sayısı: Görüntüde tanınacak birden fazla nesne olması bölütleme algoritmasının karmaşıklığını arttırır. Görüntüde birden fazla olan

nesneler birbirileriyle çakışabilir. Nesnelerin çakışması görüntüdeki iki boyutlu özniteliklerini bozar ve yanlış sınıflandırılmalarına yol açar.

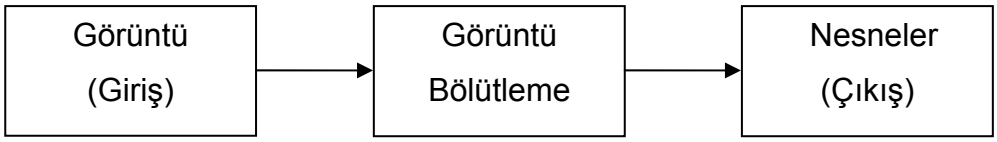
- Perspektif etkisi: Gerçek nesneler üç boyutludur. Fakat kamera ile edilen görüntüler iki boyutludur. Bu durumda iki farklı nesne farklı perspektiflerden aynı öznitelik vektörüne sahip olabilir.

Bu bölümde görüntü bölütleme, öznitelik çıkarma ve örüntü tanıma konularından bahsedilecektir.

## 3.1. Görüntü Bölütleme

Görüntü bölütleme, homojen bir küme içerisinde aynı karakteristiğe sahip alanların çıkarılması, arka plandan ayrılması ve belirgin bir hale getirilmesi işlemine denir [10].

Şekil 3-1'de temel görüntü bölütleme sistemi görülmektedir. Bölütleme işleminin girişi görüntüdür, çıkışı ise görüntüden bölütlemek istediğimiz nesnelerin işlenmemiş sınır pikselleri veya nesnenin bölge piksellerdir.




**Şekil 3-1** Görüntü bölütleme sistemi

Nesne tanıma analizinde ilk ve en zor adım görüntü bölütlemedir. Nesne tanıma analizinde daha sonra yapılacak olan öznitelik çıkarımı ve örüntü tanıma gibi yüksek seviyeli uygulamaların ve analizlerin başarısı gerçekleştirilen bölütlenmenin başarısına önemli ölçüde bağlı olacağı için görüntü bölütleme nesne tanımada önemli bir etkiye sahiptir.

Bir bölütleme algoritmasından iyi bir sonuç alınabilmesi için algoritma hangi matematiksel modele bağlanmış olursa olsun görüntü hakkındaki anlamsal veriler

ve önceki bilgilerden oluşan sezgiselliğin de çözüm yöntemine katılması gerekir [11]. Literatürde birçok bölütleme tekniği mevcuttur fakat hiçbiri bütün görüntüler için iyi bir teknik değildir [12,13]. Tüm görüntüler için iyi sonuç veren genel bir algoritma olmadığı gibi aynı algoritma farklı görüntüler üzerinde farklı sonuçlar verebilmektedir [14]. Algoritmalar genellikle görüntülerin gürültüye karşı duyarlılık, aydınlatma koşulları ve büyüklükleri gibi özelliklerinden dolayı her zaman iyi performans gösteremeyebilmektedirler [11].

Bölütleme algoritmaları için genelde 4 temel yaklaşım vardır [15,16]:

1. Eşikleme Yöntemi ile Bölütleme

2. Kenar Tabanlı Bölütleme

3. Bölgesel Tabanlı Bölütleme

4. Karma Bölütleme

Eşikleme yöntemi ile bölütlemede nesneye ait pikseller kullanılarak bölütleme yapılır. Piksellerden, gri seviye değerleri belirli değer aralığında olanlar bir gruba ait olurlar [17]. Gruplandırılan pikseller nesneye ait pikseller olurlar. Eşikleme yönteminde piksellerin uzamsal bilgisi göz ardı edilir [18].

Kenar tabanlı bölütlemede nesnenin sınırları yani kenarları kullanılarak bölütleme yapılır [7]. Nesnenin sınırları, gri seviyede devamsızlık gösterdiğinden kenar tabanlı bölütlemede aslında gri seviyedeki devamsızlıklar kullanılarak bölütleme yapılır. Görüntüye, bayırgı işleçleri olan Sobel veya Robert filtreleri uygulanabilir [19]. Uygulama sonucunda yüksek değer veren pikseller nesne sınırlarına aday olur ve daha sonra değiştirilerek nesne sınırlarını temsil eden kapalı eğriler üretilir [18].

Bölgesel tabanlı bölütlemede de, nesneye ait pikseller kullanılarak bölütleme yapılır. Pikseller, gösterdikleri uzamsal yakınlıktaki ve/veya gri seviyedeki, önceden belirlenen benzer özelliklere göre gruplandırılırlar.

Gruplandırılan bağlantılı piksellerde bölgeyi oluşturur ve oluşturulan bu bölgeler görüntüdeki nesneleri temsil eder.

Karma bölütleme yönteminde hem nesneye ait pikseller hem de nesnenin sınırları kullanarak bölütleme yapılır. Morfolojik su bölümü çizgisi (Morphological Watershed) yöntemi en temel karma bölütleme yöntemidir [20].

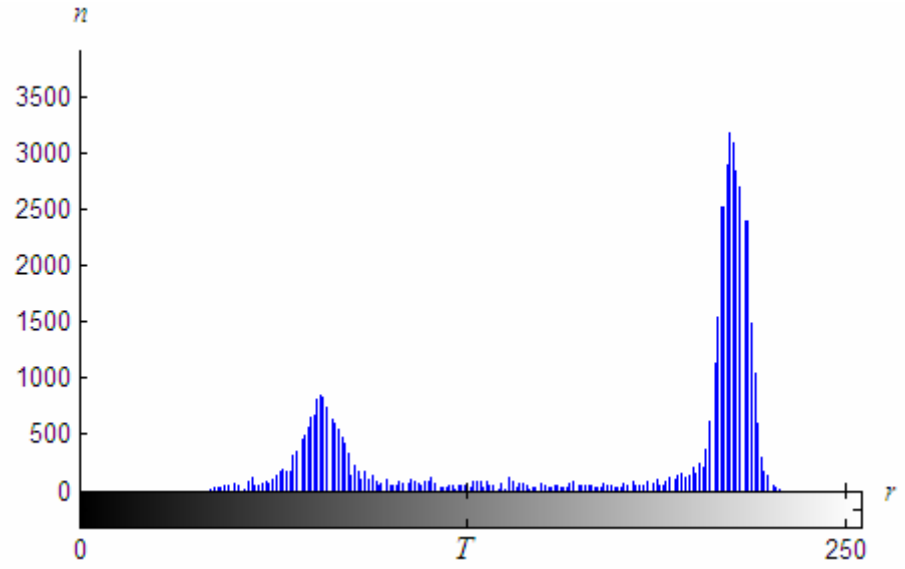
Bölütleme algoritmalar için dört temel yaklaşımın yanında, harekete dayalı bölütleme de görüntü bölütlemede kullanılan bir yöntemdir. Görüntüden bölütlenmek istenen nesne hareketli bir nesne ise, nesnenin hareket etme özelliği kullanılarak görüntüden bölütleme işlemi yapılır.

### 3.1.1. Eşikleme yöntemi ile bölütleme

Eşikleme, gri seviye ölçeğinde $L$ adet değer bulunan bir görüntüyü ele alıp, bu görüntüden gri ton ölçeği sadece iki adet değer içeren ikili görüntü elde etmektir. Birçok uygulamada *[0, L-1]* aralığında $L$ adet değerden bir değer seçilir. Hangi değerin seçileceği görüntü histogramında yapılacak çalışmaya göre belirlenir. Eşikleme global olarak uygulanırken gri seviye üzerinde bir eşik değer saptanır ve bu eşik değerinden küçük olan değerler yerine sıfır, büyük olan değerler yerine *L-1* yerleştirilir. Basit eşikleme algoritması Eşitlik 3.1'de tanımlanmıştır.

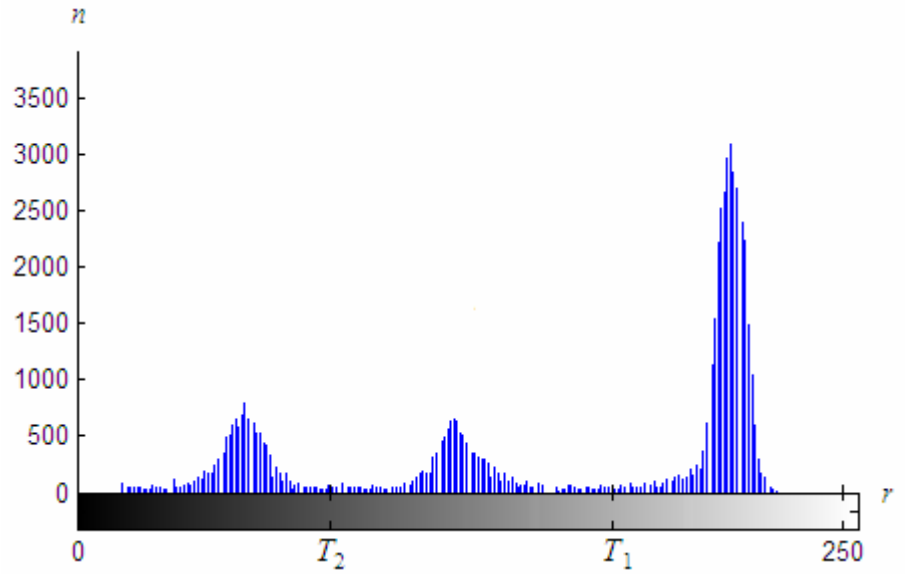
$$g(x,y) = \begin{cases} L-1 & eğer\ f(x,y) > T\ (Eşik\ Değeri) \\ 0 & eğer\ f(x,y) \le T\ (Eşik\ Değeri) \end{cases} \qquad (3.1)$$

Şekil 3-2'de, yüksek gri seviye değerli piksellere sahip arka plan üzerinde, düşük gri seviye değerli piksellere sahip nesneden oluşan bir görüntünün histogramı görülmektedir. Histogramdan anlaşılacağı gibi görüntü, homojen gri seviyeye sahip bir nesneden ve arka plandan oluşuyorsa bu tip görüntülerin histogramına iki doruklu histogram denir [21]. İki doruklu histograma sahip görüntülerde nesneyi arka plandan ayırmak için iki doruğu ayıracak eşik değeri $T$ seçilir. Herhangi bir *(x, y)* noktasında $f(x, y) < T$ ise o nokta nesne noktası, değilse arka plan noktası olarak belirlenir.




**Şekil 3-2** İki doruklu histogram

Eğer görüntüde birden fazla nesne varsa, bölütlemek için birden fazla eşik değeri kullanılabilir. Şekil 3-3'de, yüksek gri seviye değerli piksellere sahip arka plan üzerinde, düşük gri seviye değerli piksellere sahip iki ayrı nesneden oluşan bir görüntünün histogramı görülmektedir. Herhangi bir $(x, y)$ noktasında $T_1 \geq f(x, y) > T_2$ ise o nokta nesnelerden bir tanesine ait olur, $f(x, y) < T_2$ ise o nokta diğer nesneye ait olur. Genelde birden fazla eşik değeri gerektiren bölütleme işlemlerinde bölge büyütme tekniği daha iyi sonuç verir [9].




**Şekil 3-3** Birden fazla nesne içeren görüntünün histogramı

Eşikleme tekniğinde eşik değerinin seçimi görüntünün histogramına dayanır. Bir gözlemci görüntü histogramına bakarak eşik değerini belirleyebilir. Bu tür bir buluşsal eşik değeri seçimi yani histogram gözlemine dayalı bir seçim aydınlatmanın kontrollü olduğu ve otomatik olmayan bir seçimdir [9]. Eğer görüntü histogramı dorukların anlamlı ve belirgin olduğu bir histogram ise eşik değeri $T$ otomatik yöntemlerle iki doruğun arasında ortalarda bir değer bulunabilir.

İki doruklu histograma sahip görüntülerde otomatik eşik değeri bulan ve sık kullanılan yöntemlerden olan ve tez çalışmasında kullanılan iki eşik değeri bulma algoritmasından birincisi aşağıdaki adımlardan oluşmaktadır [9]:

1. **Adım:** Keyfi bir eşik değeri $T$ seçilir.

2. **Adım:** Gri seviye değeri $T$ değerinden küçük veya eşit olan piksellerin gri seviyelerinin ortalaması $\mu_1$ ve gri seviye değeri $T$ değerinden büyük olan piksellerin gri seviyelerinin ortalaması $\mu_2$ hesaplanır.

3. **Adım:** Yeni eşik değeri Eşitlik 3.2'deki gibi bulunur.

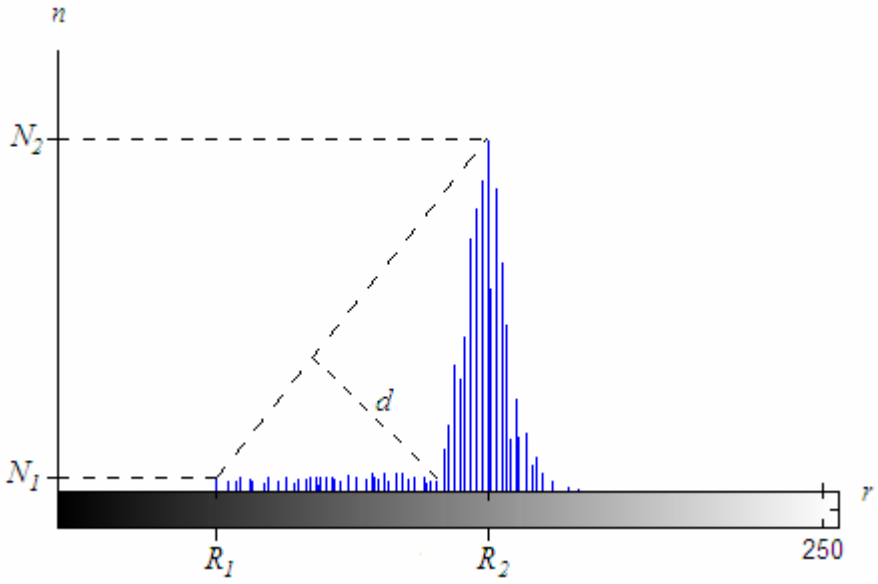
$$T = (\mu_1 + \mu_2)/2 \qquad (3.2)$$

4. **Adım:** Algoritmanın dürümü sonrasındaki $T$ değerleri arasındaki fark sıfır olana kadar, 2. ve 3. adımlar tekrar edilir.

1. adımda keyfi olarak seçilen eşik değeri $T$ için iyi bir seçim, eğer görüntüden bölütlenecek nesnenin ve arka planın kapladığı alanlar birbirine yakınsa, gri seviyelerin ortalaması olabilir. Eğer nesnenin ve arka planın kapladığı alanlar arasında çok fark varsa eşik değeri $T$ için iyi bir seçim, en yüksek ve en düşük gri seviyelerin ortasında bir değer olabilir.

Şekil 3-4'de yüksek gri seviye değerli piksellere sahip arka plan üzerinde, düşük gri seviye değerli piksellere sahip ve arka plana göre çok az yer kaplayan nesneden oluşan görüntünün histogramı görülmektedir. Bu histogramda arka plan

piksellerinin oluşturduğu doruk, nesne piksellerinin oluşturduğu doruğa göre çok yüksektir.




**Şekil 3-4** Örnek bir histogram

Bu biçimde bir histograma sahip görüntüde otomatik eşik değeri bulan ve tez çalışmasında kullanılan ikinci eşik değeri bulma algoritması aşağıdaki adımlardan oluşmaktadır :

1. **Adım:** Histogramdaki en çok sayıda pikseli gösteren nokta $(R_2, N_2)$ bulunur.

2. **Adım:** Histogramdaki diğer kümenin (nesne piksellerinin kümesinin) bulunduğu tarafta en uçtaki gri seviye değerinde kaç tane piksel olduğunu gösteren nokta $(R_1, N_1)$ bulunur.

3. **Adım:** $(R_2, N_2)$ ve $(R_2, N_2)$ noktalarının birleştiren bir doğru çizilir. Bu iki noktadan geçen doğrunun denklemi Eşitlik 3.3'de görülmektedir.

$$(N_2 - N_1)r + (R_1 - R_2)n + R_2N_1 - R_1N_2 = 0 \qquad (3.3)$$

4. **Adım:** $(R_1, N_1)$ ve $(R_2, N_2)$ noktalarının arasındaki histogram değerlerinin, Eşitlik 3.3'de gösterilen doğruya uzaklıkları bulunur. Histogram değerlerinin, $k=0, 1, \ldots, L-1$ olmak üzere $h_k$ ile gösterildiğini kabul edersek, $R_1$ ve $R_2$ gri

seviyeler arasında kalan noktaların doğruya uzaklıkları Eşitlik 3.4'deki denklem ile hesaplanır.

$$d_i = \frac{\left|(N_2 - N_1)i + (R_1 - R_2)h_i + R_2 N_1 - R_1 N_2\right|}{\sqrt{(N_2 - N_1)^2 + (R_1 - R_2)^2}} \qquad i = R_1, R_1 + 1, \ldots, R_2 \qquad (3.4)$$

5. **Adım:** En büyük $d_i$ değerini veren gri seviye, eşik değeri olarak belirlenir.

Histogramlar her zaman iki ayrı belirgin gruptan oluşmayabilir ve bu tip histogramlara sahip görüntüler tek bir eşik değeri ile bölütlenemeyebilir. Tek bir eşik değeri ile bölütlenemeyen görüntüler, daha küçük boyutta alt görüntülere ayrılır ve her alt görüntü ayrı eşik değeri ile bölütlenir [9]. Bu yöntemde önemli olan, görüntünün nasıl alt görüntülere ayrılacağı ve alt görüntülerde eşik değerinin nasıl seçileceğidir.

Eğer histogram gürültülüyse, bulunan eşik değeri bölütleme için iyi sonuç vermeyebilir. Histogramın gürültülü olduğu durumlarda, histogram yumuşatma işlemi uygulanabilir [21].

#### 3.1.1.1. Görüntü yumuşatma

Yumuşatma görüntüyü bulanıklaştırır ve gürültüyü azaltır. Yumuşatma işlemi ile görüntüde ilgilenilen esas nesneye oranla gürültü olarak kabul edilecek küçük detay noktaları görüntüden çıkarılır. Yumuşatma işlemi yumuşatma süzgeçleri kullanılarak gerçekleştirilir. Yumuşatma süzgeçleri, görüntüdeki yüksek frekans bileşenlerini görüntüden çıkararak görüntüye bulanıklık etkisi verir. En yaygın kullanılan yumuşatma süzgeçleri, ortalama süzgeçleri, ortanca süzgeçler ve Gaussian süzgeçlerdir.

Yumuşatma süzgeçleri de, genel süzgeçleme tekniği kullanılarak görüntüye uygulanır. Süzgeçleri görüntüyü bir durumdan başka bir duruma dönüştüren matrisler olarak düşünebiliriz. Değişik boyutlarda süzgeçler mevcuttur fakat en yaygın kullanılan süzgeç Şekil 3-5'de görüldüğü gibi 3x3 boyutundadır.

| | | |
|---|---|---|
| *s(x-1, y-1)* | *s(x, y-1)* | *s(x+1, y-1)* |
| *s(x-1, y)* | *s(x, y)* | *s(x+1, y)* |
| *s(x-1, y+1)* | *s(x, y+1)* | *s(x+1, y+1)* |

**Şekil 3-5** 3x3 boyutunda süzgeç matrisi

*s(i,j)* matrisin katsayılarıdır. Süzgeç matrisi görüntü üzerinde gezdirilerek her pikselin yeni değeri ayrı ayrı hesaplanır. Süzgecin katsayılarının değerleri süzgeçlemenin amacına göre değişir. Pikselin yeni değerinin hesaplanmasında pikselin 8-komşuluğundaki pikseller, ağırlıkları matrisin katsayıları tarafından belirlenerek, kullanılır.

Görüntüdeki her piksel için Eşitlik 3.5'deki denklem kullanılarak *g(x,y)* hesaplanır.

$$\begin{aligned} g(x,y) = {} & f(x-1,y-1)\times s(x-1,y-1) + f(x,y-1)\times s(x,y-1) \\ & + f(x+1,y-1)\times s(x+1,y-1) + f(x-1,y)\times s(x-1,y) \\ & + f(x,y)\times s(x,y) + f(x+1,y)\times s(x+1,y) \\ & + f(x-1,y+1)\times s(x-1,y+1) + f(x,y+1)\times s(x,y+1) \\ & + f(x+1,y+1)\times s(x+1,y+1) \end{aligned} \qquad (3.5)$$

Süzgeçleme işlemi sırasında, orijinal görüntünün pikselleri bütün piksellerin yeni değerleri hesaplanana kadar korunur. Yeni değerler süzgeçleme işlemi sırasında kullanılmaz. Yeni piksel değerleri, süzgeçlenmiş görüntünün piksel değerlerini gösterir.

Yumuşatma süzgeçleri gürültüyü azaltmada başarılı olsalar da, görüntüdeki kenar noktalarını korumak gibi özellikleri yoktur. Yumuşatıcı süzgeçlerin en büyük dezavantajı görüntüdeki kenarların keskinliğini de gürültüyle birlikte görüntüden çıkarmasıdır. Bazı yumuşatıcı süzgeçler Şekil 3-6'da görülmektedir. Dikkat edilirse süzgeçlerin katsayıları yumuşatma amaçlarına göre birbirinden farklıdır.

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} \qquad \frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

**(a)** **(b)**

**Şekil 3-6 (a)** Ortalama süzgeci **(b)** Ağırlıklı ortalama süzgeci

### 3.1.2. Bölgesel tabanlı bölütleme

Bölgesel tabanlı bölütleme temel olarak, $R$ ile temsil edilen görüntünün aşağıdaki özellikleri sağlayacak biçimde, $R_1$, $R_2$,…, $R_n$ ile temsil edilen parçalarına bölütlenmesi işlemidir [9].

**1.** $\bigcup_{i=1}^{n} R_i = R.$

**2.** $R_i$ bağlı komşulukları olan bir bölgedir, $i = 1,2,...n.$

**3.** Tüm $i$'ler ve $j$'ler için $i \neq j$ olmak üzere $R_i \cap R_j = \varnothing$.

**4.** Tüm $i$'ler için $P(R_i) = DOĞRU.$
($R_i$ bölgesinin tüm piksel değerleri bir benzerlik kriterine göre benzer ise $DOĞRU$ değeri üretilir).

**5.** $i \neq j$ iken $P(R_i \cup R_j) = YANLIŞ.$

1. özellik, bölütleme işleminin tamamlandığını gösterir. Yani bir bölgeye ki bu bölge nesne bölgesi veya arka plan bölgesi olabilir, dahil olmayan piksel kalmadığını gösterir. 2. özellik, bölütlenen bölgelerdeki piksellerin, kendi bölgesindeki piksellerle bağlı olduğunu gösterir ki zaten bu özellik Bölüm 2.3.3'de açıklandığı gibi bölgeyi bölge yapan özelliktir. 3. özellik, ise bir bölgeye dahil olan bir pikselin başka bir bölgeye dahil olmayacağını gösterir. 4. özellik, bir bölgeye dahil olan piksellerin bağlılık koşulunda belirlenen gri seviye değerlerindeki benzerlik kriterini sağladığını gösterir. 5. özellik, iki farklı bölgeye dahil olan piksellerin bağlılık koşulunda belirlenen gri seviye değerlerindeki benzerlik kriterlerinin birbirinden farklı olduğunu gösterir.

Bölge büyütme ve bölge yarma-kaynaştırma yöntemleri bölgesel tabanlı bölütlemeye örneklerdir [9].

#### 3.1.2.1. Bölge büyütme

Bölge büyütme, pikselleri veya alt bölgeleri önceden belirlenen kriterlere göre gruplandırarak daha büyük bölgeler elde etme yordamıdır [9].

Tohumlu bölge büyütme (S.R.G), bölge büyütmede en temel yaklaşımdır. S.R.G, bölütlemeyi tohum olarak adlandırılan piksellerden oluşan kümelere dayanarak gerçekleştirir [18]. $A_1, A_2, \ldots, A_n$ gibi $n$ tane kümeye gruplandırılan tohumlarla başlanır. Bazı durumlarda tohum kümeleri tek piksellerden de oluşabilir. Tohumların seçimi bölütleme yapılacak görüntünün doğasına ve görüntüde nelerin ilgi alanına girdiği, nelerin önemsiz olduğuna bağlıdır. Tohumlar belirlendikten sonra tohum noktalarından başlanarak, tohum noktalarına komşu olan ve tohum noktalarına benzer özellik gösteren piksellerin tohumlara eklenmesiyle bölgeler büyütülür.

S.R.G algoritması şu şekilde özetlenir: $A_1, A_2, \ldots, A_n$ kümelerinin başlangıç durumu olan tohumlardan başlayarak tümevarım şeklinde gelişir [18]. Algoritmanın her adımında piksel veya pikseller yukarıdaki kümelerden bir tanesine eklenir [18].

Algoritmanın herhangi bir adımına bakıldığı zaman, ilk olarak $A_1, A_2, \ldots, A_n$ bölgelerinden herhangi birinin komşuluğunda olan ama hiçbir bölgeye dahil olmayan piksellerin oluşturduğu $K$ kümesi bulunur,

$$K = \left\{ x \notin \bigcup_{i=1}^{n} A_i \mid N(x) \cap (\bigcup_{i=1}^{n} A_i \neq \varnothing) \right\} \quad (3.6)$$

$N(x)$, $x$'in komşularının kümesini belirtir. Başka bir deyişle $N_8(x)$, $x$ pikselinin 8-komşuluğunda piksellerin oluşturduğu bir kümedir. Eğer bir $x \in K$ ise, $x$ pikseli tek bir bölgenin sınırında veya birkaç bölgenin sınırında olabilir.

Birinci durum için, herhangi bir $x$ pikselinin sınırında olduğu tek bir bölge için $i$ dizini tanımlanırsa, $1 \leq i(x) \leq n$ öyle ki

$$N_8(x) \cap A_{i(x)} \neq \varnothing \quad (3.7)$$

ve $x$'in sınırında olduğu bölgeden ne kadar farklı olduğunu ölçen bir benzerlik metriği $\delta(x)$ tanımlanır. Benzerlik metriği çok değişik şekilde ifade edilebilir bunlardan en temeli Eşitlik 3.8'de gösterilmiştir.

$$\delta_i(x) = \left| g(x) - ortalama_{y \in A_i(x)}[g(y)] \right| \quad (3.8)$$

Benzerlik metriğindeki $g(x)$, $x$ piksellinin gri seviyesini gösterir. Benzerlik metriğindeki ikinci terim de $A_1, A_2, \ldots, A_n$ kümelerinin o anki adımındaki aritmetik ortalama değerleridir.

Benzerlik metriğindeki ikinci terim değiştirilerek değişik yapıda benzerlik metrikleri elde edilir. Başka bir benzerlik metriği Eşitlik 3.9'da gösterilmiştir.

$$\delta_i(x) = \left| g(x) - En\ Büyük_{y \in A_i(x)}[g(y)] \right| \quad (3.9)$$

Yukarıdaki benzerlik metriğindeki ikinci terim ise $A_1, A_2, \ldots, A_n$ kümelerinin başlangıç durumundaki yani tohum piksellerini içerdiği durumdaki her kümede en büyük sayıya sahip piksellerin gri seviye değerleridir. Bu değer tohum bölgeleri

(pikselleri) belirlendikten sonra bir kere hesaplanır ve her adımda tekrar hesaplama yapmayı gerektirmez, bundan dolayı da hesaplamaya kolaylık getirir.

Problemin koşullarına göre seçilen ve hesaplanan benzerlik metriği kullanılarak $x$ pikselinin komşuluğunda olduğu bölgeye atanıp atanamayacağı kontrol edilir ve uygunsa yani benzerlik metriği istenilen değerde veya değer aralığında ise $x$ pikseli komşu olduğu bölgeye atanır.

İkinci durum için yani $K$ kümesindeki herhangi bir $x$ pikselinin sınırında olduğu birden fazla bölge olduğu zaman ki durumda aşağıdaki iki işlemden biri uygulanır:

1. $x$ pikselinin komşu olduğu her bölge için benzerlik metriği hesaplanır ve komşu olduğu bütün bölgelere atanabilme koşulunu sağlıyorsa, benzerlik metriği $\delta_i(x)$ en küçük olan bölge $A_i$'ye atanır.

2. $x$ pikseli ayrı bir sınır bölgesine atanır.

$K$ kümesinin bütün elemanı olan pikseller için işlem yapılması bir adımı bitirir. Daha fazla büyütme işlemi gerçekleşmediği durumda yani ardı ardına gelen iki adımda ki $K$ kümeleri arasında fark olmadığı takdirde de büyütme algoritması sonlanır.

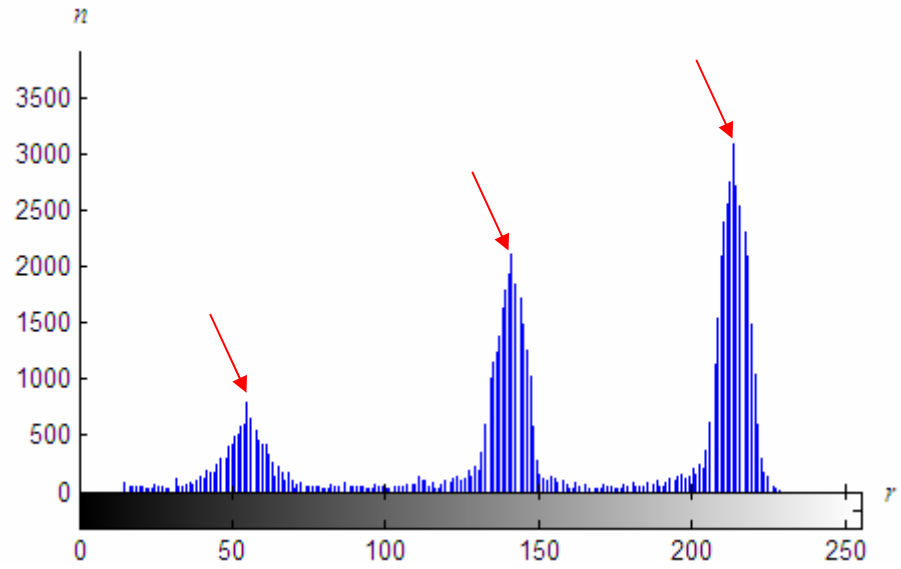
S.R.G yönteminin girdilerinden biri olan tohumların seçimi algoritmanın performansında çok önemli oynar. Tohumların seçimi bir işletici tarafından girilerek yapılabilir ki bu durum bölge büyütme işleminin yarı otomatik olmasına yol açar. Tohumlar başka bir görüntü işleme tekniğinin sonucunda bulunarak S.R.G işlemi tamamen otomatikleştirilebilir.

Tohumlar, bölgelerin istatistiksel özelliklerini gösteren tipik alanlardan seçilmelidir. Tohumların bölge dışı bir yerden seçilmeleri sağlıksız bölütlemeye yol açar. Her bölgeye ait piksellerin gri seviye değerleri kendi içinde benzer özellik gösterdiği göz önüne alınırsa, görüntü histogramı analiz edilerek tohumların

seçimi otomatik olarak gerçekleştirilebilir [21]. Görüntü histogramındaki tepe değerlerine denk gri seviyeye sahip pikseller tohum olarak seçilebilir [21].

Tohumların konumu gibi tohumların sayısı da S.R.G algoritmasında önemli bir parametredir. Bölgeler tohumlardan büyüdüğü için, çok fazla tohum seçilirse, istenmeyen yapay bölgeler bulunur. Böyle bir durumda geometri tabanlı algoritmalar veya operatör yardımlı olarak bölgelerin birleştirilmesi veya yapay bölgelerin atılması gerekir. Çok az sayıda tohum seçilmesi durumunda ise, S.R.G algoritması sonucunda ayrılması gereken bölgeler birbirinden ayrılmayabilir. Böyle bir durumda da bulunan alt bölgelere tekrar bölütleme uygulamak gerekir.

Şekil 3-7'deki örnek histogramda gri seviyelerin üç ayrı değer etrafında gruplandığı görülmektedir, bu tip bir histograma sahip bir görüntüde işaretli tepe değerleri ayrı üç bölgenin tohum pikselleri olarak belirlenebilir.




**Şekil 3-7** Örnek bir histogram

S.R.G algoritmasının başka bir girdisi de her bölge için ayrı ayrı hesaplanan benzerlik metriklerinin değer aralıklarının neler olduğudur, yani benzerlik kriteridir. Benzerlik kriteri problemin yani görüntünün türüne göre değişim gösterir. Tohumların seçiminde olduğu gibi, benzerlik kriteri de başka görüntü işleme teknikleri sonucunda bulunabilir.

### 3.1.3. Harekete dayalı bölütleme

Hareket, insanlar tarafından nesneleri arka plandan ve gereksiz ayrıntıdan ayırmak için kullanılan çok güçlü bir işarettir [9].

Birbirini izleyen sıralı görüntülerde, hareket eden nesneyi bulmak bilgisayar görmesinin en önemli konularından bir tanesidir [22]. Hareket eden nesneyi bulmadaki en büyük problemde o nesneyi görüntüden bölütlemektir. Harekete dayalı bölütleme tekniklerinin çoğu, birbirini izleyen sıralı görüntülerdeki farkı bulmaya dayanır.

Görüntü işleme sistemlerinde nesnelerin hareketleri, hareket eden nesnenin bulunduğu ortamdan $t_i$ ve $t_j$ zamanlarında sabit kamera ile alınan $f(x,y,t_i)$ ve $f(x,y,t_j)$ gibi iki birbirini izleyen sıralı görüntü piksel piksel kıyaslanarak algılanır [9]. Farklı zamanlarda aynı ortamdan sabit kamera ile elde edilmiş görüntüleri kıyaslamanın en temel yolu fark görüntüsünü elde etmektir. Hareket eden nesnenin olmadığı sadece sabit elemanların olduğu görüntüyü referans görüntüsü kabul edip hareketli nesnenin olduğu görüntü referans görüntüsünden çıkarılarak sabit elemanların iptal edildiği, hareketli nesnelerin belirgin olduğu fark görüntüsü bulunur.

$t_i$ ve $t_j$ gibi iki farklı zamanda elde edilmiş görüntülerden bulunan fark görüntüsünün ifadesi Eşitlik 3.10'de gösterilmiştir.

$$d_{ij}(x,y) = \begin{cases} 255 & eğer \left| f(x,y,t_i) - f(x,y,t_j) \right| > T \\ 0 & diğer\ durumlarda \end{cases} \qquad (3.10)$$

Fark görüntüsünün ifadesindeki $T$ değeri eşik değerini ifade eder. Referans görüntüsü ve çıkarılan görüntünün piksellerinin gri seviyeleri arasında eşik değeri ile belirlenen hatırı sayılır derecede fark var olduğu durumlarda $d_{ij}(x,y)$ 255 değerini alır ve 255 değerini alan pikseller hareketli nesne pikselleri olarak değerlendirilir.

Eşik değeri, hareket algılama uygulamalarında çok kritiktir [23]. Yanlış seçilen eşik değeri uygulamalarda hatalı sonuçlar verebilir. Çok düşük seçilen eşik değeri,

fark görüntüsüne yapay değişimler ekler, çok yüksel seçilen eşik değeri önemli değişimlerin fark görüntüsünde belirgin olmasını önler [23]. Doğru eşik değeri, ortama, değişken kamera seviyesine ve aydınlatma gibi görüntüleme koşullarına dayanır [23]. Buda gösterir ki, eşik değeri, görüntü içeriğine göre dinamik olarak eşik değeri bulan yöntemler kullanılarak bulunmalıdır ve deneysel olarak seçilen eşik değeri otomatik görüntü sistemleri için uygun değildir.

Harekete dayalı bölütlemede her zaman referans görüntüsü yani hareketli nesnenin olmadığı sadece hareketsiz elemanların olduğu görüntüyü elde etmek mümkün olmayabilir. Bu durumda hareketli nesnenin olduğu iki görüntü kullanılarak fark görüntüsü bulunur. Referans görüntü elde edilemeyen problemlerde, fark görüntüsünden hareketli nesnenin harekete başladığı ve bitirdiği noktaları bulmak mümkündür. Bu tür problemlerde Eşitlik 3.10'deki fark görüntüsünün bulunuşuyla paralel fakat daha farklı bir yol izlemek gerekir. Hareketli nesneyi görüntüden bölütlemek için ilk olarak iki görüntü birbirinden çıkarılır ve Eşitlik 3.11'deki $g_{ij}(x,y)$ görüntüsü bulunur.

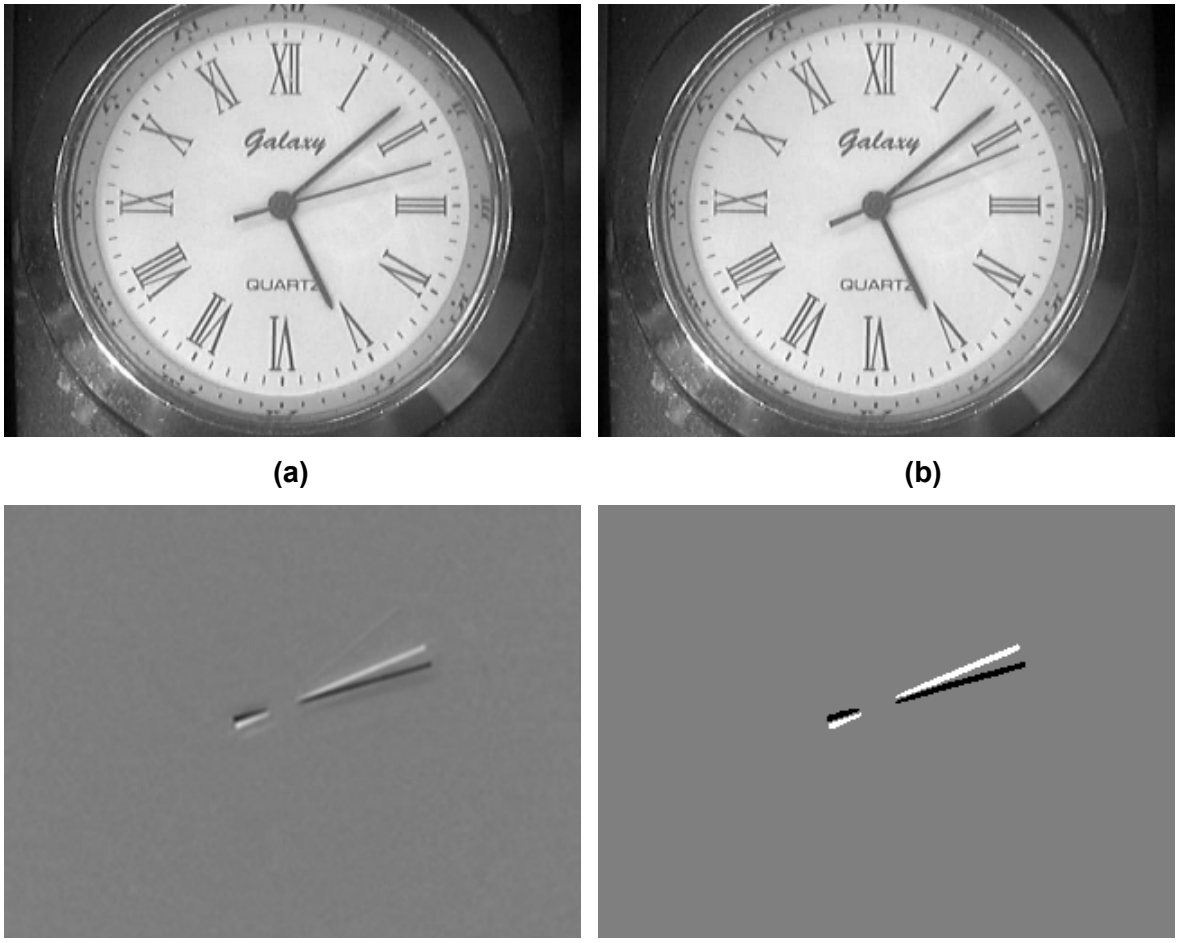
$$g_{ij}(x,y) = f(x,y,t_i) - f(x,y,t_j) \qquad (3.11)$$

Birebir piksel değerlerinin birbirinden çıkarılmasından dolayı bulunan $g_{ij}(x,y)$ görüntüsünün gri seviye değer aralığı $-255$ ile $255$ arasına kayar. Fark görüntüsünün gri seviye değerlerinin 0 ile 255 arasında olmasını istediğimiz için $g_{ij}(x,y)$ görüntüsünde ölçekleme yapmak gerekir. Yapılacak ölçekleme işleminin 0 ile 255 arasını tam olarak ve doğru şekilde kapsaması gerekir. Bu şekilde bir ölçekleme algoritması aşağıdaki 2 adımdan oluşur.

1. **Adım:** $g_{ij}(x,y)$ görüntüsünün en küçük gri seviye değerinin eksilisi, kendisinin bütün pikselleri ile toplanır. Bu adım sonucunda $g_{ij}(x,y)$ görüntüsünün en küçük piksel değeri sıfır olur.

2. **Adım:** 1. adım sonucunda bulunan $g_{ij}(x,y)$ görüntüsünün bütün pikselleri $255/Maks$ değeri ile çarpılarak piksel aralığı tam olarak $[0,255]$ arasını kapsayacak şekilde değiştirilir ve fark görüntüsü bulunur. $Maks$ değeri 1. adım sonunda bulunan $g_{ij}(x,y)$ görüntüsünün en büyük gri seviye değeridir.

Yukarıdaki gibi ölçekleme yapılan bir fark görüntüsünde nesnenin hem harekete başladığı hem de bitirdiği noktalar sırasıyla koyu ve açık gri seviyelere denk gelir. Hareketsiz elemanların olduğu noktalar ise gri seviye aralığının ortalarındaki seviyelere denk gelir. Bu tür bir fark görüntüsünde de uygulamanın amacına göre dinamik olarak eşikleme yapılarak hareketli nesne görüntüden bölütlenir. Nesnenin hem harekete başladığı noktalar hem de bitirdiği noktaları bulmak için iki ayrı eşik değeri uygulanabilir. Şekil 3-8(a)'da ve Şekil 3-8(b)'de sırasıyla bir saniye fark ara ile edinilmiş iki analog saat görüntüsü vardır. İki görüntü arasındaki fark saniye kollarının konumudur. Saniye kolu görüntüden bölütlenmek istendiğinde iki görüntünün Şekil 3-8(c)'deki fark görüntüsü elde edilir ve uygun eşikleme tekniği kullanılarak Şekil 3-8(d)'deki saniye kolunu harekete başladığı noktaları en yüksek, hareketi bitirdiği noktaları en düşük ve hareket olmayan noktalarında 127 gri seviye değeri ile gösterildiği görüntü elde edilir.




**(a)** **(b)**

**(c)** **(d)**

**Şekil 3-8 (a)** $t_i$ anında kamera ile alınan analog saat görüntüsü $f(x,y,t_i)$

**(b)** $t_j$ anında kamera ile alınan analog saat görüntüsü $f(x,y,t_j)$

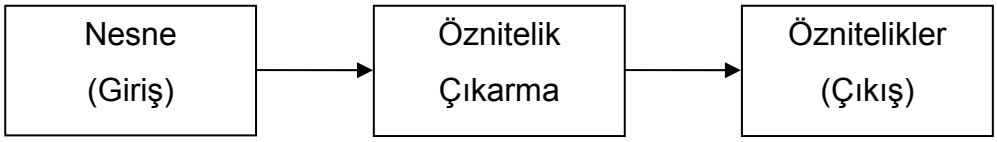
**(c)** $g_{ij}(x,y) = f(x,y,t_i) - f(x,y,t_j)$ görüntüsünün ölçeklenmiş hali

**(d)** Şekil 3-8(c)'deki görüntünün eşikleme uygulanmış hali

## 3.2. Öznitelik Çıkarma

Öznitelik çıkarma, görüntü bölütleme işlemi sonucunda arkaplandan bölütlenen, benzer veya farklı karakteristiğe sahip nesneleri (bölgeleri), birbirlerinden ayıran nitelikleri, karakteristikleri ve özellikleri bulma işlemidir.

Şekil 3-9'deki temel öznitelik çıkarma sistemi görülmektedir. Öznitelik çıkarma işleminin girişi görüntüden bölütlenmiş nesnelerin işlenmemiş sınır pikselleri veya nesnenin bölge piksellerdir, çıkışı ise görüntüden bölütlenmiş nesneden çıkarılan özniteliklerdir.




**Şekil 3-9** Öznitelik çıkarma sistemi

Öznitelik çıkarma işleminin temel hedefi, işlenmemiş pikselleri, nesne tanıma problemini çözmede gereken bütün uygun ve ayırt edici bilgileri içeren küçük öznitelik kümelerine dönüştürmektir [24].

Görüntüden bölütlenen nesneden çıkarılacak öznitelikleri seçme isteğe bağlı olarak yapılan bir işlem olup nesne tanıma analizinde bir sonraki adım olan örüntü tanıma işleminin performansını doğrudan etkiler. Örüntü tanıma işleminin performansını arttırmada en belirleyici özniteliklerin çıkarılması önemlidir. Belirleyici özniteliklerin seçimi, görüntüden bölütlenen nesnelere göre belirlenir, deneyim ve sezgisellik gerektirir. Gereksiz veya fazladan çıkarılan karmaşık öznitelikler, nesne tanıma analizinin hem öznitelik çıkarma adımında hem de örüntü tanıma adımındaki işlem süresini arttırır. Uygun ve belirleyici özniteliklerin çıkarılmaması da örüntü tanıma adımının performansını düşürür ve nesne tanıma analizinin başarısını kötü yönde etkiler.

Çıkarılacak özniteliklerin seçimindeki dikkat edilmesi gereken üç durum aşağıdaki gibi maddeleşebilir:

1. Seçilen öznitelik veya öznitelikler, nesnenin görüntü içindeki boyutunun değişimine olabildiğince duyarsız olmalıdır.

2. Seçilen öznitelik veya öznitelikler, nesnenin görüntü içindeki konumunun değişimine olabildiğince duyarsız olmalıdır.

3. Seçilen öznitelik veya öznitelikler, nesnenin görüntü içindeki dönmesine olabildiğince duyarsız olmalıdır.

Görüntüden bölütlenmiş nesneden çıkarabilecek öznitelikler iki başlık altında gruplandırılır [9].

1. Nesne bölgesinin sınır karakteristiklerine göre çıkarılabilecek öznitelikler.

2. Nesne bölgesinin bölge karakteristiklerine (nesne bölgesini oluşturan piksellere) göre çıkarılabilecek öznitelikler.

Sınır karakteristikleri tanınmak istenen nesnenin şeklini de belirler ki sınır karakteristiklerine göre öznitelik çıkarma işlemini basitçe tanınmak istenen nesnenin şeklinin algılanması olarak düşünebiliriz, yani sınır karakteristiğe göre öznitelik çıkarma işlemini nesnenin şekil karakteristiği önemli ise yani nesneyi çok benzer hatta aynı bölge karakteristiği gösterip farklı sınır karakteristiği gösteren nesnelerden ayırmak istediğimiz zaman seçebiliriz. Tam tersi olduğu durumlarda ise bölge karakteristiğine göre öznitelik çıkarma seçilebilir. Nesnenin özelliğine göre, hem sınır hem de bölge karakteristiğe göre öznitelik çıkarma işlemi de yapılabilir.

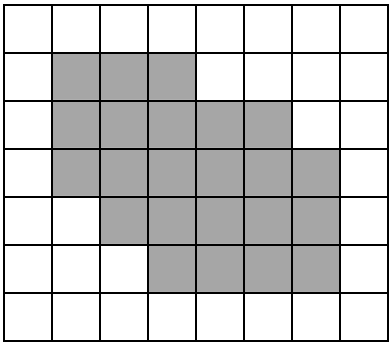
Aşağıda, literatürde geçen ve bazıları bu çalışmada kullanılan, görüntüden bölütlenmiş nesneleri içeren ikili görüntülerden çıkarılabilecek özniteliklerin ve çıkarma yöntemlerinin üzerinde durulmaktadır.

### 3.2.1. Bölge karakteristiklerine göre çıkarılabilecek öznitelikler

Bölge karakteristiği göre çıkarılacak özniteliklerin en temelleri basit, topolojik ve doku özniteliklerdir.

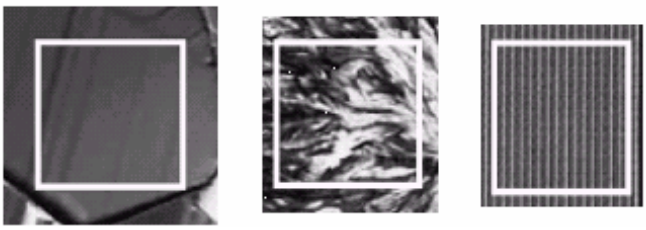
Görüntüden bölütlenen nesnenin alanı basit özniteliktir. Alan bölgeyi oluşturan piksellerin sayısı olarak tanımlanırsa, Şekil 3-10'daki nesnenin alanı *23* piksel olarak bulunur. Alan özniteliği dikkat edilirse nesnenin görüntü içindeki konumuna ve dönmesine duyarsızdır fakat görüntü içindeki nesnenin boyut değişimine duyarlıdır. Alan özniteliği, aynı görüntü içindeki alanları birbirinden farklı iki nesneyi birbirinden ayırt etmek için kullanılırsa, görüntü içinde iki nesnenin de

birbirleriyle orantılı olarak boyutlarının değişmesi durumunda nesnelerin boyut değişimlerine de duyarsız olmaktadır.



**Şekil 3-10** Nesne bölgesi

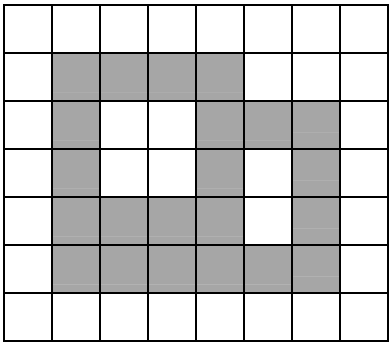
Doku öznitelikleri, nesnenin dokusunu tanımlayan özniteliklerdir. Doku öznitelikleri, nesnenin dokusunun düzgünlüğünü, kabalığını ve düzenliliğinin değerleridir ki bu değerleri istatistiksel, yapısal veya spektral yaklaşımlar yapılarak elde edilir. Şekil 3-11'da doku özellikleri birbirinden farklı üç ayrı nesneden alınmış görüntülerin parçaları vardır. Bu üç ayrı doku özelliğine sahip nesne tanınmak veya birbirinden ayrılmak istendiğinde doku öznitelikleri ayırt edici olarak kullanılabilir.



**(a)** **(b)** **(c)**

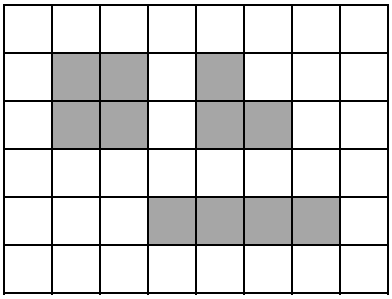
**Şekil 3-11** **(a)** Düzgün dokulu nesne [9]
**(b)** Kaba dokulu nesne [9]
**(c)** Düzenli dokulu nesne [9]

Topolojik öznitelikler, basitçe nesnede bölünme veya başka bir nesneyle birleşme olduğu zaman dışında etkilenmeyen öznitelikler olarak tanımlanabilir [9]. Nesnedeki deliklerin sayısı topolojik özniteliktir. Şekil 3-12'de iki delikli bir bölge gözükmektedir.



**Şekil 3-12** İki delikli bölge

Nesnedeki bağlı bileşenlerin sayısı da başka bir topolojik özelliktir. Bağlı bileşen kavramı Bölüm 2.3.3'de açıklanmıştır. Şekil 3-13'de üç bağlı bileşenli bir bölge gözükmektedir.



**Şekil 3-13** Üç bağlı bileşenli bölge

Nesnedeki delik sayısı $D$ ve bağlı bileşen sayısı $B$ kullanılarak Eşitlik 3.12'de tanımlanan Euler numarası $E$'de başka bir topolojik özniteliktir [9]. Şekil 3-12'de bölgenin Euler numarası -1, Şekil 3-13'deki bölgenin Euler numarası 3'dür.

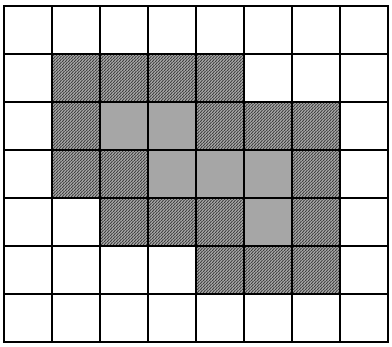
$$E = B - D \qquad (3.12)$$

### 3.2.2. Sınır karakteristiklerine göre çıkarılabilecek öznitelikler

Sınır karakteristiklerine göre çıkarılacak özniteliklerin en temelleri basit öznitelikler ve imzadır [9].

Görüntüden bölütlenen nesnenin sınırının uzunluğu basit bir özniteliktir. Sınır uzunluğu kabaca, sınırı oluşturan piksellerin sayısı olarak tanımlanırsa,

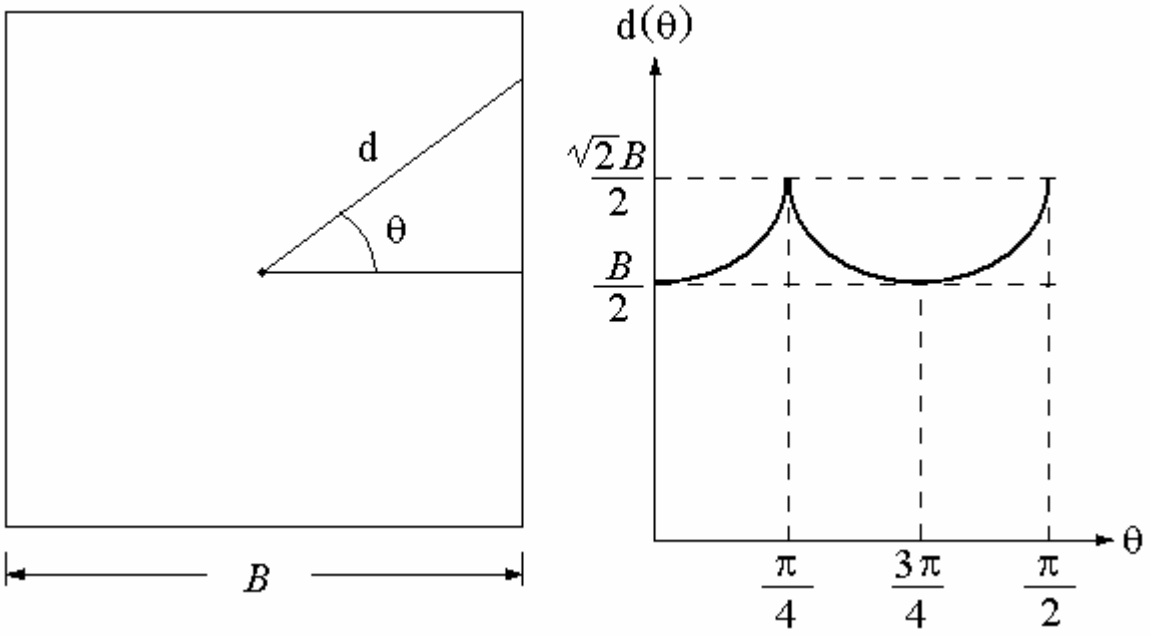
Şekil 3-14'deki nesnenin sınır uzunluğu 18 piksel olarak bulunur. Sınır uzunluğu da alan özniteliği gibi nesnenin görüntü içindeki konumuna ve dönmesine duyarsızdır fakat görüntü içindeki nesnenin boyut değişimine duyarlıdır.



**Şekil 3-14** Nesne bölgesi ve sınırları

### 3.2.2.1. İmza

İmza, nesne sınırının bir boyutlu gösterimidir ve birçok değişik yoldan oluşturulabilir [9]. Basit bir imza gösterimi Şekil 3-15(b)'de verilmektedir. Bu imza gösteriminde, Şekil 3-15(a)'daki nesnenin sınırındaki piksellerin nesnenin merkez noktasına olan öklidyen uzaklığı, iki noktayı birleştiren doğru parçasının eğiminin fonksiyonu olarak çizilmiştir. Burada bahsedilen eğim doğru parçasının yatay eksene göre açısıdır [25].




**Şekil 3-15 (a)** Örnek bir nesne sınırı **(b)** Nesne sınırının imza gösterimi

### 3.2.3. İnceltme

Düzgün bir bölgenin yapısal şeklini ifade etmenin en bilinen yöntemi, bölgenin, inceltme algoritmalarıyla iskeletini çıkarmaktır [9].

İskelet çıkarma, nesne tanıma analizlerinde sıkça kullanılan bir yöntemdir. Bir görüntüden bölütlenmiş nesnelerin temel yapısını belirlemek için bu nesneler, bir piksel kalınlıkta kalacak şekilde inceltilerek iskeletleri çıkarılabilir. Nesne tanıma analizinde, iskelet çıkarmanın amacı, görüntüden bölütlenmiş nesneden en belirleyici öznitelikleri çıkarmaktır. Problemin türüne göre görüntüden bölütlenmiş nesnenin iskeletinden çıkarılan öznitelikler, görüntüden bölütlenmiş nesneden direk olarak çıkarılan özniteliklerden daha belirleyici olabilir.

İnceltme algoritmaları disk şeklindeki bölgelerde iyi sonuç vermezler ve ideal durumda, nesnenin inceltilerek elde edilen iskeletinin, nesnenin ana ekseni üzerinde olması beklenir.

Birçok inceltme algoritması vardır. İnceltme algoritmalarında amaç, hesaplamalardaki verimliliği artırarak ideale yakın iskelet çıkarmaktır. İnceltme algoritmalarındaki temel yaklaşım, nesnenin sınırlarındaki piksellerin silindiği ardışık işlemler şeklinde olmasıdır. Pikseller silinirken dikkat edilmesi gereken üç kısıtlama vardır:

1. Nesnelerin uç noktaları çıkarılmamalıdır.

2. Nesne bölgesinin pikselleri arasındaki bağlılık bozulmamalıdır.

3. Nesne bölgesinde gereksiz aşınma olmamalıdır.

Bu tez çalışmasında kullanılan inceltme algoritması özetle şu şekildedir: İnceltilecek bölge pikselleri 255 gri seviye değerine sahip olduğu ve arkaplan pikselleri de 0 gri seviye değerine sahip olduğu varsayılırsa, inceltme algoritması uygulanacak görüntüdeki 255 ve 0 değerleri 1 ve 0 olarak düşünülebilir. Buna göre sınır pikselleri de gri seviye değeri 255 (1) olan ve 8 komşuluğundaki piksellerden en az birinin gri seviye değerinin 0 olduğu piksellerdir.

İnceltme algoritmasındaki 8-komşuluktaki piksellerin sıralanması Şekil 3-16'da gösterilmiştir.

| | | |
|---|---|---|
| $p_9$ | $p_2$ | $p_3$ |
| $p_8$ | $p_1$ | $p_4$ |
| $p_7$ | $p_6$ | $p_5$ |

**Şekil 3-16** İnceltme algoritması için 8-komşuluktaki piksellerin sıralanışı [9]

İnceltme algoritması sınır piksellerine uygulanan iki temel adımdan oluşur [9]. Algoritmanın birinci adımında aşağıdaki koşulları sağlayan sınır pikselleri silinmek üzere işaretlenir.

**(a)** $2 \leq N(p_1) \leq 6$

**(b)** $G(p_1) = 1$

**(c)** $p_2 \cdot p_4 \cdot p_6 = 0$

**(d)** $p_4 \cdot p_6 \cdot p_8 = 0$

Belirtilen koşullardaki $N(p_1)$, Eşitlik 3.13'de görüldüğü gibi $p_1$'in sıfıra eşit olmayan komşularının sayısıdır.

$$N(p_1) = p_2 + p_3 + \cdots + p_8 + p_9 \qquad (3.13)$$

$G(p_1)$ ise $p_2, p_3, \ldots, p_8, p_9, p_2$ piksel sıralamasında 0-1 geçişlerinin sayısıdır. Şekil 3-17'de $p_1$ pikseli için $G(p_1) = 2$ ve $N(p_1) = 3$ 'tür.

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | $p_1$ | 1 |
| 1 | 1 | 0 |

**Şekil 3-17** İnceltme algoritmasındaki (a) ve (b) koşullarının gösterimi

Bütün görüntü taranıp sınır piksellerinden bütün koşulları sağlayanlar işaretlendikten sonra görüntü tekrar taranır ve işaretlenen sınır pikselleri silinir. Algoritmadaki ilk adım silinme işlemi ile tamamlanmış olur. Dikkat edilmesi gereken durum, koşulları sağlayan piksellerin silinme işleminin, görüntüdeki bütün sınır pikselleri üzerinde koşullar kontrol edildikten sonra gerçekleştirilmesidir.

Algoritmanın ikinci adımında birinci adımdaki gibi sınır pikselleri silinmek üzere işaretlenir. İkinci adımda piksellerin işaretlenme koşullarından a ve b koşulları ilk adımla aynıdır. Bu adımda c ve d koşulları

**(c')** $p_2 \cdot p_4 \cdot p_8 = 0$

**(d')** $p_2 \cdot p_6 \cdot p_8 = 0$

biçiminde değiştirilir. Birinci adımda olduğu gibi görüntü taranıp sınır piksellerinden bütün koşulları sağlayanlar işaretlendikten sonra görüntü tekrar taranır ve işaretlenen sınır pikselleri silinir.

Özetle, inceltme algoritmasının bir dürümündeki sınır piksellerine birinci adım uygulanarak silinecek pikseller işaretlenir ve silinir daha sonra kalan sınır piksellerine ikinci adım uygulanarak silinecek pikseller işaretlenir ve silinir. Bu temel yordam, özyinelemeli olarak silinecek hiçbir piksel kalmayana kadar uygulanır ve silinecek piksel kalmadığı zaman algoritma sonlanır ve bölgenin iskeleti ortaya çıkar.

Sınır piksellerinin, nesne bölgesinin iskeletinin bir noktası olup olmayacağını belirleyen koşulların dayandığı gerekçeler şunlardır:

$p_1$ sınır pikselinin 8-komşuluğundaki piksellerden bir veya yedi tanesinin değeri 1 ise (a) birinci koşul ihlal edilmiş olur. Değeri 1 olan sadece bir komşusu olması, $p_1$ pikselinin iskelet çizgisinin en son noktası olduğunu ve silinmemesi gerektiğini gösterir. Değeri 1 olan yedi komşusu olması ise $p_1$ pikselinin silinmesinin bölgede gereksiz aşınma yaratacağını ve silinmemesi gerektiğini gösterir.

İskelet çizgisinin tek piksel kalınlığında olduğu noktalarda (b) ikinci koşul ihlal edilir. Bu koşul ihlal edilerek inceltme işlemi sırasında iskeletin parçalara ayrılması önlenir.

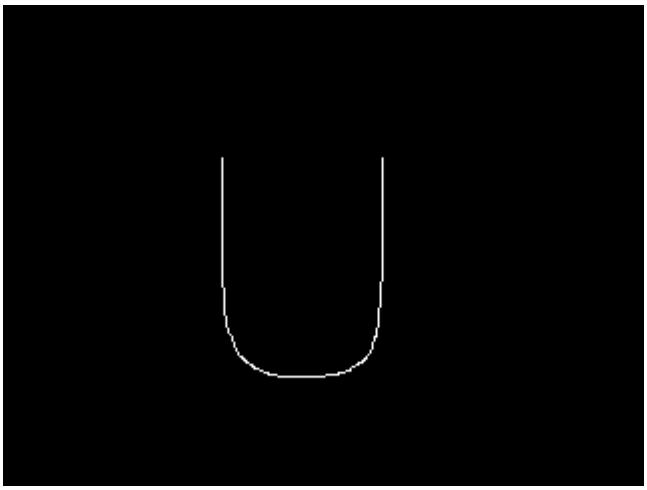
(c) ve (d) koşulları ($p_4$ = 0 veya $p_6$ = 0) veya ($p_2$ = 0 ve $p_8$ = 0) olduğu durumlarda sağlanır. (a) ve (b) koşullarının da birlikte sağlandığı pikseller Şekil 3-16'daki sıralanışa göre, iskeleti elde edilecek bölgenin doğu veya güney veya kuzey batı sınır noktasında olduğu için iskelete dahil değildir ve silinmesi gerekir.

(c') ve (d') koşulları da ($p_2$ = 0 veya $p_8$ = 0) veya ($p_4$ = 0 ve $p_6$ = 0) olduğu durumlarda sağlanır. (a) ve (b) koşullarının da birlikte sağlandığı pikseller Şekil 3-16'daki sıralanışa göre, iskeleti elde edilecek bölgenin kuzey veya batı veya güney doğu sınır noktasında olduğu için iskelete dahil değildir ve silinmesi gerekir.

Bölgenin kuzeydoğu köşe noktalarında bulunan pikseller içinde $p_2$ = 0 ve $p_4$ = 0'dır ve (c), (d), (c') ve (d') koşullarını sağladığı için bu piksellerde iskelete dahil değildir ve silinmesi gerekir. Bu durum bölgenin güneybatı köşe noktalarında bulunan ve $p_6$ = 0 ve $p_8$ = 0 yapan pikseller içinde geçerlidir.

Şekil 3-18(a)'da görülen U harfinin inceltme algoritması ile elde edilmiş iskeleti Şekil 3-18(b)'de görülmektedir. Dikkat edilirse U harfinin iskeletinde fazladan aşınma veya parçalara ayrılma durumu yoktur yani yapısal olarak bir farklılık yoktur.

**(a)**



**(b)**

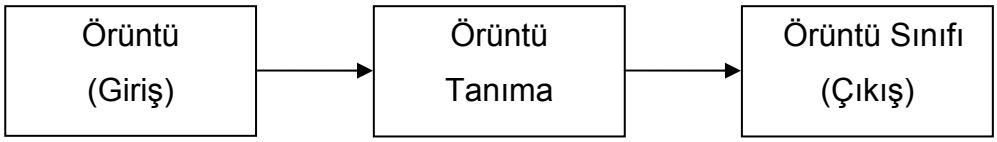
**Şekil 3-18 (a)** U harfi görüntüsü **(b)** U harfinin iskelet görüntüsü

## 3.3. Örüntü Tanıma

Öznitelikler çıkarmadan bir sonraki adım ve nesne tanıma analizinin de son adımı olan örüntü tanıma, görüntüden bölütlenmiş, öznitelikleri çıkarılmış nesneyi, özniteliklerine dayanarak, çeşitli sınıflama yaklaşımlarıyla kendisine en yakın özelliklere sahip sınıfa atayarak nesneyi tanımlar. Örüntü tanıma basitçe, sınıflandırma işlemi olarak görülebilir.

Şekil 3-19'da temel örüntü tanıma sistemi görülmektedir. Örüntü tanıma işleminin girişi nesneden çıkarılan özniteliklerin düzenlenmesi ve bir araya

getirilmesi ile oluşturulan örüntüdür. Örüntü tanıma sistemini çıkışı ise girişteki örüntü ile benzer özelliklere sahip örüntülerin oluşturduğu örüntü sınıfıdır.




**Şekil 3-19** Örüntü Tanıma Sistemi

Örüntü tanıma (sınıflandırma) yöntemleri, yaklaşımına göre üç grupta toplanabilirler [26]:

- İstatistiksel yaklaşım: İstatistiksel yaklaşımda sınıflama algoritmaları istatistiksel analiz üzerine kurulmuştur. Aynı örüntü sınıfına ait örüntüler istatistiksel olarak benzer özellikleri paylaşırlar. Sınıflandırma algoritmalarında, ölçümler ve olasılıklar gibi nitel özniteliklerden oluşturulan örüntülerin sınıflandırılması esas alır.

- Yapısal Yaklaşım: Yapısal yaklaşımlarda sınıflama algoritmaları yapısal analiz üzerine kurulmuştur. Sınıflandırma algoritmalarında nesnelerden çıkarılan sayısal öznitelikler yerine, nesnenin özellikleri arasındaki ilişki ve bağlantılardan çıkarılan yapısal özniteliklerin sınıflandırılması esas alınır.

- Yapay Sinir Ağları Yaklaşımı: Yapay sinir ağları ile örüntü tanıma yaklaşımı temel olarak veri ve kural tabanlı sistemler ile yapılan sınıflandırma olan yapısal bir yaklaşımdır [27]. Yapay sinir ağları yaklaşımı ile tanıma işleminde örüntüler giriş olarak alınarak, doğrusal olmayan hesaplama elemanları ile her biri farklı bir sınıfı gösteren çıkış ayaklarının birinden cevap üretilir.

Örüntü tanıma (sınıflama) yöntemleri, tanımlama işleminden önce tanımlanacak sınıfların bilinip bilinmemesine göre de iki grupta toplanabilir [27]:

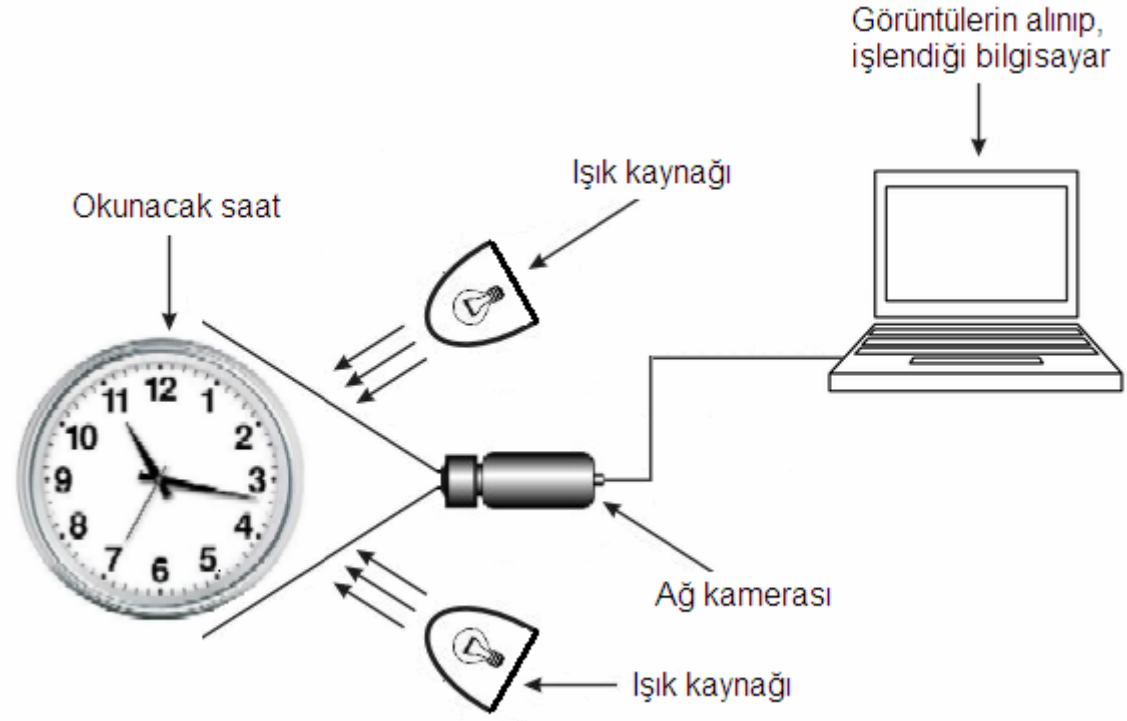
- Örüntü tanımada, tanıma işlemi yapılmadan önce tanımlanacak sınıflar biliniyorsa buna denetlenmiş öğrenme denir. Denetlenmiş öğrenmede, örüntüleri sınıflayacak karar verme mekanizması eğitim setine göre yapılır ve performansın testi için de eğitim setinden farklı olarak test seti kullanılır [26].

- Örüntü tanımada, tanıma işlemi yapılmadan önce tanımlanacak sınıflar bilinmiyorsa buna denetlenmemiş öğrenme denir. Denetlenmemiş öğrenmede, tanıma işlemi yapılmadan önce sınıflar belli olmadığından, tanınmak istenen örüntüler arasında oluşacak gruplanmalar analiz edilir ve sınıfları ortaya çıkaran kümeleme teknikleri kullanılır [26].

Bu tez çalışmasında örüntü tanıma adımı basit olarak uygulanmıştır. Çalışmamızda tanınmak istenen daha doğru bir ifadeyle birbirlerinden ayırt edilmek istenen kolların üçünün de çakışık olma olasılığı dışındaki bütün olasılıklarda kollardan çıkarılan ayırt edici öznitelikleri karşılaştırılarak kollar birbirinden ayrılır. Karşılaştırma işlemi basitçe nicel özniteliklerin birbirinden değer olarak büyük veya küçük olma durumları kontrol edilerek yapılmaktadır.

## 4. SAATİN KAÇ OLDUĞUNUN ANLAŞILMASI

Akrep, yelkovan ve saniye kolu olan bir saat görüntüsünün karşısına yerleştirilen bir video kamerası aracılığı ile görüntü alınarak bilgisayar ortamında görüntü işleme teknikleri uygulanarak çözümlenen görüntüden saatin kaç olduğunun anlaşılması amacıyla gerçekleştirilen saat okuma sistemi Şekil 4-1'de gösterilmiştir. Saatin kaç olduğunun anlaşılması birden fazla farklı saat üzerinde gerçekleştirilmiştir.




**Şekil 4-1** Saat okuma sistemi

Bu çalışmadaki saat okuma sisteminde video kamera olarak Ek1'de teknik özellikleri belirtilen Philips firması tarafından üretilmiş PCVC 840K model ağ kamerası, bilgisayar olarak özellikleri Ek2'de belirtilen HP firması tarafından üretilmiş NC6120 model diz üstü bilgisayar kullanılmıştır. Ağ kamerası ile yakalanan görüntüleri kullanarak saatin kaç olduğunu anlamak ve saatin kaç olduğunu görüntülemek için C# tabanlı program yazılmıştır.

Saat okuma sisteminde diğer bütün görsel tanıma sistemlerinde olduğu gibi bazı ön koşulların belirlenmesine ihtiyaç duyulmuştur. Bunun nedeni, olabilecek algılama yanılgılarını en aza indirmektir. Söz konusu ön koşullar şunlardır:

- Saatin kaç olduğunun anlaşılmasında kullanılacak saatlerdeki akrep, yelkovan ve saniye kolları belirgin ve yakın gri tonda olmalıdır.

- Saatin kaç olduğunun anlaşılmasında kullanılacak saatlerdeki yelkovan ve saniye kolu aynı uzunlukta olmalıdır. Akrep kolu, yelkovan ve saniye kollarından kısa olmalıdır. Akrep ve yelkovan kollarının kalınlıları aynı olmalıdır. Saniye kolu, akrep ve yelkovan kolundan ince olmalıdır.

- Saatin kaç olduğunun anlaşılmasında kullanılacak saatin göstergesinin arka planı, saat kollarının algılanışını engellememelidir.

- Saatin kaç olduğunun anlaşılmasında kullanılacak saatin göstergesi içinde, kollar dışında göstergenin arka planı ile karşıtlık farkı olan rakam, yazı ve simge gibi unsurlar olmalıdır.

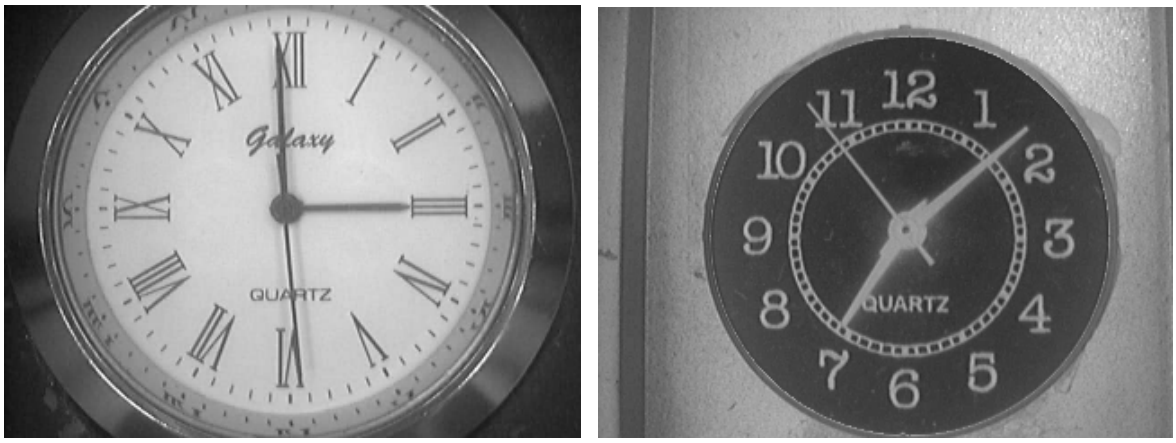
- Saati okumak için alınan görüntülerde saat görüntünün her yerinde olabilir veya kameraya uzaklığı değişebilir ama kollar görüntü içinde kalacak şekilde saatin konumu ayarlanmalıdır. Saat kameradan uzaklaştıkça saat okuma sisteminin başarısı düşmektedir.

- Saatin kaç olduğunun anlaşılmasında kullanılacak saatte 3 ve 9 noktalarının birleştiren sanal doğru görüntünün görüntü düzleminin yatay eksenine paralel olarak ayarlanmalıdır.

- Saat okuma sisteminin çalıştığı ortamdaki aydınlatma düzgün ve yeterli olmalıdır.

- Daha önceden belirtildiği gibi alınan saat görüntülerinin boyutu $240x320$ ve gri seviye değerleri de $[0,255]$ arasındadır.

Bu çalışmada kullanılan saatlerden iki örnek Şekil 4-2(a) ve Şekil 4-2(b)'de görülmektedir. Dikkat edilirse Şekil 4-2(a)'daki saatte kollar arka plana göre daha koyudur yani kolların pikselleri arka plan piksellerinden daha düşük değerlidir. Şekil 4-2(b)'deki saatte durum tam tersidir. Saatin arka planı kollara göre daha koyudur. Her iki saatteki kollar arasındaki kabul edilen boyut farklılıkları da görülmektedir. 3 ve 9 noktalarının birleştiren hayali doğruda görüntünün yatay eksenine paraleldir.




**(a)** **(b)**

**Şekil 4-2 (a)** Arka planı kollara göre daha açık olan örnek saat görüntüsü
**(b)** Arka planı kollara göre daha koyu olan örnek saat görüntüsü

Bölüm 2.1'de ayrıntılı olarak anlatıldığı gibi görüntü cisimlerden yansıyan ışıktan oluşmaktadır. Yansıyan ışık ise aydınlatma ve yansıma katsayılarının çarpımından elde edilir.

Farklı iki gri seviyede olan saatin kolları ve arka planı arasındaki grilik karşıtlığının yeterli olabilmesi için aydınlatma bileşenin de yeterli olması gerekir. Buradaki yeterli kavramı aydınlatmanın ne çok fazla ne de çok az olmasıdır.

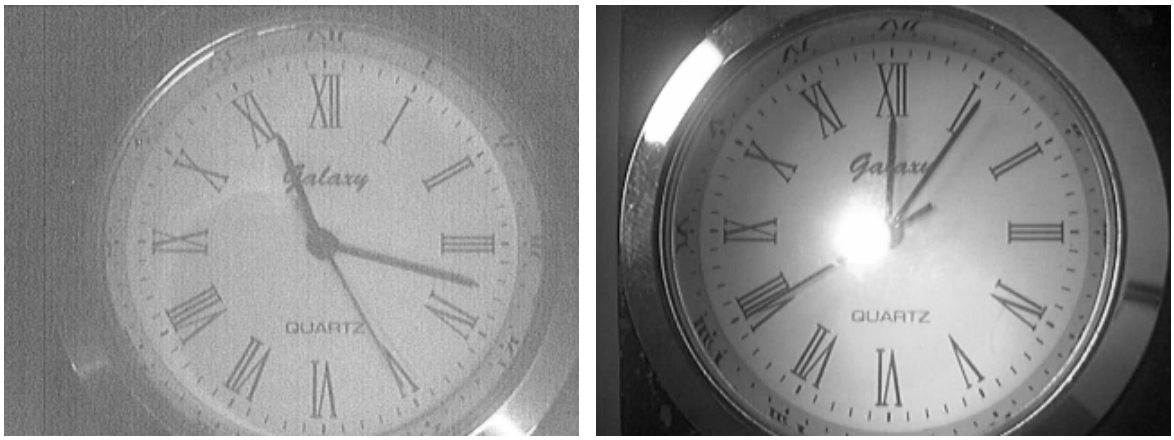
Aydınlatmada sık karşılaşılan sorunlar:

- Saat okuma sisteminin bulunduğu ortamdaki ışık kaynağının şiddetinin düşük olması, aydınlatmanın da gereğinden az olmasına yol açabilir.

- Saat okuma sistemin çevresindeki hareketli veya hareketsiz nesnelerden yansıyan ışıkların saat üstüne düşmesi istenmeyen sonuçlar yaratabilir.

- Saat okuma sisteminin çevresindeki hareketli veya hareketsiz nesnelerin gölgelerinin saat üstüne düşmesi, yine istenmeyen sonuçlar yaratabilir.

Şekil 4-3(a)'da aydınlatmanın düşük olduğu ortamda alınmış saat görüntüsü görülmektedir. Bu görüntüdeki saatin kolları ve arka planı arasındaki fark çok belirgin değildir. Şekil 4-3(b)'de aydınlatmanın düzgün olmadığı bir ortamda alınmış bir saat görüntüsü görülmektedir. Bu görüntüdeki saatin kolları ve arka planı arasındaki fark belirgin olmasına rağmen sistemin yakınındaki bir nesneden yansıyan ışık yüzünden görüntü bozulmuştur.




**(a)** **(b)**

**Şekil 4-3 (a)** Aydınlatmanın düşük olduğu saat görüntüsü
**(b)** Aydınlatmanın düzgün olmadığı saat görüntüsü

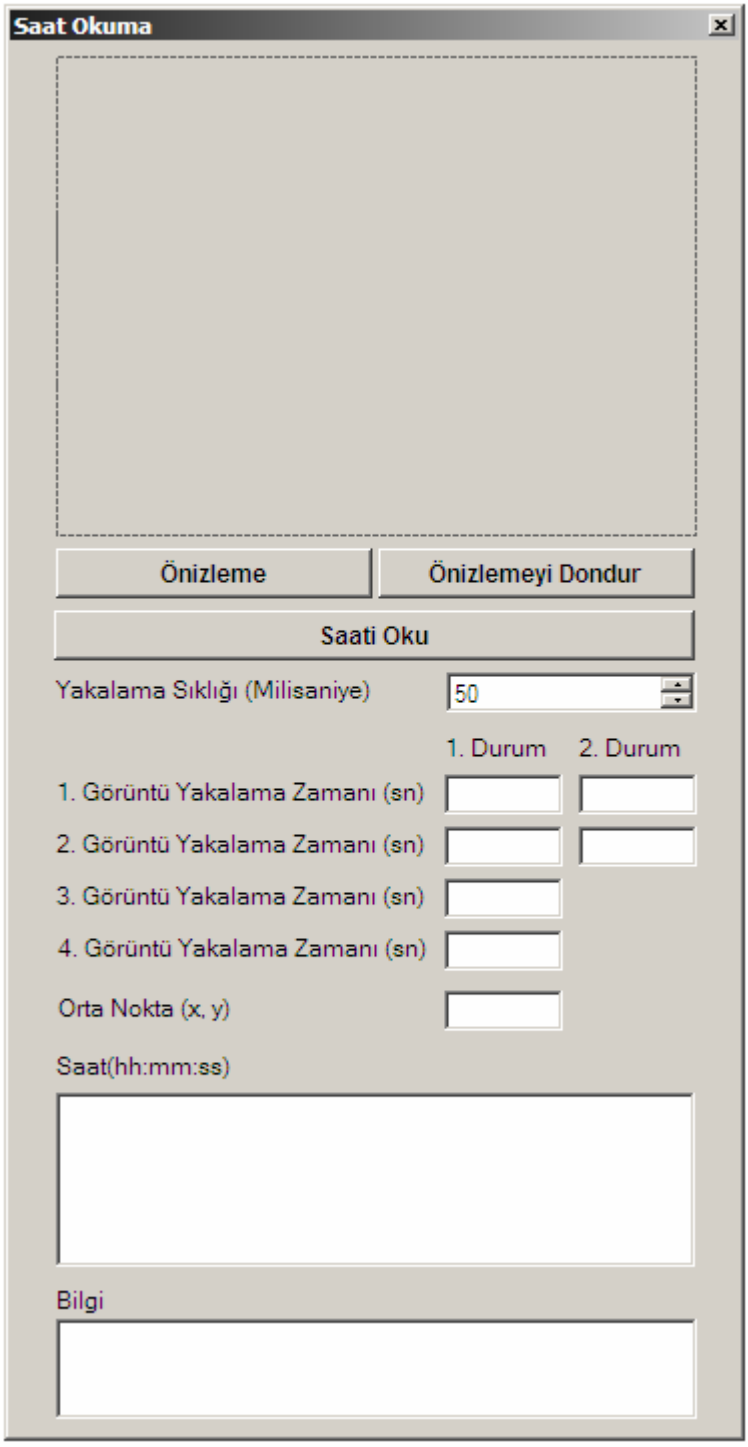
Aydınlatmanın düzgün ve yeterli olmasını sağlayacak, dış ışık kaynaklarına ve çevredeki nesnelerden yansıyan ışığa baskın gelecek LED'li aydınlatma gereçleri sisteme eklenmiştir. Sisteme eklenen bu aydınlatma gereçleri sayesinde saatten yansıyan ışık miktarına etki eden çevresel ışık kaynaklarının etkisi azaltılmıştır. LED'lerin sayısı ve konumu, ışığın düzgün ve yeterli dağılmasının sağlayacak şekilde ayarlanmıştır.

Şekil 4-4'de aydınlatmanın düzgün ve yeterli saat görüntüsü görülmektedir. Bu görüntüdeki saatin kolları ve arka planı arasındaki karşıtlık belirgin ve düzgündür.

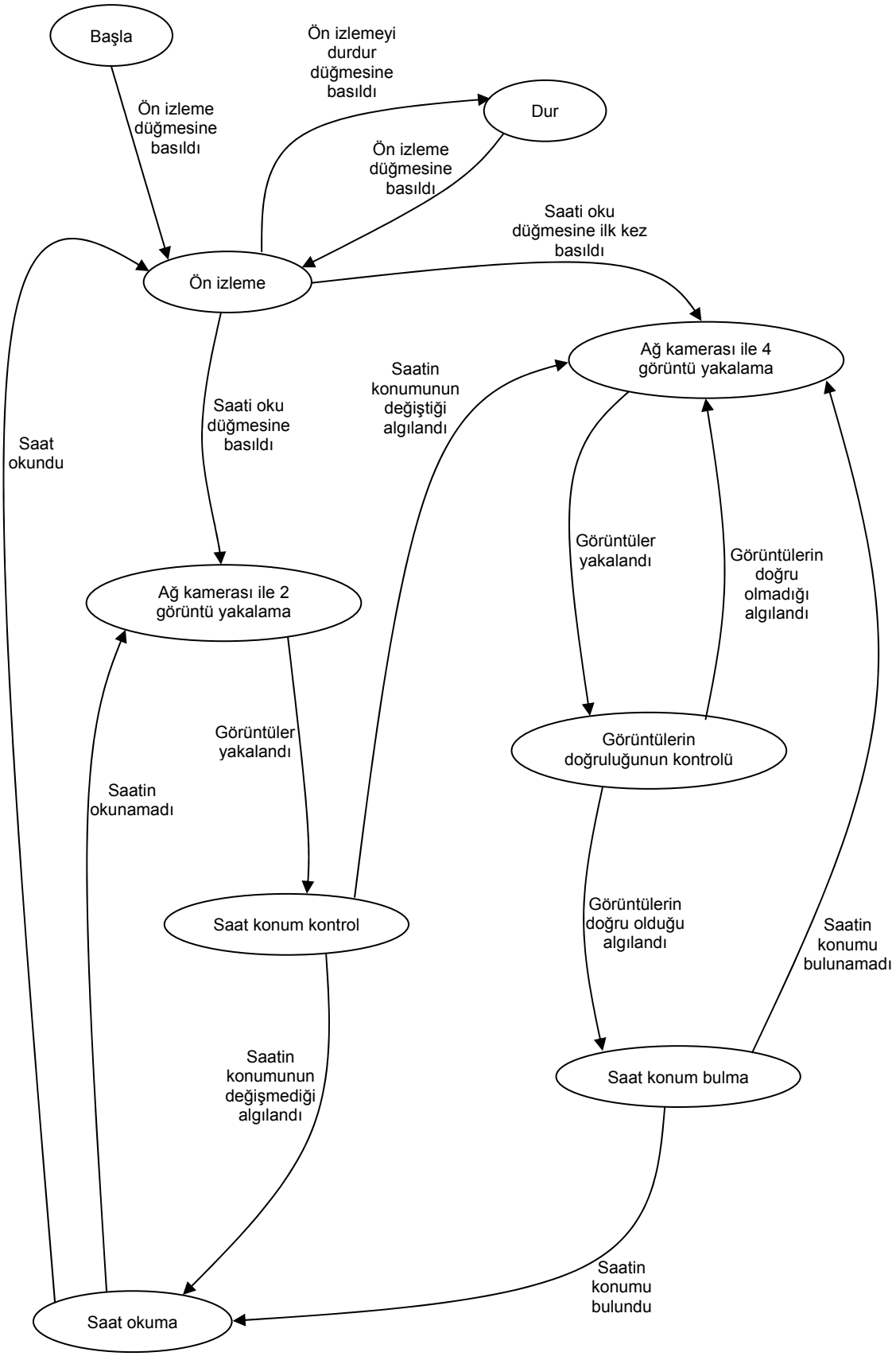
**Şekil 4-4** Aydınlatmanın türdeş ve yeterli olduğu saat görüntüsü

## 4.1. Program Arayüzü ve Genel Algoritması

Ağ kamerası ile yakalanan görüntüleri kullanarak saatin kaç olduğunu anlamak ve saatin kaç olduğunu görüntülemek için yazılan C# tabanlı programın arayüzü Şekil 4-5'de ve veri akış diyagramı Şekil 4-6'de görülmektedir




**Şekil 4-5** Programın arayüzü




**Şekil 4-6** Programın veri akış diyagramı

Arayüz de tuşların üstündeki pencere (görüntü penceresi) ağ kamerası ile elde edilen görüntüler için ayrılmıştır. Görüntü penceresi dışında arayüzde bulunan düğmeler ve yazı pencerelerinin başlıkları arayüzün üstünde bulunmaktadır.

Saatin kaç olduğunun anlaşılması için program çalıştırıldıktan sonra ekrana gelen arayüzde ilk işlem olarak ön izleme yapılması gerekir. Ön izleme işlemi, ön izleme düğmesine basıldığında ağ kamerası yaklaşık 100 milisaniyede bir görüntü alınarak görüntü penceresine konulmasıyla yapılır. Ön izlemeyi durdur düğmesi ağ kamerasının görüntü almasını durdurur ve görüntü penceresinde en son konulan görüntü görünür.

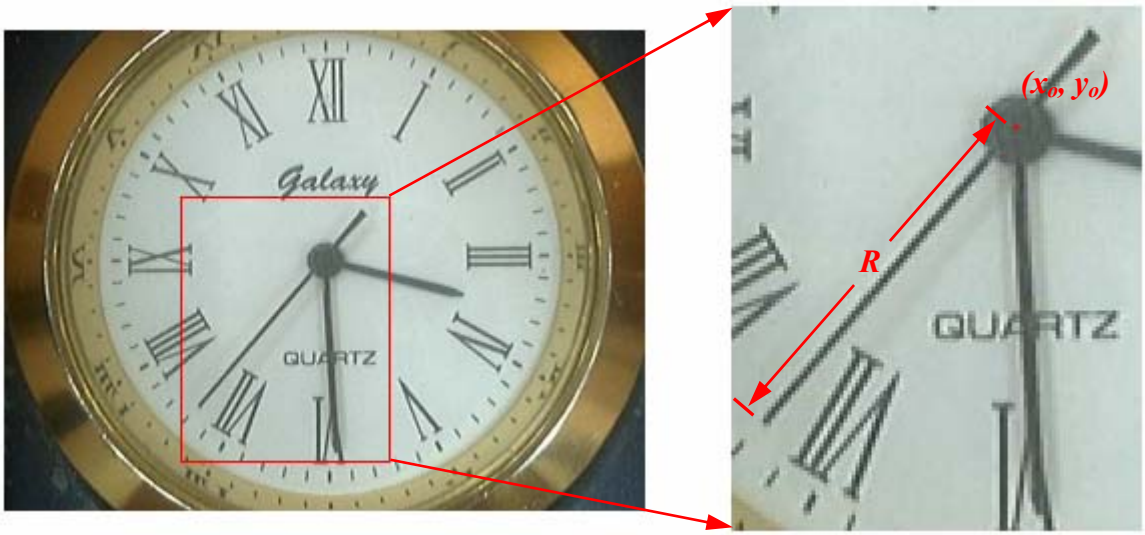
Saatin kaç olduğunu anlaşılması için saati oku düğmesi kullanılır. Saatin kaç olduğunun anlaşılma algoritması, saatin konumunun bulunması ve saatin okunması şeklinde iki aşamada gerçekleştirilir.

Saati oku düğmesine iki farklı durumda basılabilir. 1. durum, program çalıştırıldıktan sonra ilk kez saatin kaç olduğu anlaşılmak istendiğinde saati oku düğmesine basıldığı durumdur. Bu durumda saatin konumu bilinmediği için algoritmanın iki aşaması art arda uygulanır. 2. durum, en az bir kere saat okuma işlemi yapıldıktan sonra, tekrar saat saatin kaç olduğu anlaşılmak istendiğinde saati oku düğmesine basıldığı durumdur. Bu durumda saatin konumu bilindiği için sadece algoritmanın ikinci aşaması uygulanır.

Saatin kaç olduğunun anlaşılmasında kullanılacak görüntülerin yakalanması ağ kamerası ile yapılmaktadır. Yakalanacak görüntülerin sayısı saat oku düğmesine hangi durumda basıldığına göre değişmektedir. Birinci durumda saat oku düğmesine basıldığı andan itibaren aralarında birer saniye fark olan dört görüntü, ikinci durumda basıldığı andan itibaren ise bir saniye farkla iki yeni görüntü saatin kaç olduğunun anlaşılmasında kullanılmak üzere yakalanır. Yakalan görüntüler *320x240* boyutunda, bitmap biçimindedir ve renk modeli RGB'dir. Bu çalışmada kullanılan görüntüler renksiz olduğu görüntüler yakalandıktan sonra saat okuma işleminden önce, için yakalan görüntülerin RGB renk modelinden gri seviyeye dönüşümü yapılır.

**1. Durumda:** Yakalanan dört görüntüde kullanılarak, görüntülerde hata olup olmadığı kontrol edilir. Yakalanan görüntülerde hata varsa, program, otomatik olarak saati oku düğmesine 1. durumda basılmış gibi davranır. Yakalanan görüntülerde hata yoksa algoritmanın iki aşaması da işleme girer.

Algoritmanın ilk aşamasında dört adet görüntü kullanılarak saatin görüntü içindeki konumu bulunur. Saatin görüntü içindeki konumu Şekil 4-7'de görüldüğü gibi iki bileşenden oluşur. Birinci bileşen saatin orta noktası $(x_o,y_o)$, İkinci bileşen saniye kolunun orta noktaya en uzak noktasının orta noktaya olan öklidyen uzaklığı $R$'dir. Bu durumda saat, merkez noktası $(x_o, y_o)$'den en fazla $R$ uzaklığında olan noktaların oluşturduğu $R$ yarıçaplı çemberin içinde kalmaktadır. Saatin görüntü içindeki konumunun bulunmasında hata gerçekleşirse, program, otomatik olarak saat oku düğmesine 1. durumda basılmış gibi davranır.




**Şekil 4-7** Saat konumunun bileşenleri

Algoritmanın ikinci aşamasında ilk aşamada elde edilen saatin görüntü içindeki konumu ve yakalanan son iki görüntü kullanılarak saat okunur. Saatin okunmasında hata gerçekleşirse, 1. aşamada saatin konumu bulunduğu için, program, otomatik olarak saat oku düğmesine 2. durumda basılmış gibi davranır.

**2. Durumda:** Yakalanan iki görüntü kullanılarak saatin konumunda değişiklik olup olmadığı kontrol edilir. Saatin konumunda değişiklik var ise program, otomatik olarak saat oku düğmesine 1. durumda basılmış gibi davranır. Saatin konumunda değişiklik yoksa algoritmanın ikinci aşaması işleme girer.

Algoritmanın ikinci aşamasında, 1. durumda olduğu gibi, saatin görüntü içindeki konumu ve yakalanan iki görüntü kullanılarak saat okunur. Saatin okunmasında hata gerçekleşirse, saatin konumu bilindiği için, program, otomatik olarak saat oku düğmesine 2. durumda basılmış gibi davranır.

Algoritmanın ikinci aşaması iki durumda da işleme girmektedir. İkinci aşamanın iki durumda da işleme girmesindeki tek farklılık, kullanılan görüntülerin farklı olmasıdır.

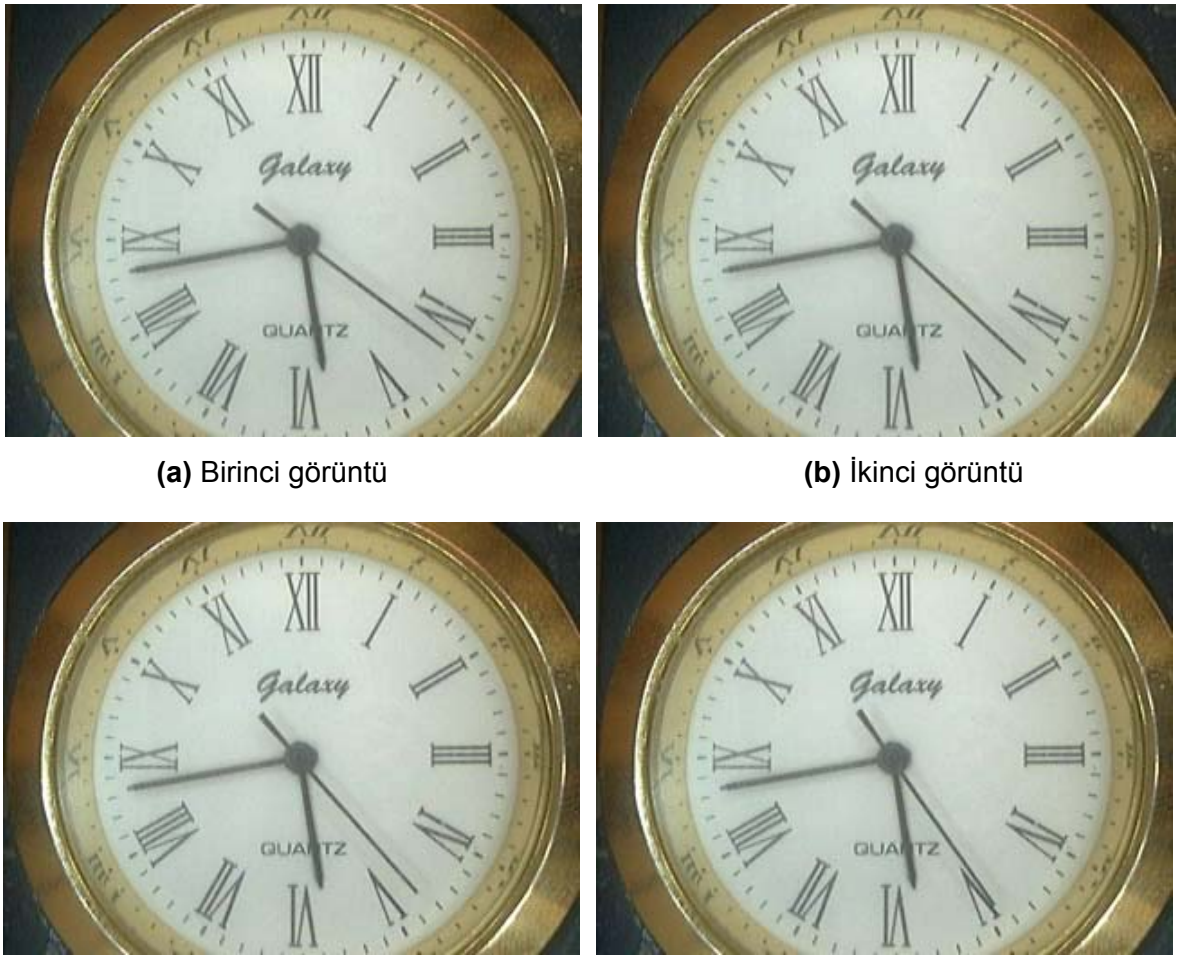
Arayüz'deki yazı pencerelerinde,

- Her iki durumda yakalanan görüntülerin yakalama zamanları,
- 1. durumda bulunan saatin orta noktası,
- Okunan saat,
- Programın çalışması süresince, programın durumu ve elde edilen hata mesajlarıyla ilgili bilgiler

görüntülenmektedir.

## 4.2. Yakalanan Görüntüler ve Gri Seviyeye Dönüşümleri

Saati oku düğmesine basıldıktan sonra art arda yakalanan iki saat görüntüsü arasında saniye kolunun değişimini en kısa sürede algılamak için, yakalanan bir görüntüden sonra bir saniye (1000 milisaniye) bekleme süresi gerekmektedir. Saati oku düğmesine 1. durumda basılmasıyla, 1000 milisaniye farklar ile yakalanmış ve saat okuma işleminde kullanılmış dört saat görüntüsü Şekil 4-8'de görülmektedir




**(a)** Birinci görüntü **(b)** İkinci görüntü

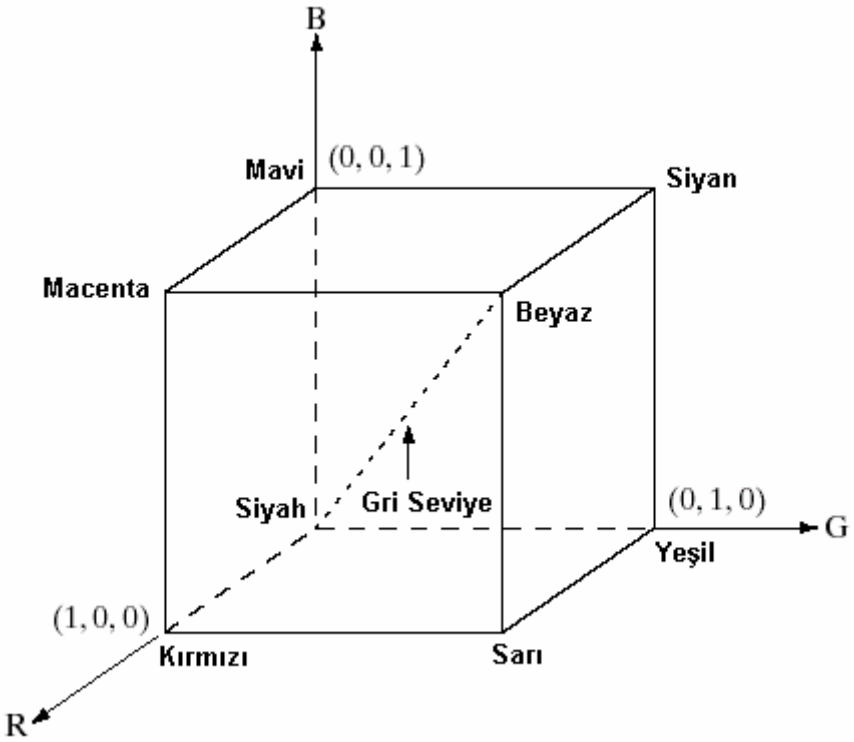
**(c)** Üçüncü görüntü **(d)** Dördüncü görüntü

**Şekil 4-8** 1000 milisaniye farklarla yakalanmış, dört renkli saat görüntüsü

Saatin kaç olduğunun anlaşılmasının gerçek zamanlı davranış sergilemesi için, yakalan görüntüler belleğe kaydedilmemektedir. Görüntülerin belleğe kaydedilmesi, tekrar bellekten okunması, görüntü Bitmap biçimindeyken üzerinde işlem yapılması, işlem süresinin gereksiz uzamasına neden olmaktadır. RGB renk modeline sahip Bitmap biçiminde yakalanan görüntüleri belleğe kaydetmek yerine, görüntülerdeki renk bilgisi, Ek3'de gösterilen C# tabanlı kod parçası ile tek boyutlu diziye aktarılmaktadır. Diziye aktarılan renk bilgisi kullanılarak, RGB renk modeline sahip bitmap biçiminde yakalanmış görüntüler gri seviyeli görüntülere dönüştürülerek iki boyutlu dizilerde tutulur.

RGB renk modeli ile ilgili kısa bir önbilginin verilmesi uygun bulunmuştur.

RGB renk modelindeki her renk, tayfın birincil bileşenleri olan kırmızı, yeşil ve maviden oluşur [9]. Bu modelin temeli kartezyen koordinat sistemidir. RGB renk modelinin renk altuzayı Şekil 4-9'da görüldüğü gibi küptür. Birincil renkler kırmızı, yeşil ve mavi küpün eksenler üzerinde bulunan köşelerindedir. Macenta, siyan ve sarı küpün diğer üç köşesindedir. Siyah orijin noktasında, beyaz ise orijinden en uzak noktadaki köşededir. Gri seviye ölçeği, birincil bileşenlerin üçünün de eşit olduğu noktaların oluşturduğu siyahtan beyaza giden doğru üzerinde bulunur. Diğer değişik renkler, küpün üzerindeki veya içindeki noktalara karşılık gelir.




**Şekil 4-9** RGB renk küpü [9]

RGB renk modelinde görüntüler, her biri ayrı birincil renge karşılık gelmek üzere üç adet görüntü bileşeninden oluşur. Görüntüde *(x,y)* koordinatındaki herhangi bir pikselin rengi, üç ayrı görüntü bileşeninin RGB ekranında karışmasıyla belirlenir. Bir görüntü bileşenindeki renk sayısı, bu görüntü bileşenindeki renk değerleri için kaç bit ayrıldığına bağlıdır. Bu tez çalışmasında yakalanan renkli görüntülerin RGB renk modelinde bir pikselin her bir birincil bileşeni için 8 bit yer ayrılmıştır. Örneğin G yani yeşil bileşeni için 8 bit yer

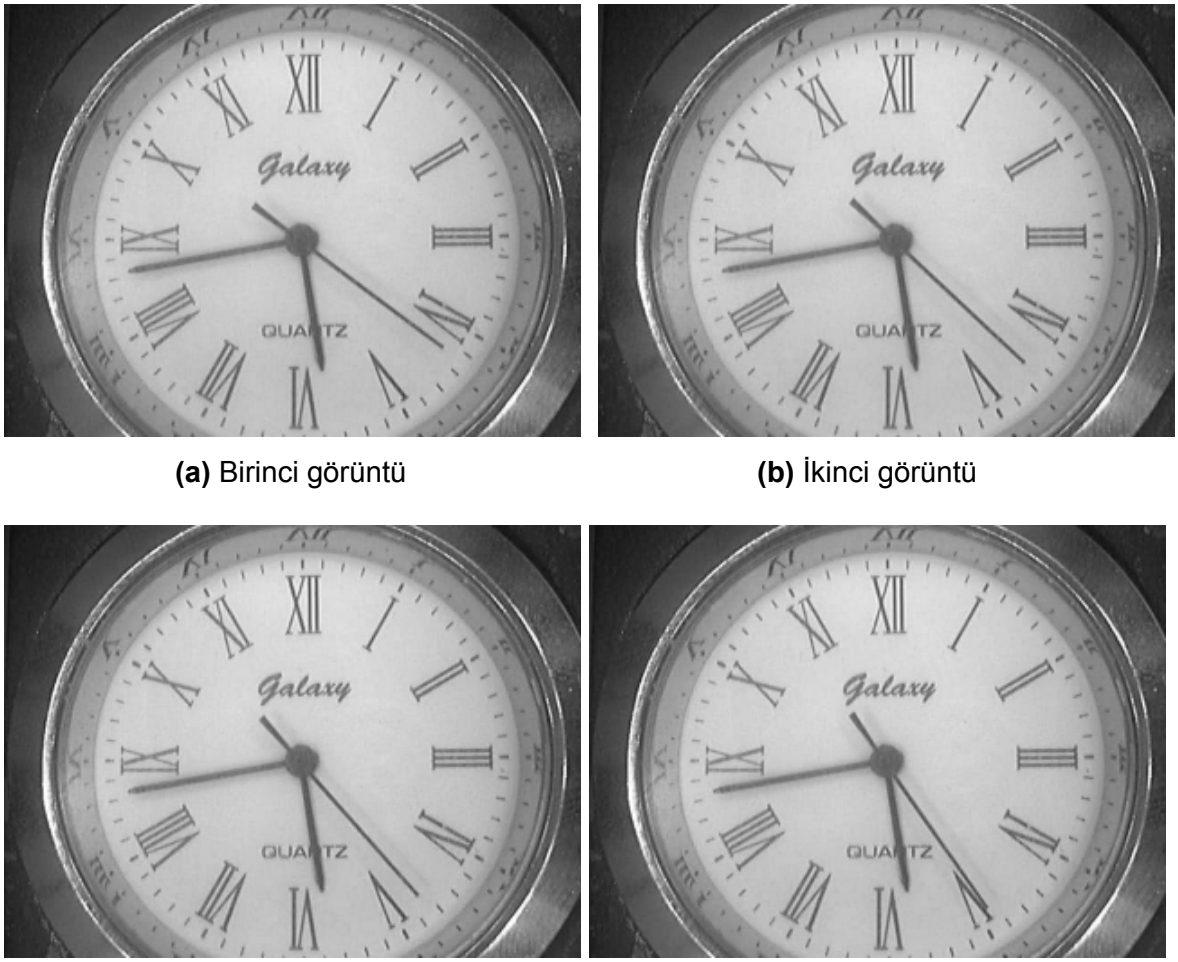
ayrılmışsa görüntüdeki bütün piksellerin yeşil bileşeni *0* ile *255* arasındaki *256* değerden birini alabilir.

Görüntülerdeki *(x, y)* koordinatlarındaki bir pikselin RGB renk modelinden gri seviyeye dönüşümü NTSC standardına göre Eşitlik 4-1'deki denklem ile gerçekleştirilir [9].

$$f(x,y) = 0.2989 * R(x,y) + 0.587 * G(x,y) + 0.114 * B(x,y) \qquad (4\text{-}1)$$

Gerçek zamanlı davranış sergilenmesi için dönüştürme işlemleri görüntüler yakalandığı andan bir sonraki görüntü yakalanana kadar geçen bekleme süresinde yapılmaktadır. 110 milisaniyelik süre bir görüntü yakalandıktan sonra başka bir görüntü yakalana kadar geçen en kısa süredir. Dönüştürme işlemi de yaklaşık 20 milisaniye sürmektedir. Art arda yakalanan iki saat görüntüsünün yakalanma zamanları arasındaki 1000 milisaniyelik süreyi tamamlamak için 870 milisaniye de gecikme süresi verilmiştir.

Şekil 4-8'deki dört saat görüntüsünün gri seviyeye dönüştürülmüş halleri de Şekil 4-10'da görülmektedir. Saatin kaç olduğunun anlaşılması için yakalanan görüntüler dışında bu bölümde gösterilen görüntüler sadece bilgilendirme amaçlı elde edilmiştir. Saat okuma sistemin normal çalışmasında görüntüler sadece iki boyutlu dizilerde tutulmaktadır.




**(a)** Birinci görüntü **(b)** İkinci görüntü

**(c)** Üçüncü görüntü **(d)** Dördüncü görüntü

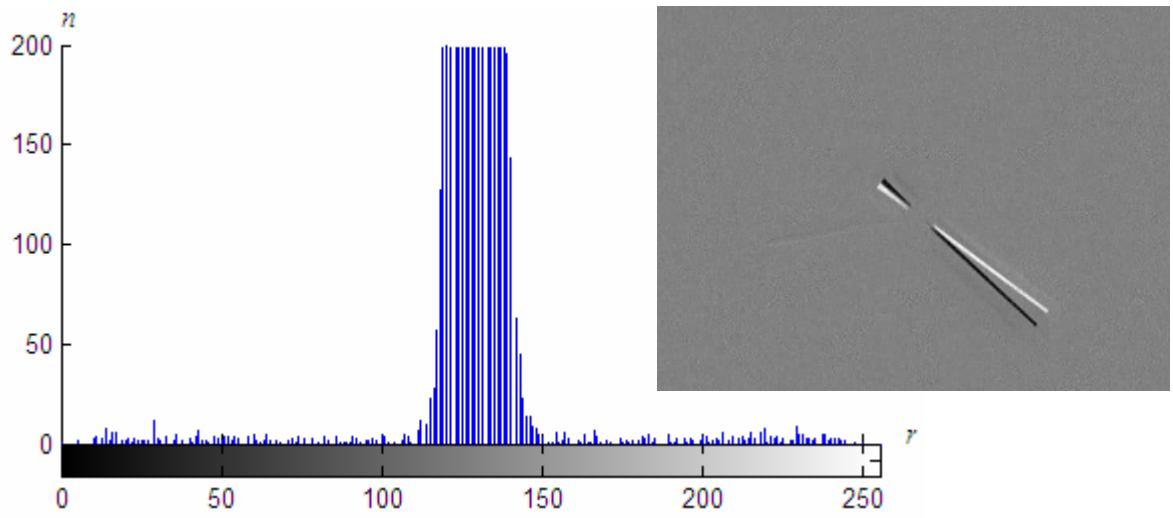
**Şekil 4-10** Yakalanan görüntülerin gri seviyeye dönüştürülmüş halleri

## 4.3. Yakalanan Görüntülerin Doğruluğu

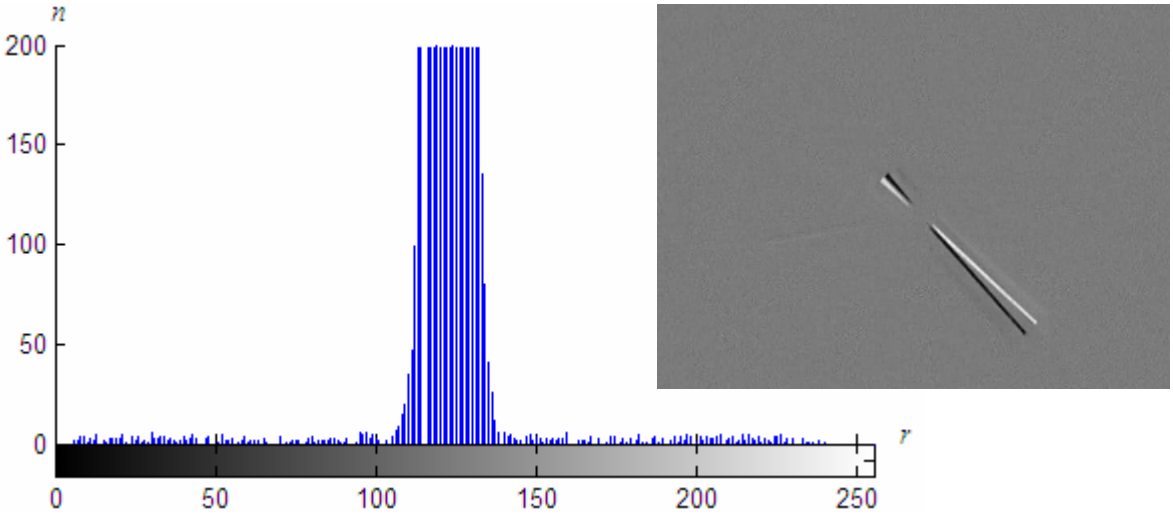
Görüntülerin yakalanması sırasında önemli olan nokta, bir saniye farkla yakalanan görüntülerde saniye kolunun hareket etmiş olması ve görüntü içinde başka hareket olmamasıdır. Bir saniye farkla yakalanmış iki görüntüde, ağ kamerası veya bilgisayardan kaynaklanan bir nedenle saniye kolu konum değiştirmemişse veya görüntülerin yakalandığı alan içinde başka nesnede hareket olmuş ise, yakalanan görüntülerin hatalı olduğu sonucuna ulaşılır. Hatalı görüntüler kullanıldığında saatin okunmasında da hata oluşur. Saatin hatalı okunmasını engellemek için sadece saniye kolunun hareket ettiğinin algılanması yani yakalan görüntülerin doğruluğunun kontrolü gerekmektedir.

Şekil 4-10'daki bir saniye farkla yakalanan görüntüler arasında saniye kolunun hareket edip etmediğinin algılanmasında aşağıdaki adımlar izlenerek bir saniye farkla yakalanmış iki görüntüde farklı konumlarda bulunan nesneler görüntüden bölütlenmiştir. Hareket eden nesnelerin alanları hesaplanarak iki görüntü arasında yeterli hareket olup olmadığı veya gereğinden fazla hareket olup olmadığı kontrol edilmiştir:

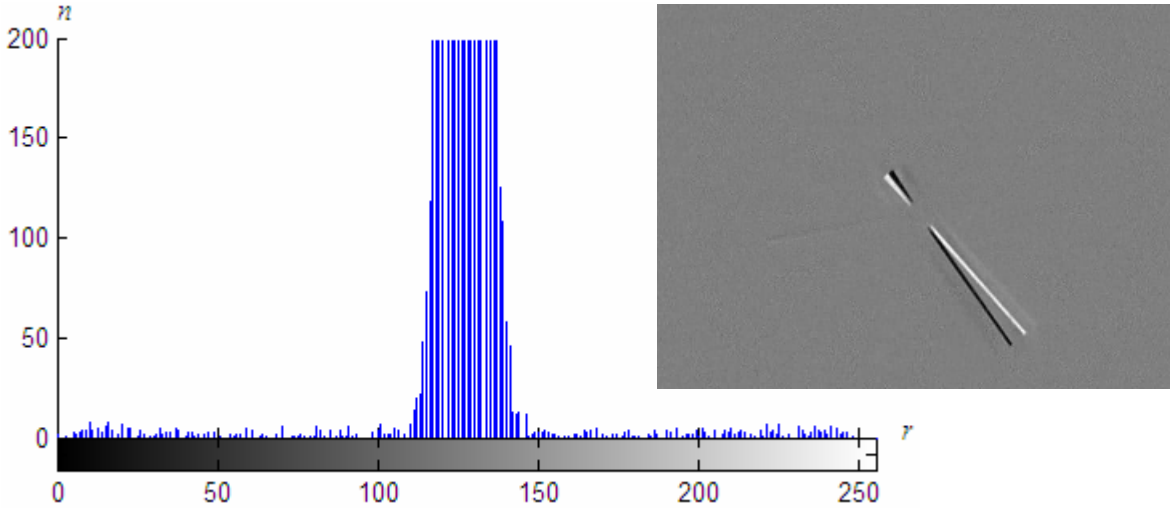
1. **Adım:** Şekil 4-10'daki görüntülerde Eşitlik 3.11 kullanılarak üç fark görüntüsü elde edilir ve görüntülerin gri seviye değerleri $[0,255]$ arasında kalacak şekilde fark görüntüleri Bölüm 3.1.3'de anlatılan ölçekleme yöntemiyle ölçeklenir. Fark görüntülerinin ölçeklenmiş hali ve histogramları Şekil 4-11'de görülmektedir. Histogramlardaki en önemli nokta, arka plan piksellerinin oluşturduğu doruğun, hareket eden nesnelerin oluşturduğu doruklara göre çok yüksek olmasıdır. Şekil 4-11(a)'daki histogramın en büyük değeri 283337, Şekil 4-11(b)'deki histogramın en büyük değeri 283380 ve Şekil 4-11(c)'deki histogramın en büyük değeri 283427'dir. Şekil 4-11'deki histogramların uygun gösterimi için en büyük değerleri 200 piksel ile sınırlandırılmıştır.




**(a)** 1. ve 2. görüntülerden elde edilen fark görüntüsü ve histogramı



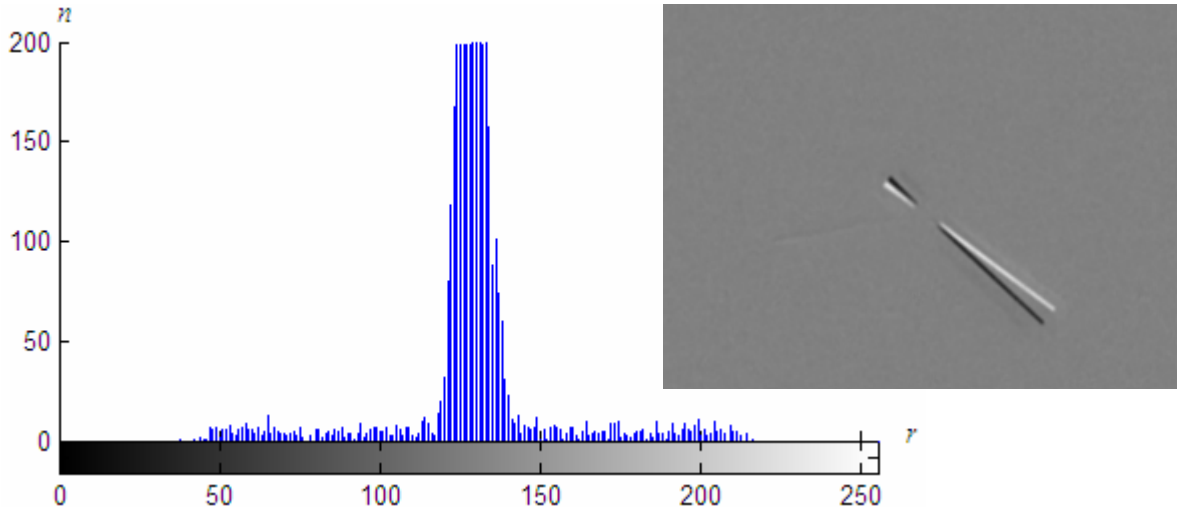

**(b)** 2. ve 3. görüntülerden elde edilen fark görüntüsü ve histogramı



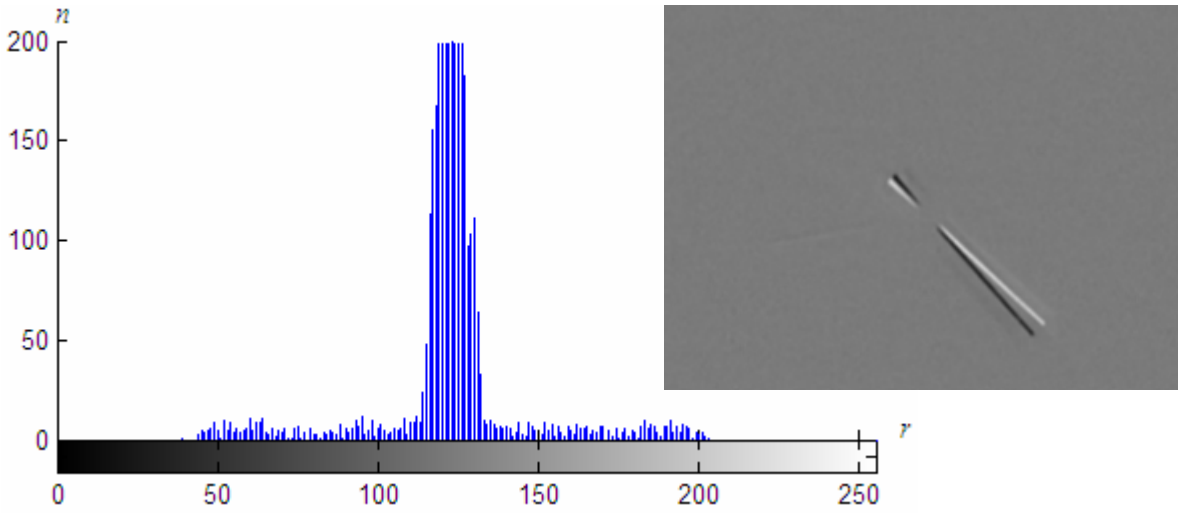

**(c)** 3. ve 4. görüntülerden elde edilen fark görüntüsü ve histogramı

**Şekil 4-11** Fark görüntüleri ve histogramları

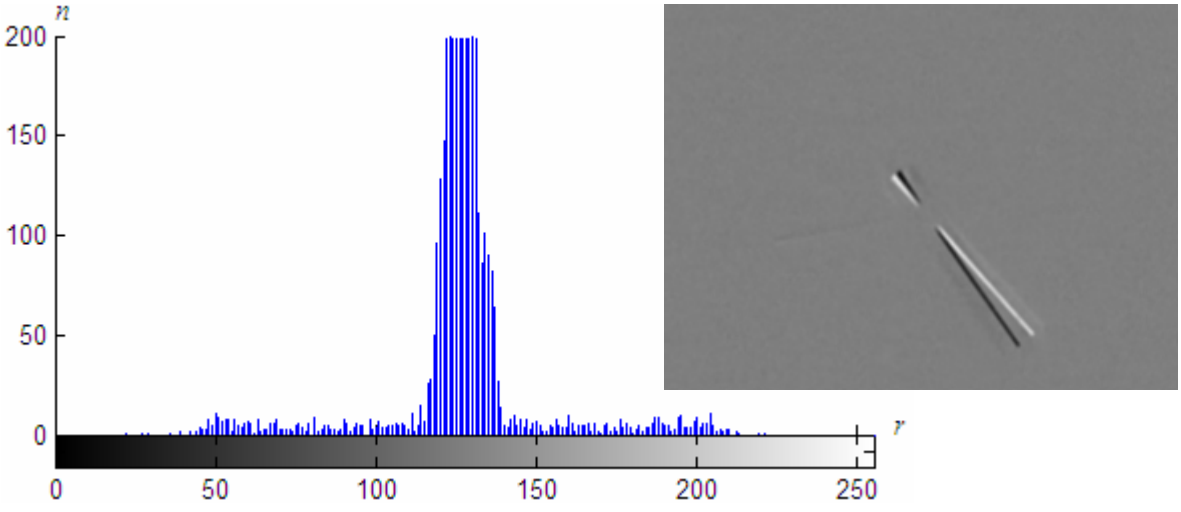
2. **Adım:** Ölçeklenen fark görüntülerine eşikleme uygulamadan önce görüntü yumuşatma işlemi uygulanarak hareket eden nesnelerin yanında gürültü kabul edilebilecek küçük değişikler fark görüntüsünden çıkarılır. Yumuşatma işlemi Şekil 3-6(a)'da gösterilen ortalama süzgeç uygulanarak gerçekleştirilmiştir. Yumuşatılmış fark görüntüleri ve histogramları Şekil 4-12'de görülmektedir.




**(a)** 1. Fark görüntüsü ve histogramı




**(b)** 2. fark görüntüsü ve histogramı




**(c)** 3. fark görüntüsü ve histogramı

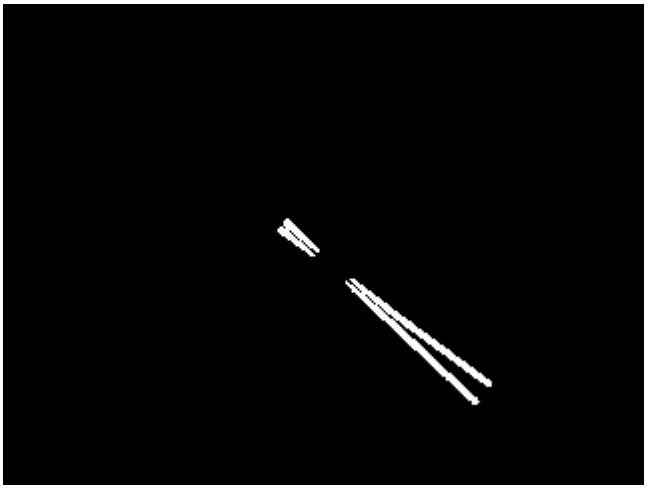
**Şekil 4-12** Yumuşatılmış fark görüntüleri ve histogramları

3. **Adım:** Yumuşatılmış fark görüntülerinin histogramlarında, değişim olmayan noktalar 127 gri seviye değerinin çevresinde gruplanmıştır. Görüntüdeki nesnelerin hareketinden kaynaklanan değişim olan noktalar ise, hareket eden nesneler iki görüntüde farklı konumlarda bulunduğu için, grubun sağ ve sol tarafına dağılmıştır. Bölüm 3.1.1'de ayrıntılı olarak anlatılan otomatik eşik değeri bulma yöntemlerin ikincisi üç fark görüntüsüne de uygulanmıştır. Değişim olmayan noktaların oluşturduğu grubun hem sağ tarafında hem de sol tarafında kalan değişim noktalarının görüntüden ayrılması için, her fark görüntüsü için iki ayrı eşik değeri bulunmuştur. Fark görüntüleri için bulunan eşik değerleri Çizelge 4-1'de gösterilmiştir. Bulunan iki eşik değeri ile görüntülere Eşitlik 4.2'deki eşikleme algoritması uygulanarak değişim olan noktalar 255 gri seviye ve değişim olmayan noktalarda 0 gri seviye değerine atanmıştır. Yumuşatılmış fark görüntülerinin eşikleme işlemi sonucundaki halleri Şekil 4-13'de gösterilmiştir.

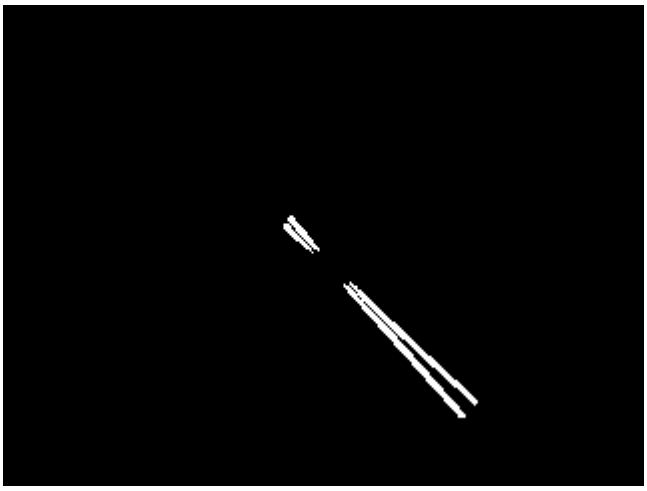
$$g(x,y) = \begin{cases} 255 & eğer\ f(x,y) \geq T_r\ veya\ f(x,y) \leq T_l \\ 0 & diğer\ durumlarda \end{cases} \qquad (4.2)$$

**Çizelge 4-1** Eşik değerleri

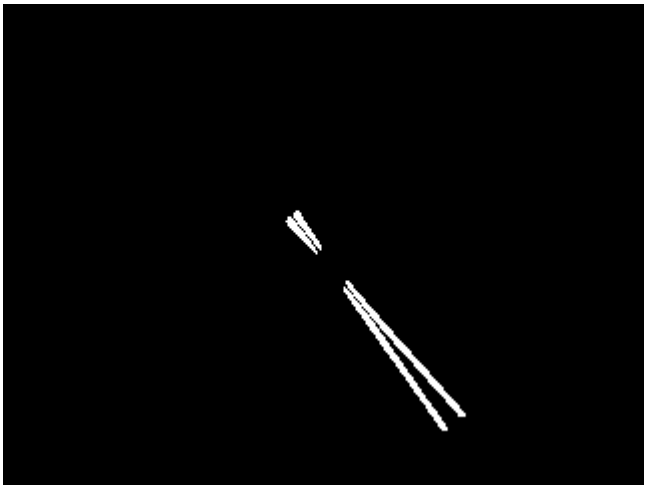
| | **1. Fark Görüntüsü** | **2. Fark Görüntüsü** | **3. Fark Görüntüsü** |
|---|---|---|---|
| **Sol Eşik Değeri ($T_l$)** | *107* | *99* | *102* |
| **Sağ Eşik Değeri ($T_r$)** | *154* | *152* | *151* |



**(a)** 1. Fark görüntüsü



**(b)** 2. Fark görüntüsü



**(c)** 3. Fark görüntüsü

**Şekil 4-13** Yumuşatılmış ve eşiklenmiş fark görüntüleri

4. **Adım:** Eşikleme işlemi sonucunda fark görüntüsünde 255 gri seviye değeri ile ifade edilen değişim noktalarının alanları her fark görüntüsü için hesaplanmıştır. Alan değerleri Çizelge 4-2'de gösterilmiştir. Bulunan alan değerleri, sadece saniye kolunun hareketini içeren fark görüntülerinden elde edilen en büyük ve en küçük değişim değerleri ile karşılaştırılmıştır. En büyük değişim değeri saniye kolunun görüntünün dışına çıkmadığı saatin ağ kamerasına en yakın olduğu konumda yakalanan test görüntüleri kullanılarak, 900 piksel olarak uygun görülmüştür. En küçük değişim değeri, saat okuma sisteminin başarılı çalışmayı sürdürebildiği saatin ağ kamerasına en uzak olduğu konumdaki görüntüler kullanılarak, 100 piksel olarak uygun görülmüştür. Bulunan alan değerlerinden herhangi biri en küçük değerden küçük ise arayüzün bilgi penceresinde görüntülerde yeterli değişim yoktur bilgisi verilir ve program, otomatik olarak saat oku düğmesine 1. durumda basılmış gibi davranır. Bulunan alan değerlerinden herhangi biri en büyük değerden büyük ise arayüzün bilgi penceresinde yakalanan görüntülerde çok fazla değişim vardır bilgisi verilir ve program, otomatik olarak saat oku düğmesine 1. durumda basılmış gibi davranır.

**Çizelge 4-2** Değişim noktaları alan değerleri

| | **1. Fark Görüntüsü** | **2. Fark Görüntüsü** | **3. Fark Görüntüsü** |
|---|---|---|---|
| **Alan Değerleri** | *733 piksel* | *650 piksel* | *701 piksel* |

Çizelge 4-2'deki alan değerleri ile en büyük ve en küçük değişim değerleri karşılaştırıldığında değişim noktalarının alanlarının istenen aralıkta kaldığı, yakalan görüntülerin doğru olduğu belirlenmiştir. Görüntülerin doğruluğu belirlendikten sonra saatin konumun bulunmasına geçilmiştir.

Görüntülerin yakalanmasında sırasında, bir saniye farkla yakalanan görüntülerde saniye kolundan başka küçük boyutlu bir nesnede hareket gerçekleşmişse, değişim noktalarının alan değerlerinin istenen aralıkta kalma olasılığı düşük de olsa vardır. Eğer bu durum gerçekleşirse, saatin konumunun bileşeni olan saatin orta noktasının bulunmasında bu durum algılanmaktadır.

## 4.4. Saatin Konumunun Bulunması

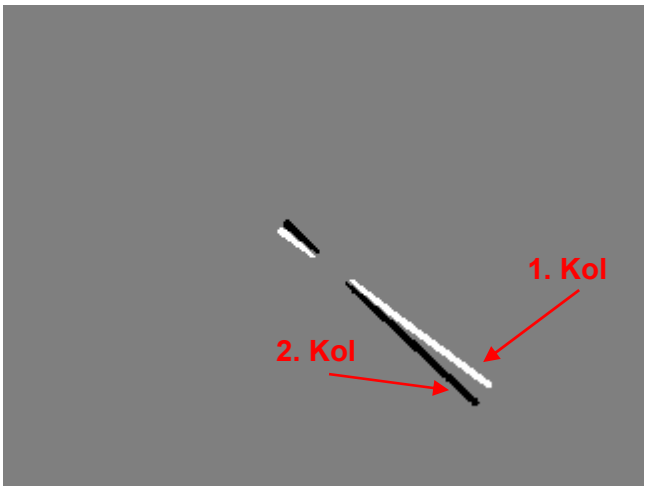
Saatin kaç olduğunun anlaşılması algoritmasının birinci aşaması saatin konumunun bulunmasıdır. Saatin konumunun yani saatin orta noktasının ve saniye kolunun orta noktaya en uzak noktasının orta noktaya olan öklidyen uzaklığının bulunmasında saniye kolunun dört farklı konumu kullanılmaktadır. Saatin konumunun birinci bileşeni olan orta noktanın bulunması, dört farklı saniye kollarının kesişim noktaları kullanılarak gerçekleşmektedir. Saatin konumunun ikinci bileşeni olan uzaklık değeri de orta nokta ve dört farklı konumdaki saniye kolu kullanılarak bulunmaktadır.

Saniye kolunun dört faklı konumda iken görüntüden bölütlenmesinde Şekil 4-12(a) ve Şekil 4-12(c)'deki yumuşatılmış fark görüntüleri kullanılmıştır. Eşitlik 4.3'deki eşikleme algoritması fark görüntülerine uygulanarak, fark görüntülerinde değişim olmayan noktalar 127 gri seviye değerine, değişim olmayan grubun sağındaki değişim olan noktalar 255 gri seviye değerine, değişim olmayan grubun solundaki değişim olan noktalar da 0 gri seviye değerine atanmıştır. Eşikleme algoritmasında Çizelge 4-1'deki eşik değerleri kullanılmıştır.

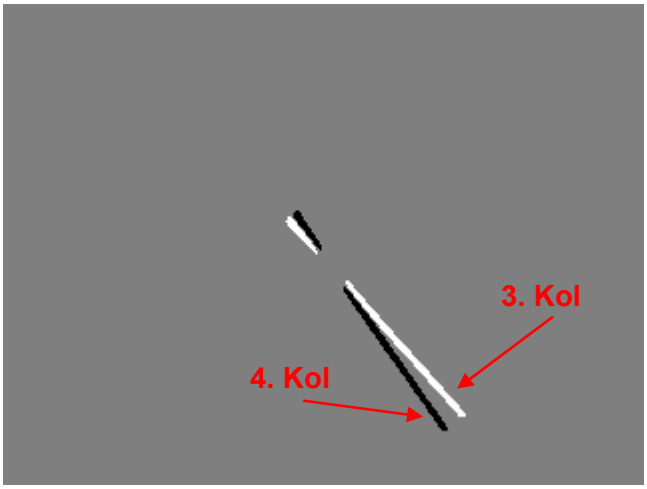
$$g(x,y)=\begin{Bmatrix}255 & eğer\ f(x,y)\geq T_r\\ 0 & eğer\ f(x,y)\leq T_l\\ 127 & diğer\ durumlarda\end{Bmatrix} \qquad (4.3)$$

Yumuşatılmış fark görüntülerinin eşikleme işlemi sonucundaki halleri Şekil 4-14'de gösterilmiştir. Her bir fark görüntüsünde iki ayrı konumda bulunan saniye kolu bulunmaktadır. Saniye kollarının kesişim noktalarını bulmak için kollar numaralandırılmıştır. 1. fark görüntüsünde bulunan 255 gri seviye değerine sahip piksellerden oluşan saniye kolu 1. kol, 0 gri seviye değerine sahip piksellerden oluşan saniye kolu 2. kol olarak belirlenmiştir. 3. fark görüntüsünde bulunan 255 gri seviye değerine sahip piksellerden oluşan saniye kolu 3. kol, 0 gri seviye değerine sahip piksellerden oluşan saniye kolu da 4. kol olarak belirlenmiştir. Numaralandırmaya göre, kullanılan saatin göstergesinin arka planı kollara göre koyu da olsa, açık da olsa her zaman 1. kol ile 3. kol ve 2. kol ile 4. kol arasında iki saniyelik konum farkı bulunur.

Şekil 4-14'de görüldüğü gibi bütün saniye kolları iki ayrı parçadan oluşmaktadır. Bunun nedeni saatlerde orta noktaları merkez olacak şekilde küçük yarıçaplardaki çemberler içindeki bölgede değişim olmamasıdır. Bu çemberler içindeki piksellerde değişim olmadığı için eşikleme işlemi sırasında saatin arka planındaki pikseller gibi 127 değerini almaktadır.




**(a)** 1. Fark görüntüsü




**(b)** 3. Fark görüntüsü

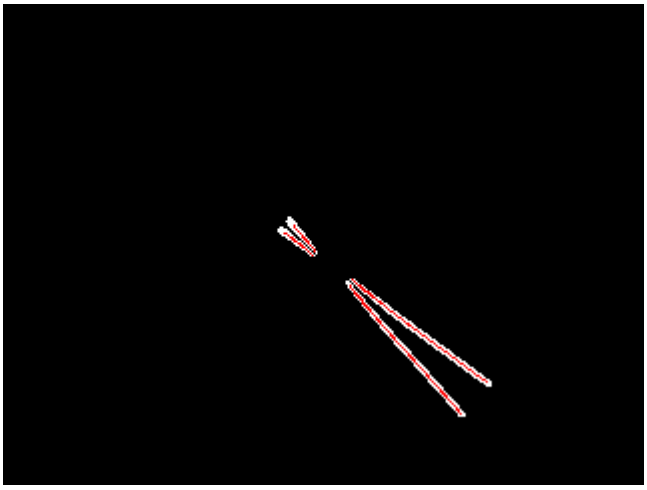
**Şekil 4-14** Yumuşatılmış ve eşiklenmiş fark görüntüleri

Farklı konumlarda bulunan iki saniye kolunun iki ayrı doğru parçası olarak ele alırsak iki doğru parçasının kesiştiği nokta saatin orta noktasına çok yakın sonuç vermektedir. Sadece saniye kolunun konum değiştirmesi için geçen en kısa süre bir saniyedir. Gerçek zamanlı çalışma açısından, saatin orta noktasının bulunması için, aralarında en kısa süre olan bir saniyelik konum farkı olan saniye

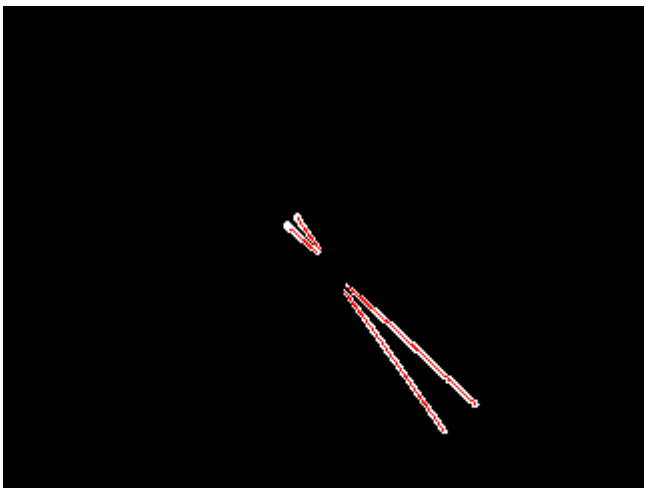
kollarının kullanılması uygun olmaktadır, fakat bu sürede konum farkı olan iki saniye kolunun aralarındaki açının çok az olması sebebiyle, kesiştikleri nokta beklenilen sonucu vermemektedir. Bu nedenle saniye kolunun konum değiştirmesi için geçen ikinci en kısa süre olan 2 saniyelik konum farkı kullanılmaktadır.

Tek bir kesişim noktasının da hatalı olma olasılığı bulunduğu için iki ayrı kesişim noktası bulup kesişim noktalarının birbirlerine yakınlığının kontrolü gerekmektedir. Bu iki ayrı kesişim noktası, numaralandırılan saniye kolları kullanılarak bulunmaktadır. Aralarında 2 saniyelik konum farkı olduğu bilinen 1. ile 3. saniye kolunun kesiştiği nokta 1. kesişim noktası, 2. ile 4. saniye kolunun kesiştiği nokta ise 2. kesişim noktası olarak belirlenir.

Kesişim noktalarının iyi sonuç vermesi için 4 farklı konumdaki saniye kolu Bölüm 3.2.3'de bahsedilen inceltme algoritmasıyla inceltilmiştir. İnceltme işlemi sonucunda, saniye kollarının ana ekseni üzerinde bulunan iskeletleri çıkarılmıştır. 1. ile 3. saniye kolları ve iskeletleri Şekil 4-15(a)'da, 2. ile 4. saniye kolları ve iskeletleri Şekil 4-15(b)'de görülmektedir. Şekillerde, iskeletler kırmızı çizgilerle gösterilmektedir.



**(a)**



**(b)**

**Şekil 4-15 (a)** 1. ile 3. saniye kolları ve iskeletleri
**(b)** 2. ile 4. saniye kolları ve iskeletleri

Daha önceden belirtildiği gibi her saniye kolu iki ayrı parçadan oluşmaktadır. Bu nedenle bütün iskeletler de iki ayrı doğru parçasından oluşmaktadır. Kesişim noktaları bulurken iskeletlerin iki doğru parçasından oluşmaları önem taşımamaktadır. Kesişim noktalarının bulunması ile ilgili kısa bir önbilginin verilmesi uygun bulunmuştur.

Geometrik olarak aynı düzlem içindeki iki doğru birbirine paralel değilse kesişir. Ancak, bilgisayar destekli çizim ve genel olarak grafik programlamada iki doğrunun değil iki doğru parçasının kesişimi önemlidir [28]. Eşitlik 4.3 ve Eşitlik 4.4'de gösterilen denklemlerin belirttiği doğruların kesişim noktasının $x$ koordinatı $x_s$ Eşitlik 4.6'de, $y$ koordinatı $y_s$ Eşitlik 4.7'de verilmiştir.

$$Ax + Bx + C = 0 \quad (4.4)$$

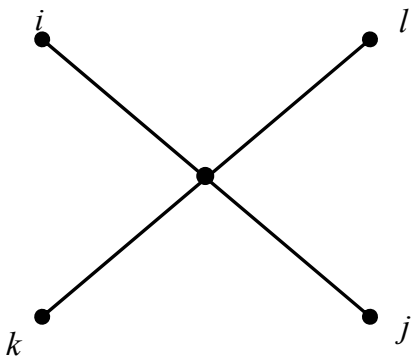
$$Ax + Bx + C = 0 \quad (4.5)$$

$$x_s = (GB - FB)/(FA - EB) \quad (4.6)$$

$$y_s = (CE - AG)/(FA - EB) \quad (4.7)$$

İki doğru paralel ise, $FA - EB = 0$ olur. Bu durumda iki doğrunun kesişiminden söz edilemez.

Yukarıdaki denklemler kullanılarak önce kesişim noktası hesaplanır, sonra kesişim noktasının iki doğrunun üzerinde olup olmadığı araştırılır. Çeşitli özel durumların dikkate alınmasını gerektiren bu yaklaşım yerine, programlama açısından daha kullanışlı olan ve nokta koordinatlarına dayanan aşağıdaki yöntem izlenebilir [28]. Şekil 4.16'da görüldüğü gibi birinci doğru parçasının başlangıç ve bitim noktaları $i(x_i, y_i)$ ve $j(x_j, y_j)$, ikinci doğru parçasının başlangıç ve bitim noktaları $k(x_k, y_k)$ ve $l(x_l, y_l)$ olsun.




**Şekil 4-16** İki doğru parçasının kesişimi [28]

İlk olarak doğru parçalarının paralel olup olmadığı kontrol edilir. Doğru parçalarının paralel olup olmadığının kontrolünü yapmak için Eşitlik 4.8'deki $d$ parametresi hesaplanır.

$$d = (x_j - x_i)(y_k - y_l) - (x_k - x_l)(y_j - y_i) \quad (4.8)$$

$d = 0$ ise doğru parçaları paralel olup kesişim hesaplanamaz. $d \neq 0$ ise doğru parçaları kesişir. Eşitlik 4.9'daki $p_1$ ve Eşitlik 4.10'deki $p_2$ parametreleri hesaplanır.

$$p_1 = \frac{(x_k - x_l)(y_i - y_k) - (x_i - x_k)(y_k - y_l)}{d} \quad (4.9)$$

$$p_2 = \frac{(x_i - x_k)(y_j - y_i) - (x_j - x_i)(y_i - y_k)}{d} \quad (4.10)$$

$0 < p_1 < 1$ ve $0 < p_2 < 1$ ise kesişim noktası doğru parçalarının üzerindedir. Kesişim noktasının koordinatları Eşitlik 4.11 ve Eşitlik 4.12'daki gibi bulunur.

$$x_s = x_i + p_1(x_j - x_i) \quad (4.11)$$

$$y_s = y_i + p_1(y_j - y_i) \quad (4.12)$$

1. kesişim noktasını bulmak için 1. ve 3. saniye kollarının iskeletlerini doğru parçaları olarak düşünebiliriz. İki doğru parçası için ayrı ayrı birbirine en uzak iki nokta bulunur. Bulunan noktalar aynı zamanda birbirini kesen iki doğru parçasının başlangıç ve bitiş noktalarıdır. Her iki doğru parçası için hesaplanan başlangıç ve bitiş noktaları kullanılarak 1. ve 3. saniye kollarının kesişim noktası bulunur.

2. kesişim noktasının bulunması için 2. ve 4. saniye kollarının iskeletleri doğru parçaları olarak ele alınır ve 1. kesişim noktası bulunurken izlenen yol izlenerek 2. kesişim noktası da bulunur.

1., 2., 3., ve 4. doğru parçalarının başlangıç ve bitiş noktalarının koordinatları, Çizelge 4-3'de görülmektedir. 1. ve 2. kesişim noktalarının koordinatları Çizelge 4-4'de görülmektedir.

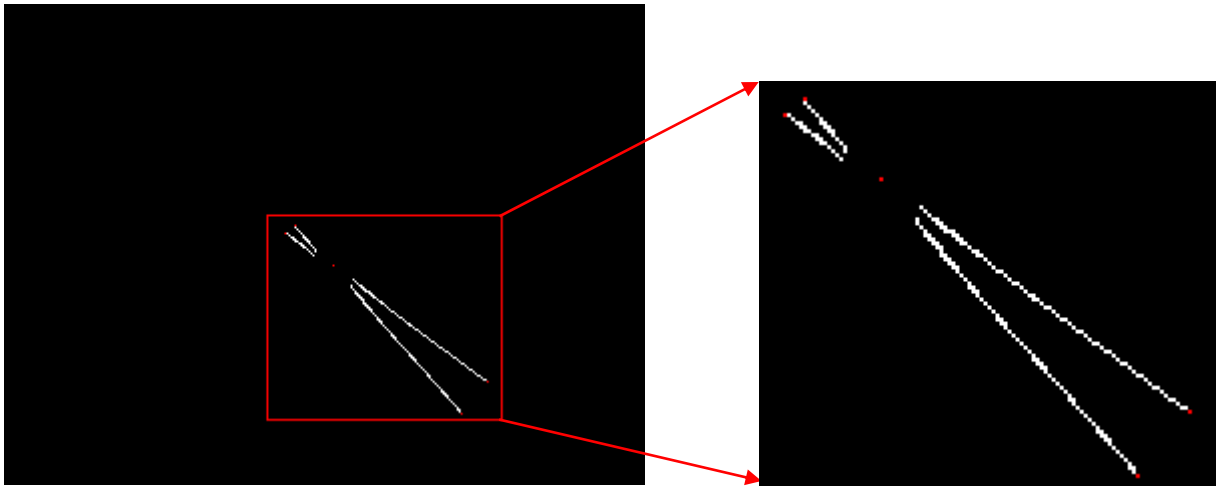
**Çizelge 4-3** Doğru parçalarının başlangıç ve bitiş noktalarının koordinatları

| | **Başlangıç Koordinatı** | | **Bitiş Koordinatı** | |
|---|---|---|---|---|
| | ***x*** | ***y*** | ***x*** | ***y*** |
| **1. Doğru parçası** | *149* | *114* | *241* | *188* |
| **2. Doğru parçası** | *143* | *111* | *235* | *198* |
| **3. Doğru parçası** | *145* | *110* | *228* | *204* |
| **4. Doğru parçası** | *147* | *105* | *219* | *211* |

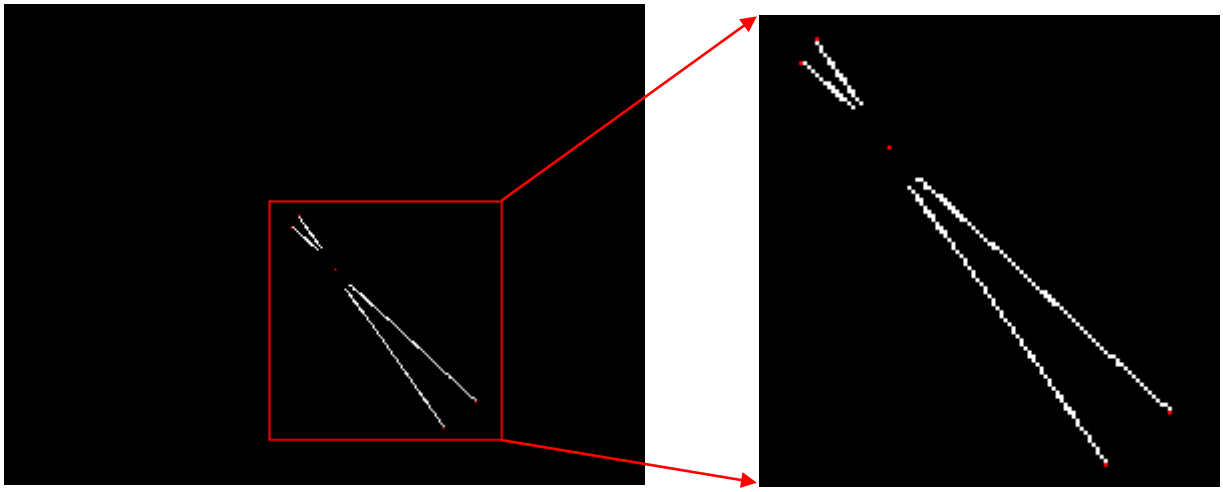
**Çizelge 4-4** 1. ve 2. kesişim noktalarının koordinatları

| | ***X*** | ***y*** |
|---|---|---|
| **1. Kesişim Noktası** | *164* | *130* |
| **2. Kesişim Noktası** | *165* | *132* |

1. ve 3. doğru parçaları, doğru parçalarının başlangıç, bitiş noktaları ve 1. kesişim noktası Şekil 4.17(a)'da görülmektedir. 2. ve 4. doğru parçaları, doğru parçalarının başlangıç, bitiş noktaları ve 2. kesişim noktası Şekil 4.17(b)'de görülmektedir.



**(a)**



**(b)**

**Şekil 4-17 (a)** 1. ve 3. doğru parçalarının başlangıç, bitiş ve kesişim noktaları
**(b)** 2. ve 4. doğru parçalarının başlangıç, bitiş ve kesişim noktaları

1. ve 2. kesişim noktalarının birbirine çok yakın çıkması beklenmektedir. 1. kesişim noktası $(x_1, y_1)$, 2. kesişim noktası $(x_2, y_2)$ olsun, iki nokta arasındaki öklidyen uzaklık Eşitlik 4.13'de gösterildiği gibi 8 pikselden küçük ise kesişim noktaları birbirine yakın kabul edilmektedir.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \leq 8 \qquad (4.13)$$

Kesişim noktalarının birbirine yakın olmamasının temel nedeni: görüntülerin yakalanması sırasında, bir saniye farkla yakalanan görüntülerde saniye kolundan başka küçük boyutlu nesne veya nesnelerde hareket gerçekleşme olasılığıdır.

Görüntülerin doğruluğu kontrol edilirken, değişim noktalarının alan değerleri istenen aralıkta kalabilir. Böyle bir durumda, Şekil 4.15'deki görüntülerde birbirini kesen iki doğru parçasına ek olarak başka noktalarda görülür ve görüntülerdeki doğru parçalarının başlangıç ve bitiş noktaları hatalı bulunur. Başlangıç ve bitiş noktalarının hatalı bulunması sonucunda iki görüntüden bulunan iki ayrı kesişim noktası birbirinden çok farklı çıkar.

Çizelge 4-4'de görülen kesişim noktaları Eşitlik 4.13'deki yakınlık tanımlamasına göre birbirine yakındır. Eğer kesişim noktaları birbirine yakın çıkmazsa arayüzün bilgi penceresinde orta nokta bulunurken hata gerçekleşti bilgisi verilir ve program, otomatik olarak saat oku düğmesine 1. durumda basılmış gibi davranır.

Kesişim noktaları birbirine yakın kabul edilse de birbirine çakışık çıkmadığı için kesişim noktalarından hangisinin saatin orta noktasına yakın olduğu belirsizdir. İskelet çıkarma algoritmasının ideale yakın sonuç vermesine rağmen ideal sonuç vermediği için de doğru parçalarının başlangıç ve bitiş noktalarında ufak sapmalar meydana gelebilir. Bu nedenle noktaları birbirine yakın çıksa da orta noktadan sapma gösterebilir. Bu sorunlara 4 farklı konumdaki saniye kollarının iskeletleri ayrı ayrı ele alınarak çözüm bulunmuştur. Daha önceden belirtildiği gibi her bir iskelet iki ayrı parçadan oluşmaktadır ve Şekil 4.17(a) ve Şekil 4.17(b)'de görüldüğü gibi bulunan kesişim noktaları iskeletlerin iki ayrı parçasının ortasında kalan bölgededir. Kesişim noktalarının bu özelliği kullanılarak ve aşağıdaki adımlar izlenerek saatin orta noktası bulunmuştur:

1. **Adım:** Her iskelet için ayrı ayrı 1. kesişim noktası $(x_1, y_1)$'e öklidyen uzaklığı en küçük olan iskelet noktası $(x_m, y_m)$ bulunur. Bulunan noktayı, 1. kesişim noktası ile birleştiren doğru parçasının, 1. kesişim noktası orijini olan ve yatay ekseni görüntü düzleminin yatay eksenine paralel olan koordinat sisteminin yatay eksenine göre pozitif açısı $\alpha_1$ Eşitlik 4.14 kullanılarak hesaplanır.

$$\alpha = \arctan\left(\frac{y'-y}{x-x'}\right) \qquad (x', y') \textit{ merkez nokta} \qquad (4.14)$$

2. **Adım:** Her iskelet için ayrı ayrı 1. kesişim noktası $(x_1, y_1)$'e öklidyen uzaklığı en küçük olan ve 1. kesişim noktası ile birleştiren doğru parçasının, 1. adımda tanımlanan koordinat sisteminin yatay eksenine göre pozitif açısı $\alpha_2$'nin $\alpha_1$'le arasındaki açı farkının 170° ve 190° arasında olduğu ikinci bir iskelet noktası $(x_n, y_n)$ bulunur.

3. **Adım:** Dört iskelet için ayrı ayrı bulunan $(x_n, y_n)$ ve $(x_m, y_m)$ noktaları arasındaki öklidyen uzaklıklar bulunur.

4. **Adım:** Öklidyen uzaklığı en küçük olan $(x_n, y_n)$ ve $(x_m, y_m)$ noktalarının orta noktasının koordinatları Eşitlik 4.15 kullanılarak bulunur. Bu nokta aynı zamanda saatin konumunun birinci bileşeni olan saatin orta noktasıdır.

$$x_o = \frac{x_m + x_n}{2}$$
$$y_o = \frac{y_m + y_n}{2} \qquad (4.15)$$

Dört farklı konumdaki saniye kollarının iskeletleri için bulunan $(x_n, y_n)$ ve $(x_m, y_m)$ noktalarının koordinatları ve noktaların arasındaki öklidyen uzaklıklar Çizelge 4-5'de görülmektedir.

**Çizelge 4-5** $(x_n, y_n)$ ve $(x_m, y_m)$ noktalarının koordinatları ve noktalar arasındaki uzaklılar

| | $x_n$ | $y_n$ | $x_m$ | $y_m$ | **Öklidyen Uzaklıklar** |
|---|---|---|---|---|---|
| **1. İskelet** | *174* | *137* | *154* | *125* | *23 piksel* |
| **2. İskelet** | *172* | *140* | *156* | *122* | *24 piksel* |
| **3. İskelet** | *173* | *140* | *155* | *123* | *24 piksel* |
| **4. İskelet** | *170* | *142* | *158* | *121* | *24 piksel* |

Çizelge 4-5'deki değerlere göre aralarındaki öklidyen uzaklık en küçük olan, 1. iskelette buluna $(x_n, y_n)$ ve $(x_m, y_m)$ noktalarıdır. Eşitlik 4.15 kullanılarak saatin orta noktasının koordinatları

$$x_o = 164$$
$$y_o = 131 \qquad (4.16)$$

olarak bulunmuştur. Orta nokta bulunduktan sonra saatin konumunun ikinci bileşeni olan saniye kolunun orta noktaya en uzak noktasının orta noktaya olan öklidyen uzaklığı, orta nokta ve dört farklı konumdaki saniye kollarının iskeletleri kullanılarak bulunur. Her doğru parçası (iskelet) için Çizelge 4-3'deki başlangıç ve bitiş noktalarından orta noktaya en uzak olanının, orta noktaya olan uzaklığı bulunur. Dört iskelet için ayrı ayrı bulunan uzaklık değerleri birbirleriyle karşılaştırılır ve en büyük olan uzaklık, saatin konumunun ikinci bileşeni $R$ olarak belirlenir. Dört farklı konumdaki saniye kollarının iskeletleri kullanılarak bulunan uzaklık değerleri Çizelge 4-6'da gösterilmiştir.

**Çizelge 4-6** Orta noktaya en uzak olan iskelet noktalarının orta noktaya olan uzaklıkları

| | **Öklidyen Uzaklıklar** |
|---|---|
| **1. İskelet** | *96 piksel* |
| **2. İskelet** | *98 piksel* |
| **3. İskelet** | *97 piksel* |
| **4. İskelet** | *97 piksel* |

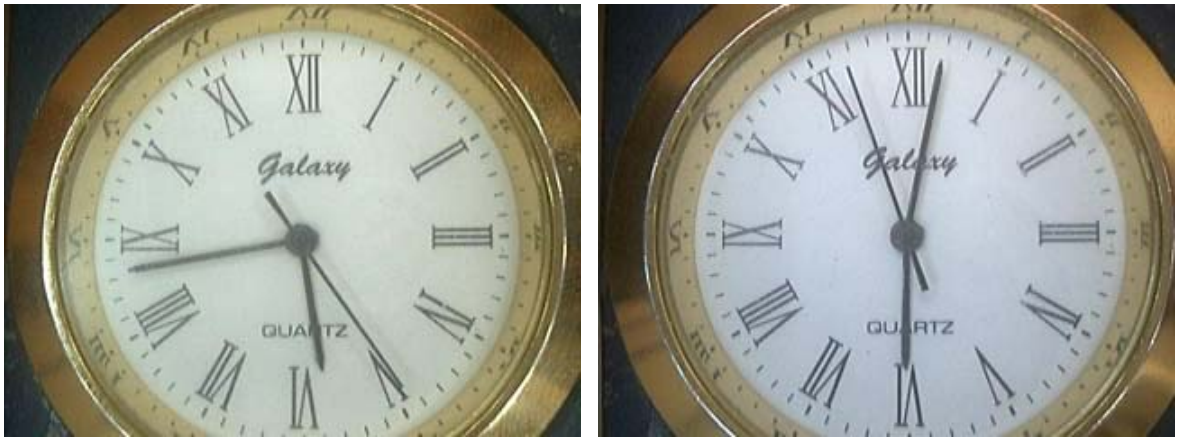
Çizelge 4-6'daki değerlere göre orta noktaya en uzak iskelet noktası 2. iskelettedir ve saatin konumun ikinci bileşeni $R = 98$ olarak bulunmuştur.

Gerçek zamanlı çalışma açısından, saatin konumu bulunduktan sonra görüntüler üzerinde uygulanan algoritmalar, görüntü içindeki $(x_o, y_o)$ noktası merkezi olan ve $R$ çember içinde kalan görüntü parçasında uygulanmaktadır.

## 4.5. Saatin Konumunun Kontrolü

Saati oku düğmesine 2. durumda basılmasında, saatin kaç olduğunun anlaşılması işleminden önce saatin konumunda değişiklik olup olmadığının kontrolü gerekmektedir. Saatin konumunda değişiklik olup olmadığının belirlenmesinde 2. durumda saat oku düğmesine basılmasıyla yakalanan iki yeni

görüntüden ilki ve saatin kaç olduğunun anlaşılmasında kullanılan en son görüntü kullanılmaktadır. Saatin konumun kontrolünün nasıl yapıldığını göstermede kullanılan örnek iki görüntü Şekil 4-18'de görülmektedir. İki görüntüde arasında saatteki konum değişikliği gözle zor fark edilebilmektedir.




**(a)** **(b)**

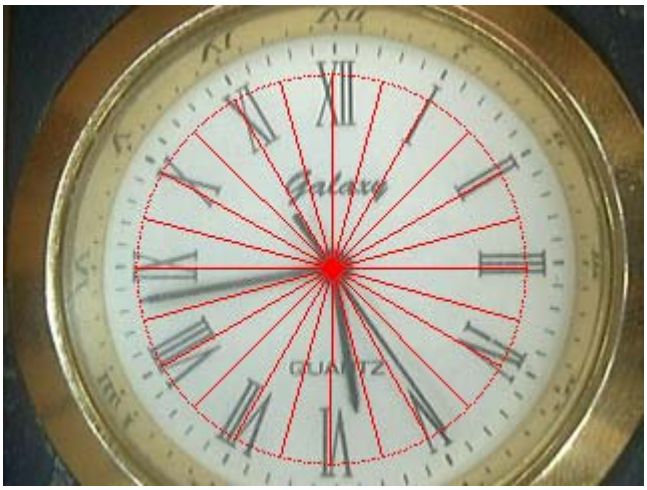
**Şekil 4-18 (a)** 1. durumda yakalan 4 görüntüden sonuncusu
**(b)** 2. durumda yakalan 2 görüntüden birincisi

Saatin konumunda değişik olup olmadığının belirlenmesinde kollarla birlikte, saat göstergesi içinde kalan kollar dışındaki göstergenin arka planı ile karşıtlığı farkı olan rakam, yazı ve simge gibi unsurlar kullanır. Bu unsurların konumunda değişiklik olup olmaması, saatin konumunun değişip değişmemesinin belirlenmesinde belirleyicidir. Saatin konumunun değişip değişmediği aşağıdaki adımlar izlenerek belirlenir:

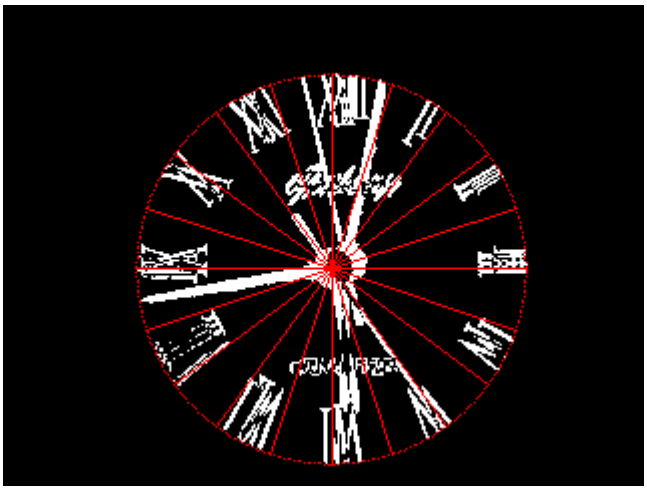
**1.** Şekil 4-18'deki görüntüler kullanılarak Eşitlik 3.10'da ifade edilen fark görüntüsü bulunur. Bu işlemdeki fark görüntüsü bulunurken kullanılan eşik değeri büyük önem taşımamaktadır. Önemli olan nokta, gerçek zamanlı çalışma açısından işlem süresini artırmadan kabul edilebilir eşik değeri bulmaktır. Fark görüntüsünü bulurken kullanılan eşik değeri için $T = 40$ uygun görülmüştür.

**2.** Bulunan fark görüntüsünde, $(x_o, y_o)$ noktası merkezi olan, $R$ yarıçaplı çember eşit dilimlere bölünür ve dilimlerin içinde kalan değişim noktaları bulunur. Bu çalışmada 24 eşit dilim uygun görülmüştür. Dilim sayısı 24 seçilerek Şekil

4-19(a)'da görüldüğü her iki dilimde bir dilimi ayıran sınırların, saat göstergesinde kollar dışındaki diğer unsurların (romen rakamlarının) ortasından geçmesi sağlanmıştır. Bu nedenle saatin konumu değiştiği zaman fark görüntüsünde bütün dilimlerde değişim noktasının olması amaçlanmıştır. Çemberin (saatin göstergesinin) dışında kalan değişimler dikkate alınmamıştır. Fark görüntüsünün üstüne 24 eşit dilime bölünmüş çemberin oturtulmuş hali Şekil 4-19(b)'de gösterilmiştir.



**(a)**



**(b)**

**Şekil 4-19 (a)** Şekil 4-18(a)'daki görüntünün üzerine çember oturtulmuş hali
**(b)** Fark görüntüsünün üzerine çember oturtulmuş hali

**3.** Dilimlerin içinde kalan noktalar bulunduktan sonra kaç dilimde değişim noktası olduğu hesaplanır. Değişim olan dilimlerin sayısı 15'i geçerse, saatin

konumunda değişiklik olduğu kabul edilir ve arayüzün bilgi penceresinde saatin konumunda değişiklik olduğu algılandı bilgisi verilir ve program, otomatik olarak saat oku düğmesine 1. durumda basılmış gibi davranır. Değişim olan dilimlerin sayısı 15'in altında kalırsa saatin kaç olduğunun anlaşılması algoritmasının ikinci aşaması saati okumaya geçilir.

15 değerinin belirlenmesinin nedeni: Saat göstergesi içinde 3 ayrı kol olduğu bilinmektedir. Saati oku düğmesine, saatin kaç olduğu anlaşıldıktan sonra ne kadar sürede basılacağı bilinemeyeceğinden 3 kolun da hareket etme olasılığı vardır. Bir kol hareket ettiği zaman saatin konumu değişmeden, değişim noktasının olabileceği dilim sayısı en büyük 4'tür. Göstergede 3 kol olduğu için, saatin konumu değişmeden değişim noktasının olabileceği dilim sayısı en büyük 12'dir. 3 dilim de güvenlik dilimi olarak eklenerek 15 dilim sayısı sınır olarak belirlenmiştir. Şekil 4-19(b)'daki fark görüntüsünde değişim noktalarının bulunduğu dilim sayısı 20 olarak hesaplanmıştır.

## 4.6. Saati Okuma

Saatin kaç olduğunun anlaşılma algoritmasının ikinci aşaması analog göstergeli saatin okunmasıdır. Analog göstergeli saatin işaretçileri dairesel hareket eden akrep, yelkovan ve saniye kolu olduğu ve üç adet de şekli daire olan ölçeği olduğu Bölüm1'de belirtilmiştir. Fakat saniye ve yelkovan kollarının ölçeği aynı kabul edilebilir. Saniye ve yelkovan kolları için $6°$'lik dönme $1$ birimlik değişime denk gelir, sadece kolların birimleri farklıdır. Saniye kolu için $1$ birim $1$ saniyedir, yelkovan kolu için $1$ birim $1$ dakikadır. Akrep kolu için $30°$'lik dönme $1$ birimlik değişime denk geldiği için kolun ölçeği farklı kabul edilebilir. Akrep kolun için de $1$ birim $1$ saattir.

Saatin orta noktası $(x_o, y_o)$ merkez pikseli olan, $R$ yarıçaplı çemberin (saat göstergesinin) içinde kalan akrep, yelkovan ve saniye kolları (işaretçileri) Bölüm 3'de ayrıntılı olarak bahsedilen nesne tanıma analizi ile tanınarak, saat, dakika ve saniyenin kaç olduğu işaretçileri ve ölçekleri kullanılarak hesaplanır.

Saati okumak için sadece saniye kolunun hareketini içeren iki görüntü gerekmektedir. Eğer saati oku düğmesine 1. durumda basılmış ise saatin konumun bulunmasında kullanılmak üzere yakalan dört görüntüden son iki tanesi saatin okunmasında da kullanılır. Eğer saati oku düğmesine 2. durumda basılmış ise bir saniye farkla yakalanan iki yeni görüntü, saatin konumun kontrolü yapıldıktan sonra saatin okunmasında kullanılır.

Saatin okunmasında aşağıdaki adımlar izlenmiştir:

1. **Kolların (İşaretçilerin) Bölütlenmesi:** Bir saniye fark ile yakalanmış iki görüntü kullanılarak akrep, yelkovan ve saniye kolları görüntüden bölütlenir. Saniye kolunun tanınması bu adım sonunda gerçekleşmektedir. Görüntüler arasında, gerçek zamanlı çalışma açısından, saniye kolunun hareket ettiği en kısa süre olan bir saniyelik fark olması yeterli görülmüştür.

2. **Kolların Sayısının ve Açılarının Bulunması:** Kolların çakışma durumuna göre, görüntüden bölütlenmiş olan tek kol (üç kol üst üste), iki ayrı kol (iki kol üst üste üçüncü kol ayrı) veya üç ayrı kol olma olasılığı vardır. Bölütlenen bölgelerden öznitelikler çıkarılarak, kolların sayısı ve açıları bulunur.

3. **Kolların Tanınması :** Kolların sayısına göre yeni öznitelikler çıkarılır, ve çıkarılan öznitelikler kullanılarak akrep ve yelkovan kolları tanınır.

4. **Saat, Dakika ve Saniye'nin Hesaplanması:** Tanınan kollar kullanılarak saat, dakika ve saniye hesaplanır.

### 4.6.1. Kolların (işaretçilerin) bölütlenmesi

Kolların bölütlenmesinde kullanılacak yöntemi seçmek için saat görüntüsünde görsel analiz yapılmıştır. Görsel analize elde edilen sonuçlar şunlardır:

- Saniye kolu bir saniyede gözle algılanabilecek şekilde konum değiştirmektedir. Akrep ve yelkovan kolunun gözle algılanabilecek şekilde konum değiştirmesi uzun sürmektedir.

- Akrep, yelkovan ve saniye kolları çok yakın gri tondalar ve bütün haldedir.

- Kollar ile saat göstergesinin arka planı arasında karşıtlık farkı yüksektir.

- Saatin göstergesi içinde kolların dışında, kolların gri seviye değerlerine yakın gri seviye değerlerine sahip rakam, yazı ve simge gibi unsurların mevcuttur.

Yukarıdaki analiz sonuçlarına göre, saniye kolunun bölütlenmesinde, Bölüm 3.1.3'de anlatılan harekete dayalı bölütleme yaklaşımının kullanılmasına karar verilmiştir. Bu yaklaşımın kullanılması için saniye kolunun bir saniyelik süredeki hareketi sonucunda iki farklı konumda olduğu bir saniye farkla yakalanmış iki görüntü gerekmektedir.

Akrep ve yelkovan kollarının bölütlenmesinde, Bölüm 3.1.2.1'de anlatılan bölgesel bölütlemeye dayanan S.R.G. algoritmasının kullanılmasına karar verilmiştir. Eşikleme yönteminin kullanılmamasının başlıca sebebi pikseller arasındaki uzamsal ilişkinin göz ardı edilmesidir. Eşikleme yöntemi tek başına kullanıldığı düşünülürse, saatin göstergesi içinde bulunan kollara yakın gri seviyedeki diğer unsurlarında bölütlenme olasılığı yüksektir. Kenar tabanlı bölütleme yöntemlerin kullanılmamasının başlıca sebebi ise bölütleme işlemi sonucunda kolların sadece sınır piksellerinin elde edilmesi ve yine kollar dışındaki unsurların bölütlenme olasılığının yüksek olmasıdır. S.R.G. algoritmasının uygulanması için bir görüntü yeterlidir.

#### 4.6.1.1. Saniye kolunun bölütlenmesi

Saniye kolunun bölütlenmesinde kullanılacak görüntüler Şekil 4-10(c) ve Şekil 4-10(d)'deki saatin konumunun bulunmasında kullanılan son iki görüntüdür. Gerçek zamanlı çalışma açısından işlem zamanını artırmamak için, iki görüntünün

kullanılmasıyla elde edilen Şekil 4-14(b)'deki yumuşatılmış ve eşiklenmiş fark görüntüsü, saniye kolunun bölütleme işleminde değişiklik yapılmadan kullanılmıştır.

Eğer saatin okunmasına, saati oku düğmesine 2. durumda basılmasıyla geçilmiş işe yeni yakalan iki görüntü kullanılarak, Şekil 4-14(b)'deki görüntünün bulunmasında izlenen adımların aynısı izlenir ve yumuşatılmış ve eşiklenmiş fark görüntüsü elde edilir.

Şekil 4-14(b)'deki görüntüde saniye kolu, siyah kol ve beyaz kol olarak iki farklı konumda bulunmaktadır. Saniye kolunun bölütlenmesi ve tanınması iki farklı konumdaki koldan, saat yönünde ilerde olanın bulunmasıyla gerçekleşir. Saat yönünde ilerde olan kolun rengi önemli bir bilgi daha verir. Eğer siyah kol ilerdeyse, Şekil 4-14(b)'deki fark görüntüsünün bulunmasında kullanılan görüntülerdeki saatin kolları, saatin göstergesinin arka planından koyudur, eğer beyaz kol ilerdeyse saatin göstergesinin arka planı, kollardan koyudur.

Saat yönünde ilerde olan kol aşağıdaki adımlar izlenerek bulunmuştur:

**1.** Kolların inceltme algoritmasıyla inceltilerek iskeletleri elde edilir. Kolların iskeletleri, $(x_o, y_o)$ noktasının bulunma işlemi sırasında daha önceden elde edilmiştir.

**2.** İskeletler üzerinde bulunan ve $(x_o, y_o)$ noktasına öklidyen uzaklığı en büyük olan noktalar bulunur.

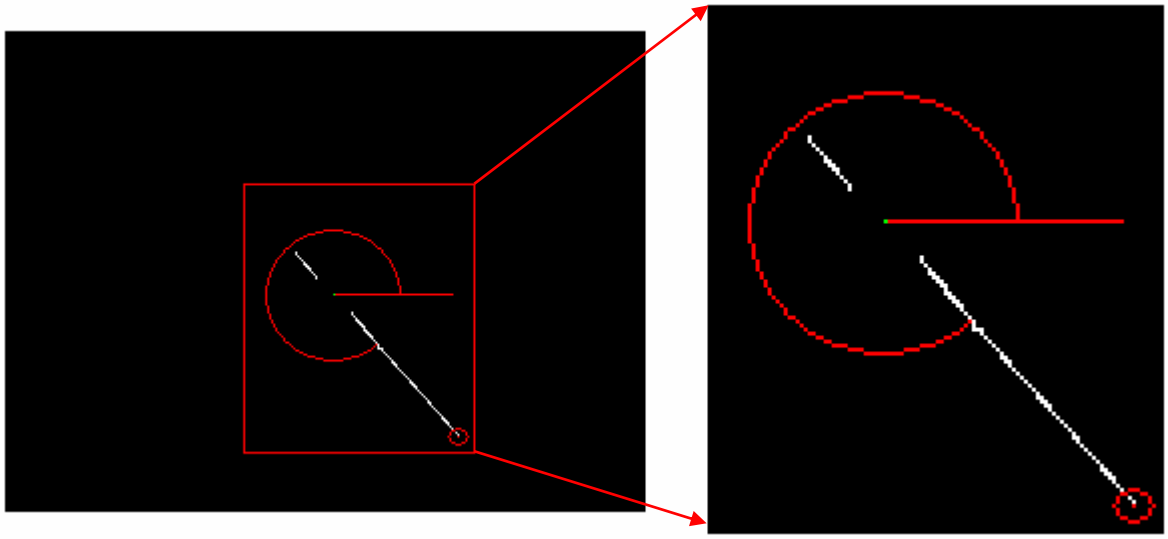
**3.** Bir önceki adımda bulunan noktalar ve $(x_o, y_o)$ noktası Eşitlik 4.14'deki denkleme uygulanarak, iki kolun, $(x_o, y_o)$ noktası orijini olan ve yatay ekseni görüntü düzleminin yatay eksenine paralel olan koordinat sisteminin yatay eksenine göre pozitif açıları bulunur.

**4.** Kolların açıları kullanılarak, saat yönünde ilerde olan kol algılanır. İlerde olan kolun rengine göre, saatin göstergesinin arka planın mı yoksa kolların mı koyu olduğu belirlenir.

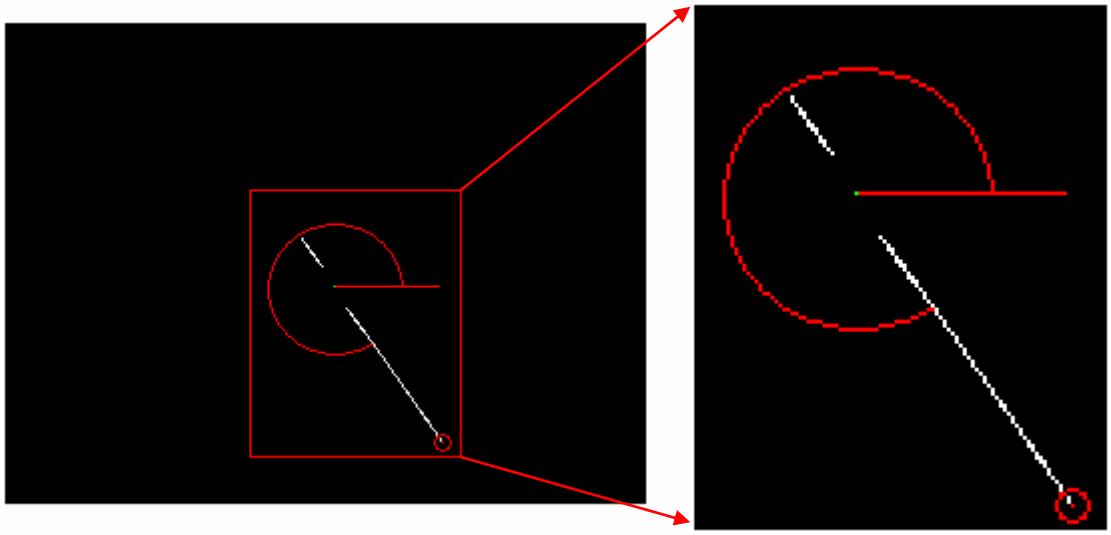
Kolların iskeletleri üzerindeki $(x_o, y_o)$ noktasına en uzak olan noktaları ve açıları Şekil 4-20'de görülmektedir. Noktaların ve açıların değerleri de Çizelge 4-7'de görülmektedir. Bu değerlere göre siyah kol saat yönünde ilerde olduğu için, saat göstergesinin arka planının kollardan koyu olduğu anlaşılmıştır, son görüntüdeki saniye kolunun Şekil 4-14(b)'deki görüntüdeki siyah kol olduğu ve saniye kolunun açısının $(\theta_s)$ $304.69°$ olduğu belirlenmiştir.

**Çizelge 4-7** En uzak nokta koordinatları ve açı değerleri

| | **En Uzak Noktaların Koordinatları** | | **Yatay Eksene Göre Açılar** |
|---|---|---|---|
| | ***x*** | ***y*** | |
| **3. İskelet (beyaz kol iskeleti)** | *226* | *202* | *311.13°* |
| **4. İskelet (siyah kol iskeleti)** | *218* | *209* | *304.69°* |



**(a)** Beyaz (3.) kolun iskeleti



**(b)** Siyah (4.) kolun iskeleti

**Şekil 4-20** İskeletler üzerindeki en uzak noktalar ve iskeletlerin açıları

Diğer kolların bölütlenmesine devam etmeden önce beyaz ve siyah saniye kollarının (iskeletlerinin) yatay eksenle yaptıkları açılar arasındaki farkın kabul edilebilir bir değerde olup olmadığı kontrol edilir. Bu kontrolün sebebi: son iki görüntüdeki saniye kollarının herhangi birinin yelkovan koluyla tamamen çakışması durumunda, saniye kolunun bölütlenmesinde ve tanınmasında hataya yol açmasıdır. Eğer böyle bir çakışma gerçekleşirse, iki görüntü kullanılarak elde edilen yumuşatılmış ve eşiklenmiş fark görüntüsünde saniye kolu tek konumda bulunur. Tek bir konumdaki saniye kolu (siyah veya beyaz olabilir) kullanılarak,

saniye kolunun olduğu görüntü ve saat göstergesinin arka planın mı yoksa kolların mı daha koyu olduğu belirlenemez.

Kontrol işleminde, iki kol arasında 1 saniyelik fark olması gerektiği için açısal olarak yaklaşık $360°/60 = 6°$'lik fark olması beklenmektedir. $3°$'lik güvenlik farkı da eklenerek eğer iki kol arasındaki açı farkı $9°$'den büyük ise arayüzün bilgi penceresinde saniye bulmada hata gerçekleşti bilgisi verilir ve program, otomatik olarak saat oku düğmesine 2. durumda basılmış gibi davranır. Eğer açı farkı $9°$'den küçük ise kolların bölütlenmesi işlemine devam edilir. Çizelge 4-7'deki açı değerlerine göre açı farkı $6.44°$ olarak hesaplanmıştır ve bu değere göre akrep ve yelkovan kollarının bölütleme işlemine geçilmiştir.

#### 4.6.1.2. Akrep ve yelkovan kollarının bölütlenmesi

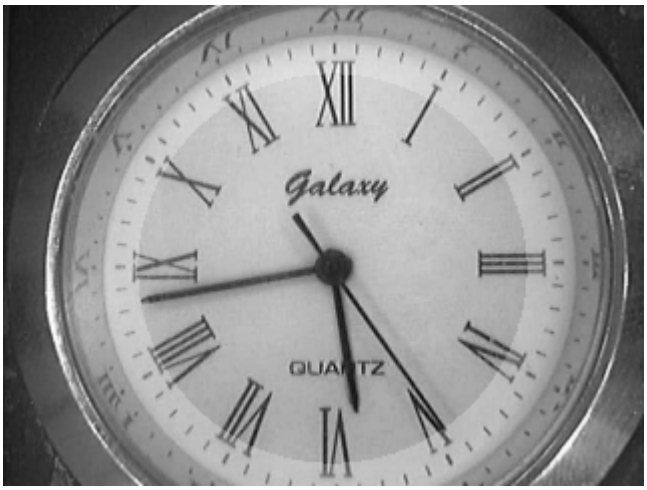
Akrep ve yelkovan kollarının bölütlenmesinde kullanılacak, S.R.G. algoritmasının uygulanacağı görüntü, Şekil 4-10(d)'deki saatin konumunun bulunmasında kullanılan son görüntüdür.

S.R.G. algoritması kullanılarak bölütlenecek bölge sayısı bir olarak uygun görülmüştür. Uygun görülen bölge, üç farklı kolu ve kolların kesiştikleri noktadaki küçük çaplı çemberi içine alacak bölge olarak belirlenmiştir. S.R.G algoritmasının seçimindeki temel neden bu tanımlanan bölgenin, göstergenin arka planına göre karşıtlık farkının yüksek olması ve piksellerin uzamsal ilişkilerinin olmasıdır. S.R.G algoritmasının girdileri olan tohum noktaları ve benzerlik kriteri belirlenmeden önce karşıtlık farkının önemli bir unsur olmasından dolayı artırılması sağlanmıştır.

S.R.G. algoritmasının uygulanacağı Şekil 4-10(d)'deki görüntüye Bölüm 2.5'de anlatılan karşıtlık artırma dönüşümünü uygulanmıştır. Bu tez çalışmasında kullanılan karşıtlık artırma fonksiyonunda $(r_1, s_1) = (r_{endüşük}, 0)$ ve $(r_2, s_2) = (r_{enbüyük}, 255)$ olarak seçilmiştir. Seçilen noktalardaki $r_{endüşük} = 66$ giriş görüntüsünde saat göstergesinin içindeki en düşük gri seviye, $r_{enyüksek} = 255$ giriş görüntüsünde saat göstergesinin içindeki en yüksek gri seviye değerleri olarak bulunmuştur. Dönüşüm fonksiyona göre giriş görüntüsünde saat göstergesinin içindeki bölümün gri seviye

aralığı *66* ile *255* aralığından, gri seviyelerin sırası bozulmadan, doğrusal olarak *0* ile *255* değerleri arasına arttırılmıştır.

Gerçek zamanlı çalışma açısından karşıtlık artırma dönüşümü uygulanmadan önce giriş görüntüsündeki göstergenin içinde kalan bölümün gri seviye aralığı kontrol edilir. Gereksiz işlem süresi harcamamak için *[0, 255]* aralığında olan görüntülerde dönüşüm uygulanmaz. Şekil 4-10(d)'deki görüntünün karşıtlık artırma dönüşümü uygulanmış hali Şekil 4-21'de görülmektedir.

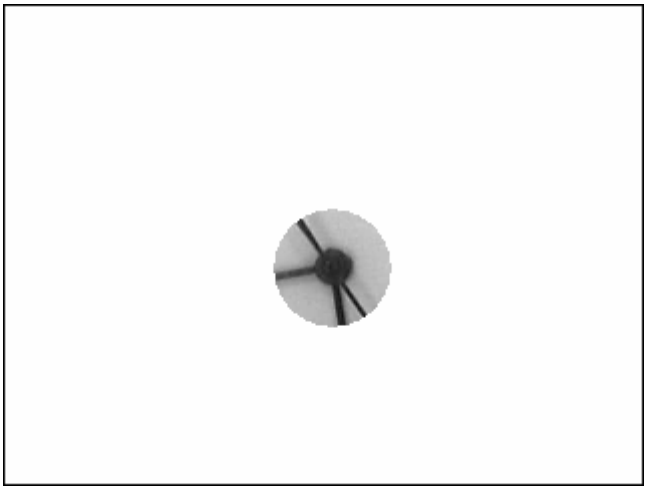



**Şekil 4-21** Şekil 4-10(d)'deki görüntünün karşıtlık artırma uygulanmış hali

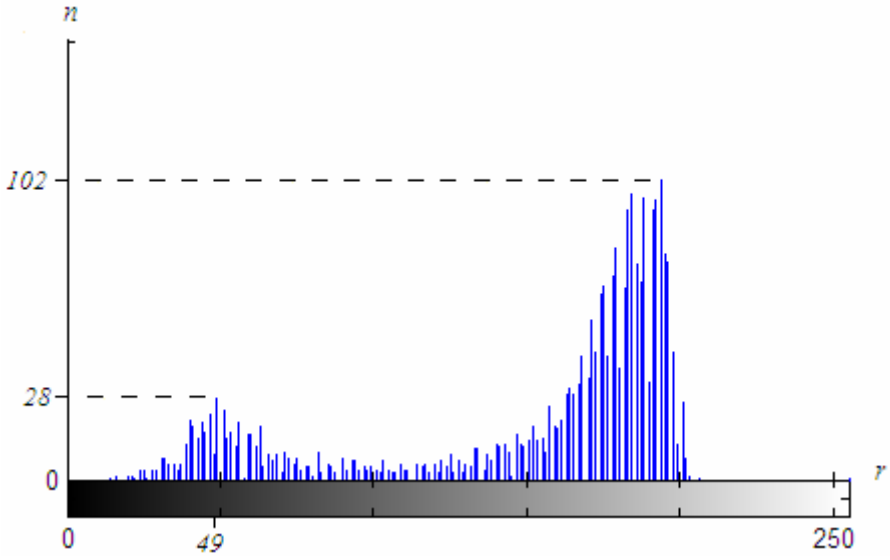
S.R.G algoritmasının girdileri olan tohum noktaları ve benzerlik kriteri seçiminin otomatik olarak yapılması için Şekil 4-21'deki görüntüde tohumların seçileceği ve benzerlik kriterinin elde edileceği görüntü parçasının alanının histogramı çıkarılmıştır. Bu parça, $(x_o, y_o) = (164, 131)$ merkez pikseli olan, $R*3/10 = 98*3/10$ yarıçaplı çemberin içinde kalmaktadır. Bu görüntü parçası Şekil 4-22(a)'da ve histogramı da Şekil 4-22(b)'de görülmektedir. Bu parçanın seçilmesinin iki temel nedeni vardır:

1. Histogramın tüm saat göstergesinden çıkarmak yerine daha küçük bir görüntü parçasından çıkarılarak, tohumların ve benzerlik kriterinin seçiminin süresinin azalması sağlanmıştır.

2. İki doruklu olan histogramda, benzerlik kriterinin seçiminde kullanılacak eşik değeri bulma yönteminden başarılı bir sonuç almak için dorukların boyutlarının birbirine yakın olması sağlanmıştır.



**(a)**




**(b)**

**Şekil 4-22 (a)** Görüntü parçası **(b)** Görüntü parçasının histogramı

Tohum noktalarının seçimi için iki doruğun birbirinden ayrılması gerekmektedir. Uygun eşik değerinin seçimi için Bölüm 3.1.1'de anlatılan birinci eşik değeri bulma algoritması kullanılmıştır. Algoritmanın girdisi olan ve ilk başta belirlenmesi gereken keyfi eşik değeri gri seviyelerin ortalaması olarak seçilmiştir. Çizelge 4-8'de algoritmanın kaç dürümde uygun eşik değerine ulaştığı ve her dürümde bulunan eşik değerleri görülmektedir.

**Çizelge 4-8** Eşik değerleri

| | |
|---|---|
| **Başlangıç Eşik Değeri:** | *151.61* |
| **1. Dürüm Sonunda Bulunan Eşik Değeri:** | *131.18* |
| **2. Dürüm Sonunda Bulunan Eşik Değeri:** | *127.17* |
| **3. Dürüm Sonunda Bulunan Eşik Değeri:** | *125.39* |

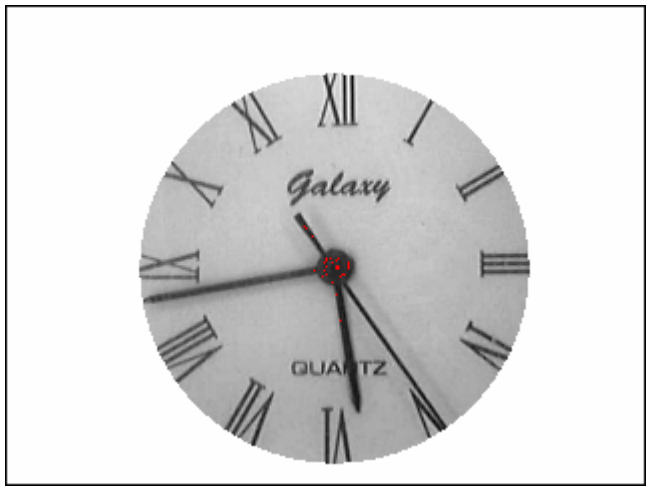
3. dürüm sonunda iki doruğu ayıran eşik değeri 125 olarak bulunmuştur. Daha önceden saat göstergesinin arka planının kollara göre daha açık olduğu belirlendiği için, eşik değerinden daha küçük olan değerler içerisinde en sık gözüken değere sahip olan noktalar kök nokta olarak seçilmiştir. Bu değer Şekil 4-22(b)'deki histogram üzerinde eşik değerden daha küçük olup, en büyük değerli gri seviye olan 49 değeridir. Tohum noktaları yani 49 gri seviye değerine sahip noktaların olduğu, S.R.G algoritmasının başlangıç görüntüsü Şekil 4-23'de görülmektedir.

**Şekil 4-23** S.R.G. algoritmasının başlangıç görüntüsü

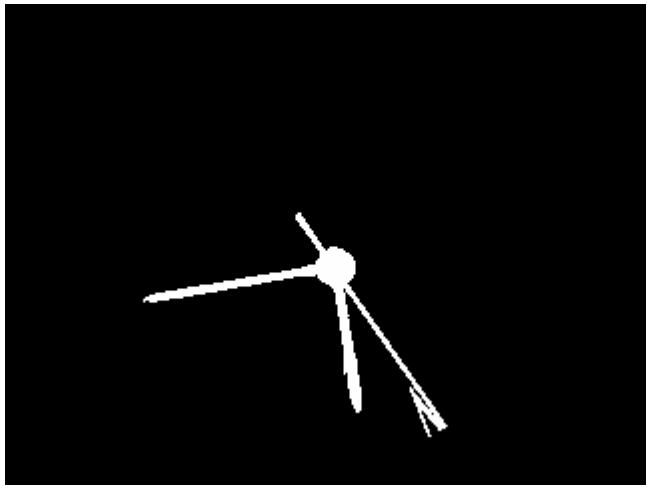
Bölgeye eklenecek noktaların gri seviye değerlerinin aralığı yani benzerlik kriteri, 0 ile histogramdaki iki doruğu ayıran eşik değeri olan 125 değerinin aralığı olarak belirlenmiştir.

S.R.G. algoritmasının girdileri belirlendikten sonra algoritma Şekil 4-24(a)'daki saatin göstergesinin bulunduğu görüntü parçasına kök noktalarından başlanarak uygulanmıştır. Görüntüde kök noktaları kırmızı ile gösterilmiştir. S.R.G

algoritmasının çıkış görüntüsü Şekil 4-24(b)'de gösterilmiştir. Beklendiği gibi çıkış görüntüsünde tek bir bölge vardır.



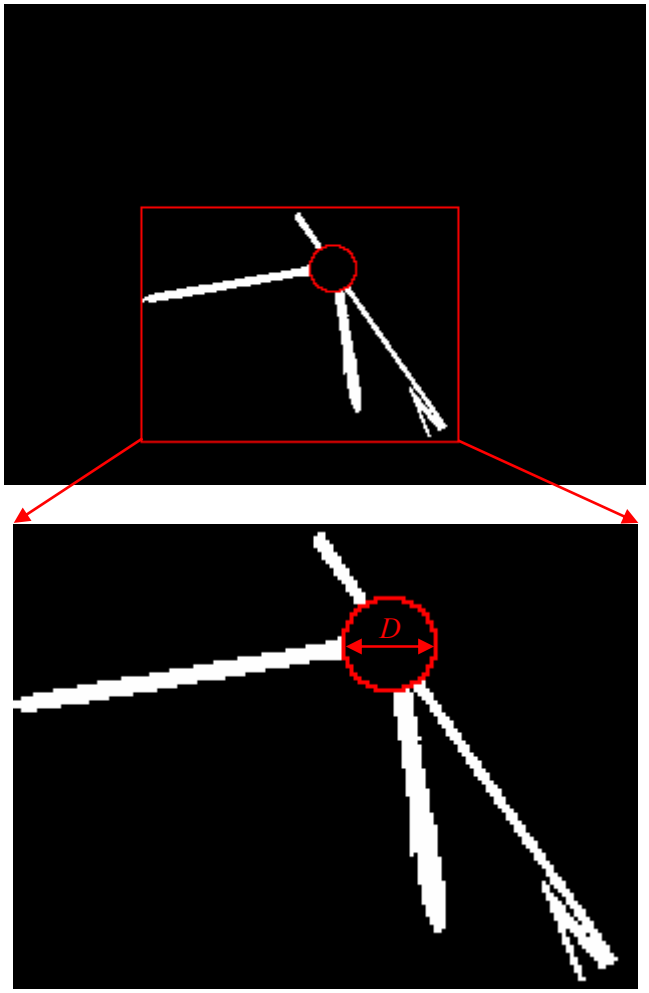

**(a)**



**(b)**

**Şekil 4-24 (a)** S.R.G. algoritmasının uygulandığı görüntü
**(b)** S.R.G. algoritmasının çıkış görüntüsü

Şekil 4-24(b)'deki görüntüde kollar bir bütün halindedir. Kolları birbirinden ayırmak için bölütleme aşamasında son bir işlem yapılmıştır. Çizelge 4-5'deki öklidyen uzaklıklardan en küçük olanının çapı ($D=23$) piksel olduğu ve merkez noktası da $(x_o, y_o) = (164, 131)$ olan çemberin içinde kalan noktalar, S.R.G. algoritmasının uygulanmasıyla elde edilen bölgeden çıkarılmıştır. Son işlemin uygulanmasıyla elde edilen bölgeleri içeren görüntü Şekil 4-25'de görülmektedir. Görüntüden çıkarılan noktaların içinde kaldığı çember kırmızı ile gösterilmiştir.




**Şekil 4-25** Bölütleme işlemi sonucunda elde edilen bölgeler

### 4.6.2. Kol sayısının ve açılarının bulunması:

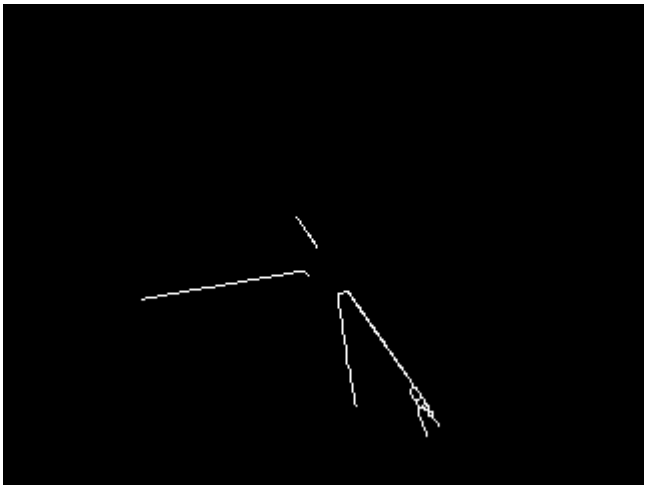
Kolların çakışma durumunun aşağıda özetlendiği gibi 5 olasılığı vardır,

1. Bütün kolları birbirinden ayrıdır.
2. Bütün kollar çakışmıştır.
3. Saniye kolu ayrıdır, akrep ve yelkovan kolu çakışmıştır.
4. Akrep kolu ayrıdır, saniye ve yelkovan kolları çakışmıştır.
5. Yelkovan kolu ayrıdır, saniye ve akrep kolları çakışmıştır.

Yukarıdaki 5 olasılığa göre bölütleme işlemi sonunda kol sayısı en fazla üç en az bir olabilir. Tek kol var ise üç kol çakışmıştır. İki kol var ise üç koldan ikisi çakışmıştır diğer kol ayrı konumdadır. Üç kol var ise bütün kollar ayrı konumdadır.

Kolların tanınabilmelerinde belirleyici olan özniteliklerin çıkarılması için, kolların, Şekil 4-25'deki bölütlenmiş görüntüdeki sayılarının ve açılarının bulunması gerekir. Bölütlenmiş kol sayısının ve kol açılarının bulunması için aşağıdaki adımlar izlenmiştir:

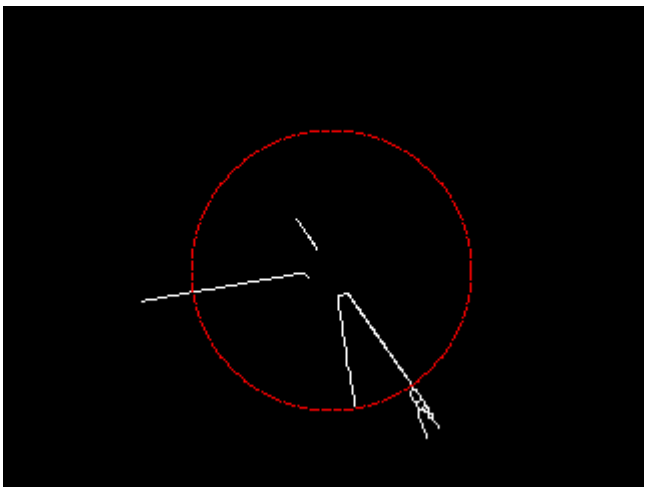
**1.** Bölütleme işlemi sonunda elde edilen görüntü inceltme algoritması inceltilerek iskeleti çıkarılır. Şekil 4-25'deki görüntünün iskeleti Şekil 4-26'da gösterilmiştir.



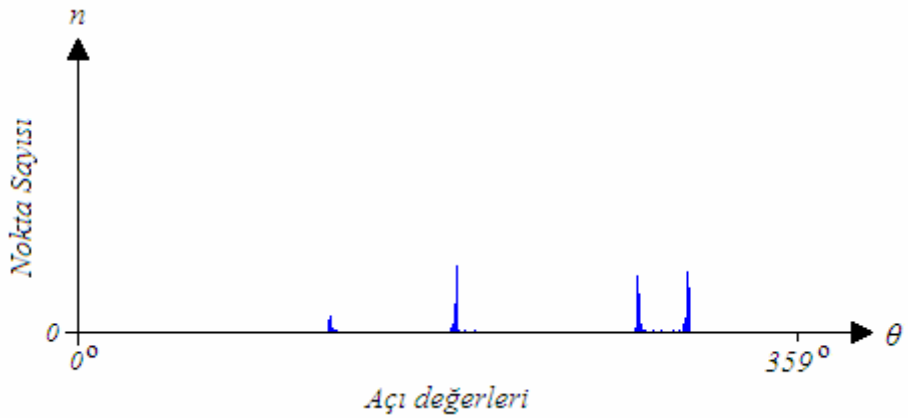
**Şekil 4-26** Şekil 4-25'daki görüntünün iskeleti

**2.** Bir önceki adımda çıkarılan iskeletin imza gösterimi yapılır: Bu imza gösteriminde, Bölüm 3.2.2.1'de anlatılan imza gösteriminden farklı olarak, iskelet noktalarının sayısı, iskelet pikselini $(x_o, y_o)$ noktası ile birleştiren doğru parçasının eğiminin fonksiyonu olarak çizilir. Eğim, doğru parçasının, $(x_o, y_o)$ noktası orijini olan ve yatay ekseni görüntü düzleminin yatay eksenine paralel olan koordinat sisteminin yatay eksenine göre pozitif açısıdır. İmza gösterimi yapılırken, merkez noktası $(x_o, y_o)$ olan ve $0.72*R$ yarıçaplı çember içinde kalan iskelet noktaları kullanılır. Çemberin Şekil 4-26'daki görüntünün üzerine oturtulmuş hali Şekil 4-27 görülmektedir. $0.72$ değeri yaklaşık olarak saat göstergelerindeki akrep ve saniye kollarının $(x_o, y_o)$ noktasına en uzak noktaları

ile $(x_o, y_o)$ noktası arasındaki uzaklıklarının oranıdır. Çemberin içindeki iskelet noktalarının kullanılmasının sebebi: kolların, çemberin dışında kalan parçalarının saat göstergesindeki diğer unsurlarla çakışma olasılığın olmasıdır. Şekil 4-28'de açı değerlerindeki $(\theta)$ nokta sayılarını $(n)$ veren grafik (imza gösterimi) gösterilmiştir.



**Şekil 4-27** Şekil 4-26'daki görüntünün, üzerine çember oturtulmuş hali




**Şekil 4-28** Açı değerlerindeki nokta sayılarını veren grafik (imza gösterimi)

**3.** Şekil 4-28'deki grafikte, en çok noktanın bulunduğu açı değeri $\theta_1$ bulunur. $\theta_1$ ile arasındaki açının farkı en az $5°$ olan ve en çok noktanın bulunduğu ikinci açı değeri $\theta_2$ bulunur. $\theta_1$ ve $\theta_2$ ile arasındaki açı farkları en az $5°$ olan ve en çok noktanın bulunduğu üçüncü açı değeri $\theta_3$ bulunur. Çakışma durumu olmaması halinde en fazla üç ayrı kol olduğundan dolayı üç açı değeri bulunması

yeterlidir. Bulunan üç açı değeri ve açılardaki nokta sayıları Çizelge 4-9'da gösterilmiştir.

**Çizelge 4-9** Açı değerleri ve açılardaki nokta sayıları

| | **Açı Değerleri** | **Açılardaki Nokta Sayıları** |
|---|---|---|
| $\theta_1$ | *189°* | *33 piksel* |
| $\theta_2$ | *304°* | *30 piksel* |
| $\theta_3$ | *279°* | *28 piksel* |

**4.** Bulunan açı değerlerindeki nokta sayılarını karşılaştırmadan önce bulunan üç açıdaki nokta sayılarına, $\theta_n+1^o$, $\theta_n+2^o$, $\theta_n-1^o$ ve $\theta_n-2^o$ $n=1,2,3$ açılarındaki nokta sayıları da eklenir. Toplam nokta sayıları $T_n$ $n=1,2,3$ büyükten küçüğe olacak şekilde $\theta_n$ açıları sıralanır. Toplam nokta sayıları ve sıralama Çizelge 4-10'da gösterilmiştir.

**Çizelge 4-10** Toplam nokta sayıları

| | **Açı Değerleri** | | | | | **Toplam Nokta Sayıları** |
|---|---|---|---|---|---|---|
| ***n*** | $\theta_{n-2}$ | $\theta_{n-1}$ | $\theta_n$ | $\theta_{n+1}$ | $\theta_{n+2}$ | $T_n$ |
| *1* | *302°* | *303°* | *304°* | *305°* | *306°* | *63 piksel* |
| *2* | *277°* | *278°* | *279°* | *280°* | *281°* | *53 piksel* |
| *3* | *187°* | *188°* | *189°* | *190°* | *191°* | *52 piksel* |

**5.** $T_n$ $n=1,2,3$ sayılarının oluşturduğu grubun ortalaması $m$ ve standart sapması $\sigma$ bulunur. Çizelge 4-10'daki toplam nokta sayıları kullanılarak, $m=56$ ve $\sigma=6$ olarak bulunmuştur.

**6.** Bulunan $m$ ve $\sigma$ değerleri kullanılarak aşağıda belirtilen koşullar sırasıyla kontrol edilir ve bölütlenmiş ayrı kol sayısı $k$ ve kolların açıları $\gamma_l$ $l=1, \ldots, k$ bulunur. Kolların açıları: kolların iskeletlerin $(x_o, y_o)$ noktası orijini olan ve yatay ekseni görüntü düzleminin yatay eksenine paralel olan koordinat sisteminin yatay eksenine göre pozitif açılarıdır. Bölütlenmiş üç kol olma koşulu Eşitlik 4.17'de

gösterilmiştir. Eğer kol sayısı üç olarak bulunursa kolların açıları $\gamma_1=\theta_1$, $\gamma_2=\theta_2$ ve $\gamma_3=\theta_3$ olarak bulunur.

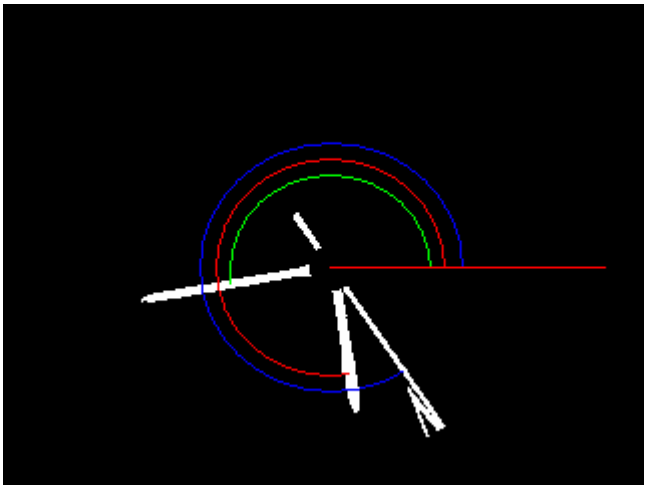
$$\begin{aligned} &T_2 \geq m \; ve \; T_3 \geq m \; veya \\ &T_2 \geq m \; ve \; T_3 \geq \sigma \; veya \\ &T_2 \geq \sigma \; ve \; T_3 \geq \sigma \end{aligned} \qquad (4.17)$$

Bölütlenmiş iki kol olma koşulu Eşitlik 4.18'de gösterilmiştir. Eğer kol sayısı iki olarak bulunursa kolların açıları $\gamma_1 = \theta_1$ ve $\gamma_2 = \theta_2$ olarak bulunur.

$$T_2 \geq \sigma \; ve \; T_3 < \sigma \qquad (4.18)$$

Bahsedilen iki koşul da sağlanmıyorsa bölütlenmiş tek kol vardır. Eğer tek kol var ise $\gamma_1 = \theta_1$ olarak bulunur.

Çizelge 4-10'deki gösterilen, bölütleme işlemi sonucunda Şekil 4-25'daki görüntüdeki bölgeden çıkarılan özniteliklere göre, üç ayrı kol olduğu kolların hiçbirinin çakışmadığı algılanmıştır. Üç ayrı kol olduğu için, kolların açılarının $\gamma_1=\theta_1=304^o$, $\gamma_2=\theta_2=279^o$ ve $\gamma_3=\theta_3=189^o$ olduğu belirlenmiştir. Şekil 4-25'deki görüntüdeki kolların açıları Şekil 4-29'da $\gamma_1$ mavi, $\gamma_2$ kırmızı ve $\gamma_3$ yeşil renkte gösterilmiştir.



**Şekil 4-29** Kolların açıları

### 4.6.3. Kolların tanınması

Saat, dakika, ve saniyenin hesaplanması için, akrep, yelkovan ve saniye kollarının, $(x_o, y_o)$ noktası orijini olan ve yatay ekseni görüntü düzleminin yatay eksenine paralel olan koordinat sisteminin yatay eksenine göre açıları gerekir. Bölütlenen kol sayısı bulunurken, kolların açıları hesaplanmıştır fakat kollar sınıflandırılmamıştır. Saat, dakika, ve saniyenin hesaplanması için kolların tanınarak sınıflandırılması gerekir.

Görüntüden bölütlenen kolların sayısı ve açıları belirlendikten sonra kolların tanınması için, kolların sayısına göre, kolları doğru sınıflandıracak öznitelikler çıkarılır. Kollardan çıkarılan öznitelikler kolların sayısına göre karşılaştırılarak, görüntüden bölütlenen kollardan akrep ve yelkovan kolu olan bulunur. Bu işlem sonunda kolların açıları daha önceden bulunduğu için akrep ve yelkovan kollarının açıları bulunmuş olur.

#### 4.6.3.1. Kollardan öznitelik çıkarma

Şekil 4-2'deki örnek saat görüntülerinde görsel analiz yapıldığında, Bölüm 3.1.2'de anlatılan bölge karakteristiklerine göre çıkarılabilecek doku ve topolojik özniteliklerin çıkarılmasının uygun olmadığı çünkü bu tür özniteliklerin, kolların hepsinde benzer hatta aynı olduğu görülmektedir.

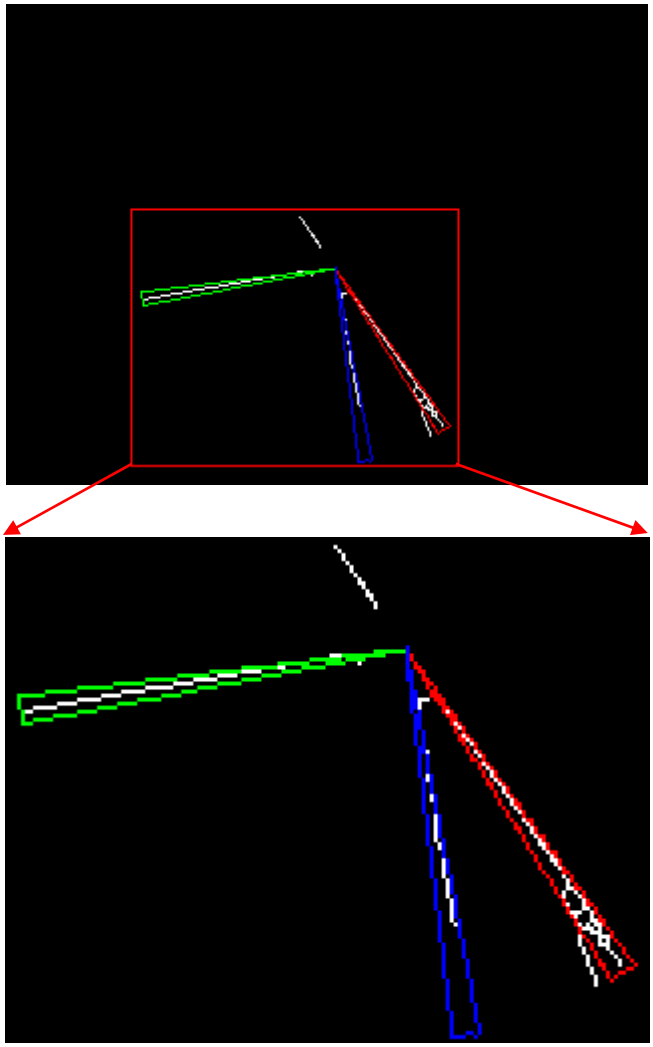
Bu çalışmada, kolların alanı ve uzunluğu kolların sınıflandırılmasında belirleyici olmaktadır. Bu nedenle kollardan çıkarılacak öznitelikler kolların alanları açıları $a_l$ $(l=1,...,k)$ değerleri ve uzunlukları açıları $u_l$ $(l=1, ..., k)$ değerleri olarak belirlenmiştir. Kolların alanı, görüntü düzleminde kolları oluşturan noktaların sayısı olarak tanımlanır. Kolların uzunluğu da kolun iskeleti üzerinde bulunan ve $(x_o, y_o)$ noktasına en uzak noktanın, $(x_o, y_o)$ noktasına olan öklidyen uzaklığı olarak tanımlanır. Saniye kolu ve yelkovan kolu arasındaki temel fark alanlarının faklı olmasıdır. Akrep ile yelkovan arasında fark uzunluklarının ve alanlarının farklı olmasıdır. Akrep ile yelkovan kolu arasında olan uzunluk farklılığı, saniye kolu ve akrep kolu arasında da vardır.

Seçilen öznitelikler belirleyici olmasının yanında kolların görüntü içindeki konumunun değişimine duyarsızdır. Saat göstergesi görüntünün neresinde olursa olsun, kolların görüntü düzleminde kapladıkları alan, kolların iskelet yapısı ve uzunlukları aynıdır. Seçilen öznitelik değerleri saat göstergesinin kameraya olan uzaklığının değişimi ile değişim göstermekle birlikte kolların alan ve uzunluklarının aynı oranda artıp azalması nedeniyle özniteliklerin sınıflandırmadaki işlevleri korunur. Ayrıca saniye kolunun önceden bölütlenip tanınmış olması ve kolların açılarını önceden bulunması, akrep ve yelkovan kollarının tanınmasını da kolaylaştırmaktadır.

Kolların uzunlukları $u_l$ $(l=1, \ldots, k)$ değerlerinin bulunmasında bölütleme işlemi sonunda elde edilen görüntüdeki bölgelerin inceltme algoritmasıyla bulunan iskeletleri kullanılır.

Bir kolun uzunluğunun bulunması için, kolun iskeleti üzerinde bulunan, $(x_o, y_o)$ noktası ile birleştiren doğru parçasının açısı $\gamma_l+2°$ ve $\gamma_l-2°$ ($\gamma_l$ uzunluğu bulunmak istenen kolun açısı) dereceleri arasında kalan ve $(x_o, y_o)$ noktası ile arasında Eşitlik 2.9'da tanımlanan öklidyen uzaklığı en büyük olan nokta bulunur. Bulunan noktanın uzaklığı kolun uzunluğu $u_l$ olarak belirlenir.

Şekil 4-26'daki görüntüde üç koldaki en uzak iskelet noktalarının aranacağı, $\gamma_l+2°$ ve $\gamma_l-2°$'leri ile belirlenen bölgeler, açısı $\gamma_3=189^o$ olan iskelet için yeşil, açısı $\gamma_1=304^o$ olan iskelet için mavi ve açısı $\gamma_2=279^o$ olan iskelet için kırmızı çizgilerle Şekil 4-30'da gösterilmiştir. En uzak iskelet noktası koordinatları ve kolların uzunlukları $u_1$, $u_2$, ve $u_3$ Çizelge 4-11'de verilmiştir.



**Şekil 4-30** En uzak iskelet noktalarının aranacağı bölgeler

**Çizelge 4-11** En uzak iskelet noktası koordinatları ve kolların uzunlukları

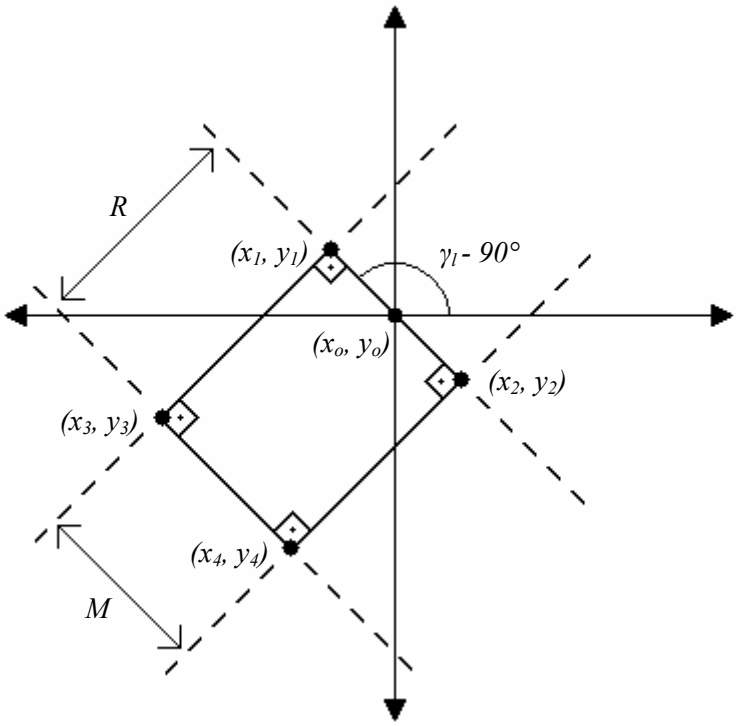
| | *x* | *y* | *Uzunluklar* ($u_l$) |
|---|---|---|---|
| **1. Kol** ($\gamma_1$=304°) | 69 | 147 | 96 |
| **2. Kol** ($\gamma_2$=279°) | 176 | 200 | 70 |
| **3. Kol** ($\gamma_3$=189°) | 217 | 210 | 96 |

Kolların alanı $a_l$ $(l=1, \ldots, k)$ değerlerini bulmak için kolları oluşturan noktaların bulunması gerekmektedir. Kollar tanınmamış olsa da, kolların açıları bilinmektedir. Bölütlenen kol sayısına göre, her kol için, bölütleme işlemi sonunda

elde edilen görüntü içindeki noktalardan aşağıdaki şartları sağlayan noktalar kolların noktası olarak belirlenir:

**1.** $f(x, y) = 255$

**2.** $(x, y)$ noktasının, Şekil 4-31'daki şekilde gösterilen, $(x_o, y_o)$ noktası orijini olan ve yatay ekseni görüntü düzleminin yatay eksenine paralel olan koordinat sistemindeki dört doğrunun arasındaki bölgede olması gerekir. $R$ değeri saatin konumun ikinci bileşeni, $\gamma_l$ kolun açısıdır. $M$ değeri de Şekil 4-25'de gösterilen $D$ uzunluğunun 1/4'ü kadar olması uygun görülmüştür. Yani 1. şartı sağlayan noktaların kol noktası olması için bir kenarı $R$ diğer kenarı $D$ uzunluğunda olan dikdörtgenin içinde olması gerekir. Kollar için belirlenen dikdörtgenler arasındaki farkı yaratan ise $\gamma_l$ açılarıdır.




**Şekil 4-31** Kolu içine alan doğrular

$(x, y)$ noktasının doğruların arasında olup olmadığının bulunması için doğruların denklemleri gerekir. Doğruların denklemlerinin bulunması içinde $(x_1, y_1)$, $(x_2, y_2)$, $(x_3, y_3)$ ve $(x_4, y_4)$ noktalarının koordinatları gerekir. Bu noktaların koordinatları aşağıdaki denklemlerle bulunur.

1. *Nokta koordinatları* : $x_1 = \cos(\gamma_l - 90°) * \frac{M}{2} \quad y_1 = \sin(\gamma_l - 90°) * \frac{M}{2}$ ( 4.19 )

2. *Nokta koordinatları* : $x_2 = \cos(\gamma_l + 90°) * \frac{M}{2} \quad y_2 = \sin(\gamma_l + 90°) * \frac{M}{2}$ ( 4.20 )

3. *Nokta koordinatları* : $x_3 = \cos(\gamma_l) * R + x_1 \quad y_3 = \sin(\gamma_l) * R + y_1$ ( 4.21 )

4. *Nokta koordinatları* : $x_4 = \cos(\gamma_l) * R + x_2 \quad y_4 = \sin(\gamma_l) * R + y_2$ ( 4.22 )

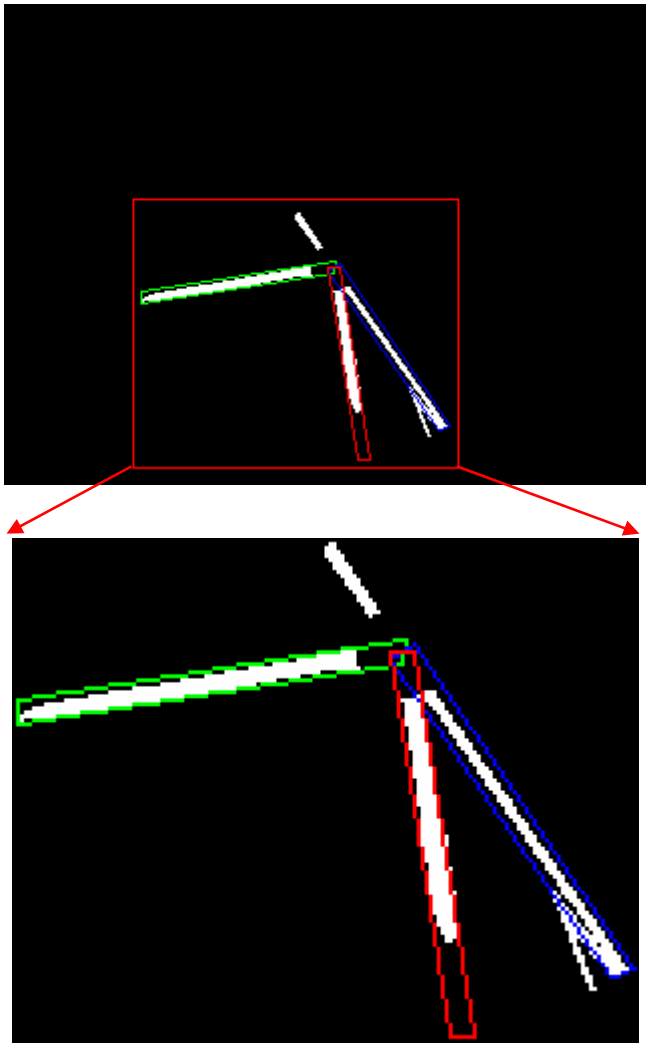
Şekil 4-25'deki görüntüdeki üç kol için, yukarıdaki denklemler kullanılarak bulunan koordinatlar Çizelge 4-12'de verilmiştir.

**Çizelge 4-12** Nokta koordinatları

| | $x_1$ | $y_1$ | $x_2$ | $y_2$ | $x_3$ | $y_3$ | $x_4$ | $y_4$ |
|---|---|---|---|---|---|---|---|---|
| **1. Kol ($\gamma_1$=304°)** | *161* | *133* | *167* | *129* | *215* | *213* | *221* | *209* |
| **2. Kol ($\gamma_2$=279°)** | *161* | *131* | *167* | *131* | *176* | *227* | *182* | *227* |
| **3. Kol ($\gamma_3$=189°)** | *164* | *128* | *164* | *134* | *68* | *143* | *68* | *149* |

Şekil 4-25'deki üç kol için bulunan $(x_1, y_1)$, $(x_2, y_2)$, $(x_3, y_3)$ ve $(x_4, y_4)$ noktalarının Eşitlik 4.23'teki iki noktası bilinen doğru denklemine uygulanmasıyla bulunan karşılıklı doğruların oluşturduğu bölgeler, açısı $\gamma_3$=189° olan yeşil, açısı $\gamma_1$=304° olan mavi ve açısı $\gamma_2$=279° olan kırmızı ile Şekil 4-32'de gösterilmiştir.

$$0 = (x - x_n) * (y_n - y_m) - (y - y_n) * (x_n - y_m) \quad (4.23)$$



**Şekil 4-32** Doğruların oluşturduğu bölgeler

Noktalar bulunduktan sonra, 1. şartı sağlayan *(x, y)* noktası, Şekil 4-31'deki iki noktası bilinen dört doğrunun denklemine koyularak aşağıdaki *S1, S2, S3* ve *S4* değerleri bulunur.

$$S1 = (x - x_o - x_1) * (y_1 - y_3) - (y_o - y - y_1) * (x_1 - y_3) \quad (4.24)$$

$$S2 = (x - x_o - x_2) * (y_2 - y_4) - (y_o - y - y_2) * (x_2 - y_4) \quad (4.25)$$

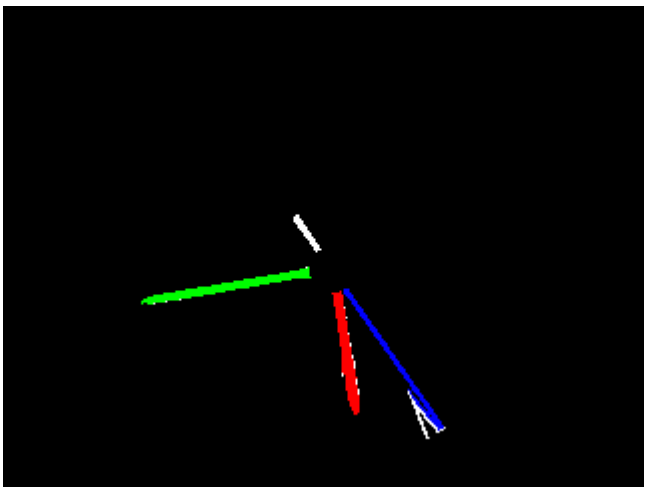
$$S3 = (x - x_o - x_1) * (y_1 - y_2) - (y_o - y - y_1) * (x_1 - y_2) \quad (4.26)$$

$$S4 = (x - x_o - x_3) * (y_3 - y_4) - (y_o - y - y_3) * (x_3 - y_4) \quad (4.27)$$

*S1, S2, S3* ve *S4* değerleri kullanılarak, *(x, y)* noktasının karşılıklı doğruların arasında olup olmadığı Eşitlik 4.28'deki koşul ile kontrol edilir. Eğer koşul sağlanıyorsa, *(x, y)* noktası, açısı $\gamma_1$ olan kolun noktası olarak belirlenir.

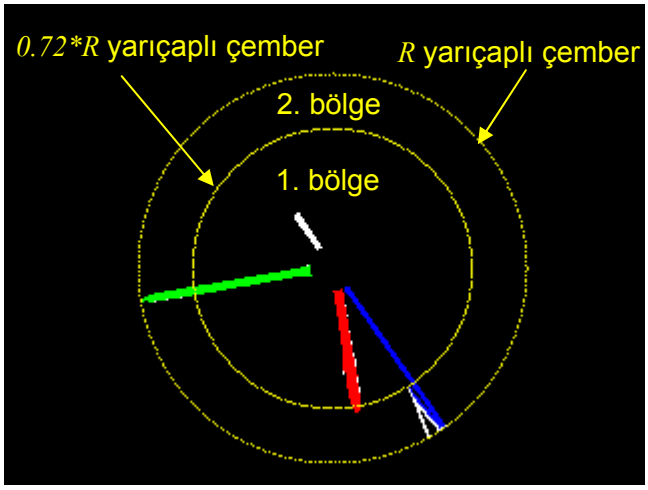
$$\begin{aligned}&\left[\left[(S1>0)\ ve\ (S2<0)\right] veya\ \left[(S1<0)\ ve\ (S2>0)\right]\right]\\ &\qquad\qquad ve\\ &\left[\left[(S3>0)\ ve\ (S4<0)\right] veya\ \left[(S3<0)\ ve\ (S4>0)\right]\right]\end{aligned} \qquad (4.28)$$

Şekil 4-25'deki görüntüde kol sayısı ve kolların açıları önceden bulunmuştu. Görüntüdeki noktalardan, kol noktası olma koşulunu sağlayanlar belirlenerek, açısı $\gamma_3=189^o$ olan yeşil kol, açısı $\gamma_1=304^o$ olan mavi kol ve açısı $\gamma_2=279^o$ olan kırmızı kol olarak Şekil 4-33'daki görüntüde gösterilmiştir.



**Şekil 4-33** Kol noktaları

Kol noktaları bulunduktan sonra kolların alanlarının bulunması için, her kola ait olan noktaların toplam sayısı bulunur. Kolların alanları bulunurken her kol için iki ayrı alan değeri bulunur. 1. alan değeri, 1. bölge olarak tanımlanan $(x_o, y_o)$ merkez noktalı $0.72*R$ yarıçaplı çemberin içinde kalan noktaların sayısıdır. İkinci alan değeri ise, 2. bölge olarak tanımlanan $(x_o, y_o)$ merkez noktalı $0.72*R$ yarıçaplı çember ile $R$ yarıçaplı arasında kalan noktaların sayısıdır. Tanımlanan çemberlerin Şekil 4-33'ün üstüne oturtulmuş hali Şekil 4-34'de görülmektedir. 2 farklı alan değerinin bulunmasının temel nedeni, görüntüdeki kolların, özellikle saatin göstergesinde ikinci bölgede bulunan diğer unsurlarla çakışma olasılıklarının olmasıdır.




**Şekil 4-34** Alan değerlerinin bulunduğu bölgeler

Şekil 4-34'deki kollar için bulunan alan değerleri $a_1$, $a_2$ ve $a_3$ Çizelge 4-13'de verilmiştir.

**Çizelge 4-13** Kolların alan değerleri

| | **1. Alan Değerleri** $(a_l)$ | **2. Alan Değerleri** $(a_l)$ |
|---|---|---|
| **1. Kol** ($\gamma_1=304^o$) | *154* | *120* |
| **2. Kol** ($\gamma_2=279^o$) | *285* | *7* |
| **3. Kol** ($\gamma_3=189^o$) | *253* | *110* |

### 4.6.3.2. Çıkarılan özniteliklerin karşılaştırılması

Görüntüden bölütlenen, sayısı $k$ ve açıları $\gamma_l$ $l=1, \ldots,k$ değerleri bilinen kollardan çıkarılan öznitelikler ve saniye kolunun açısı $\theta_S$ kullanılarak, kolların hangisinin akrep ve yelkovan kolu olduğu dolayısıyla akrep ve yelkovan kollarının açıları bulunur.

Gerçek zamanlı çalışma açısından gereksiz işlem süresi harcamamak için alan ve uzunluk özniteliklerinden hangilerinin çıkarılacağı, bölütleme işlemi sonunda elde edilen görüntüdeki kol sayısına $(k)$ göre belirlenir. Belirlenen öznitelikler kollardan çıkarılarak, açıları $\gamma_l$ $l=1, \ldots,k$ değerleri bilinen farklı kollara ait öznitelikler birbirleriyle karşılaştırılır. Karşılaştırma işlemi ile akrep ve yelkovan

kolunun bölütlenen kollardan hangisi olduğu bulunur. Kolların açıları bilindiği için akrep ve yelkovan kolunun bulunmasıyla, akrep $(\theta_A)$ ve yelkovan $(\theta_Y)$ kollarının açıları da bulunmuş olur.

Kolların saat göstergesindeki diğer unsurlarla çakıştığı veya herhangi iki kolun birbirine çok yakın olup da çakışmadığı durumlarda kollara ait öznitelikler bozulabilmektedir. Özniteliklerin bozulduğu durumlarda saatin yanlış okunmasını önlemek için kollara ait öznitelikler karşılaştırılırken birbirlerinden uygun görülen farklar kadar büyük veya küçük olması beklenmektedir. Uygun görülen fark değerleri, hangi iki kolun veya hangi özniteliklerin karşılaştırılacağına göre belirlenmiştir.

Bölütlenen görüntüdeki kol sayısı üç ise tek bir olasılık vardır. Bu olasılığa göre, kolların tanınmasında aşağıdaki adımlar izlenir:

**1.** Bölütlenen kolların sayısının bulunması işleminde bulunan kolların açıları $\gamma_1$, $\gamma_2$ ve $\gamma_3$, bölütleme işleminde tanınan saniye kolunun açısı $\theta_S$'le karşılaştırılarak, üç kol içinden saniye kolu olan ayrılır.

**2.** Saniye kolu ayrılmasıyla geriye akrep ve yelkovan kolları olan iki kol kalır. Akrep ve yelkovan arasındaki en temel fark olan uzunluklarını karşılaştırılabilmek için kalan iki kolun uzunlukları bulunur.

**3.** İki kolun uzunluğu karşılaştırılır. İki kol arasındaki mutlak uzunluk farkı 6 pikselden büyük ise uzun olan kol yelkovan ve diğer kolda akrep kolu olarak tanınır. $\gamma_1$, $\gamma_2$ ve $\gamma_3$ açılarından saniye kolunun açısı ayrıldıktan sonra, uzun olan kolun açısı yelkovan kolunun açısı $\theta_Y$ ve kısa olan kolun açısı da akrep kolunun açısı $\theta_A$ olarak belirlenir ve sınıflandırma işlemi bitirilir. Eğer fark 6 pikselden küçük ise bir sonraki adıma geçilir.

**4.** Önkoşullardaki kabullenmeye göre akrep kolu, yelkovan kolundan daha kısadır ve iki kolu ayırmak için sadece uzunluklarının karşılaştırılması yeterli olmaktadır fakat Şekil 4-2(a)'daki örnek saatte de olduğu gibi akrep kolunun saat göstergesindeki kollar dışındaki diğer unsurlarla çakışma olasılığı vardır.

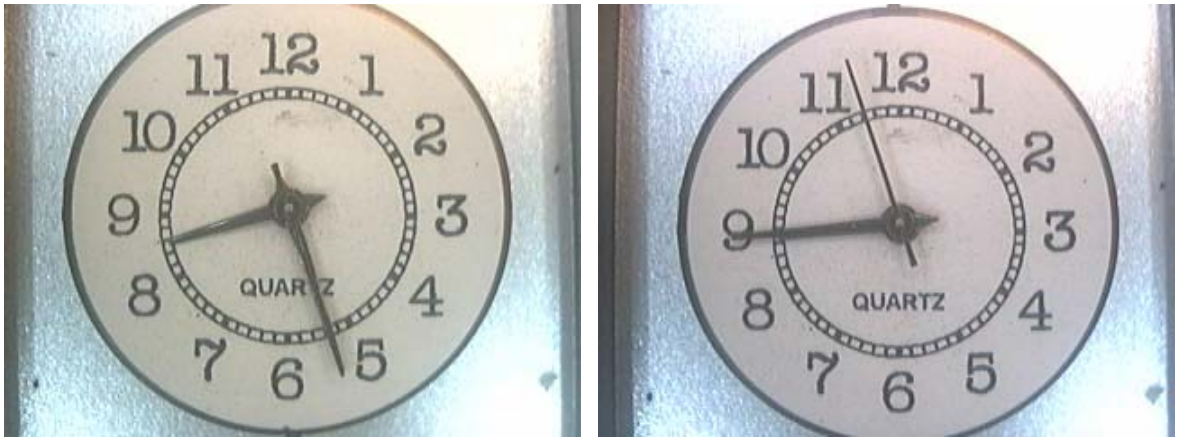
Böyle bir çakışma durumunda akrep kolunun uzunluğu olduğundan fazla çıkabilir ki bu da uzunluk özniteliğinin bozulması demektir. Bu durumda iki kolun 2. bölgedeki kapladıkları alanlar bulunur. Akrep ve yelkovan kolunun 1. bölgedeki alan değerleri, iki kolun kalınlığı aynı olduğu için, birbirine yakın olacağından, 2. bölgedeki alan değerlerinin karşılaştırılması uygun görülmüştür.

**5.** Bulunan alan değerleri karşılaştırılır. İki kol arasındaki alan değerlerinin mutlak farkı 10 pikselden büyük ise alan değeri büyük olan kol yelkovan ve diğer kolda akrep kolu olarak tanınır. $\gamma_1$, $\gamma_2$ ve $\gamma_3$ açılarından saniye kolunun açısı ayrıldıktan sonra, alanı büyük olan kolun açısı yelkovan kolunun açısı $\theta_Y$ ve alanı küçük olan kolun açısı da akrep kolunun açısı $\theta_A$ olarak belirlenir ve sınıflandırma işlemi bitirilir. Eğer fark 10 pikselden küçük ise arayüzün bilgi penceresinde saat okunamadı bilgisi verilir ve program, otomatik olarak saat oku düğmesine 2. durumda basılmış gibi davranır.

Şekil 4-33'deki görüntüde 3 kol olduğu için önceden bulunmuş olan $\gamma_3=189^o$, $\gamma_1=304^o$ ve $\gamma_2=279^o$ değerleri, saniye kolunun açısı $\theta_S=304.69°$ değeri ile karşılaştırılmıştır. Saniye kolu ile arasındaki açı farkı en düşük olan, açısı $\gamma_1=304^o$ olan kolun saniye kolu olduğu belirlenmiştir.

Açısı $\gamma_1=304^o$ olan kolun saniye kolu olduğu belirlendikten sonra, $\gamma_3=189^o$ ve $\gamma_2=279^o$ açılarındaki kolların, Çizelge 4-11'deki uzunlukları $u_3=96$ *ve* $u_3=70$ birbirleriyle karşılaştırılmıştır. İki kolun uzunluk değerleri karşılaştırıldığında $\gamma_3=189^o$ açısındaki uzun kolun yelkovan kolu olduğu, $\gamma_2=279^o$ açısındaki kısa kolun yelkovan kolu olduğu belirlenmiştir. Bu sınıflandırmaya göre akrep kolunun açısı $\theta_A=279^o$, yelkovan kolunun açısı $\theta_Y=189^o$ olmaktadır.

Bölütlenen görüntüdeki kol sayısı iki olduğu zaman üç olasılık vardır. Birinci olasılık: Akrep kolu ayrıdır, saniye ve yelkovan kolları çakışır. İkinci olasılık: Saniye kolu ayrıdır, akrep ve yelkovan kolları çakışır. Üçüncü olasılık ise: Yelkovan kolu ayrıdır, akrep ve saniye kolu çakışır. Üç olasılığın gerçekleştiği örnek saat görüntüleri Şekil 4-35'de gösterilmiştir.




**(a)** **(b)**

**(c)**

**Şekil 4-35 (a)** Saniye ve yelkovan kollarının çakıştığı görüntü
**(b)** Akrep ve yelkovan kollarının çakıştığı görüntü
**(c)** Saniye ve akrep kollarının çakıştığı görüntü

Üç olasılığa göre, kolların tanınmasında aşağıdaki adımlar izlenir:

**1.** Bölütlenen kolların sayısının bulunması işleminde bulunan kolların açıları $\gamma_1$ ve $\gamma_2$ ile bölütleme işleminde tanınan saniye kolunun açısı $\theta_S$ arasındaki farklar bulunur.

**2.** Birinci olasılığa göre kolların tanınmasında izlenecek adımlar: Bu olasılık, bölütlenen üç kol olduğunda saniye kolu ayrıldıktan sonra iki kolun kalmasıyla aynı durumdur. Bu nedenle, bölütlenen üç kol olduğundaki adımlar, 2. adımdan başlanarak ve bir koşul eklenerek izlenir. Bu koşul, uzunluk veya alan değeri belirtilen farkla diğer kola göre büyük olan kolun, saniye kolu ile arasındaki

açının $3°$'den küçük olmasıdır. Eğer birinci olasılığa göre kolların sınıflandırma işlemi gerçekleşmemişse, ikinci olasılığın kontrolüne geçilir.

**3.** İkinci olasılığa göre kolların tanınmasında izlenecek adımlar:

**i.** Yelkovan ve saniye kolunun uzunluğu aynı olduğundan, yelkovan ve akrep kolunun çakışarak oluşturduğu kolun ve saniye kolunun 1. bölgede kapladıkları alanlar bulunur. Akrep ve yelkovan kolunun 2. bölgedeki alan değerleri, iki kolun, göstergedeki kollar dışındaki diğer unsurlarla çakışma olasılığı olduğu için, doğru çıkmama olasılığı nedeniyle, 1. bölgedeki alan değerlerinin karşılaştırılması uygun görülmüştür.

**ii.** Bulunan alan değerleri karşılaştırılır. İki kol arasındaki alan değerlerinin mutlak farkı 15 pikselden büyük ve alan değeri küçük olan kolun, saniye kolu ile arasındaki açı $3°$'den küçük ise alan değeri küçük olan kol saniye ve diğer tek kolunda akrep ve yelkovanın çakışmasıyla oluşan kol olarak tanınır. $\gamma_1$, $\gamma_2$ açılarından alanı büyük olan kolun açısı, yelkovan kolunun açısı $\theta_Y$ ve akrep kolunun açısı $\theta_A$ olarak belirlenir ve sınıflandırma işlemi bitirilir. Eğer fark 15 pikselden küçük veya açı $3°$'den büyük ise üçüncü olasılığın kontrolüne geçilir.

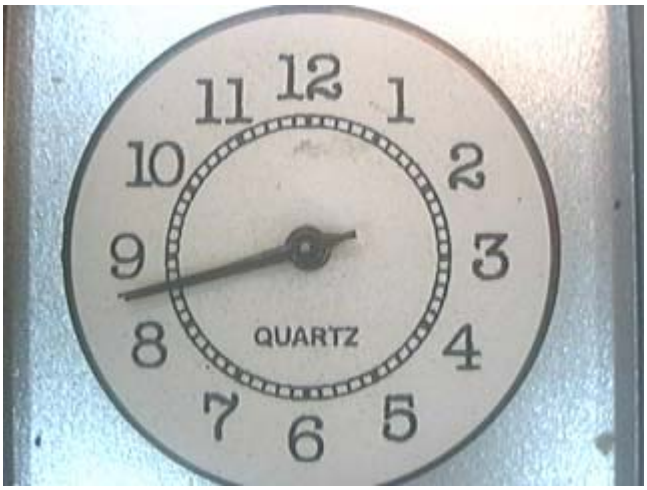
**4.** Üçüncü olasılığa göre kolların tanınmasında izlenecek adımlar:

**i.** Yelkovan ve saniye kolunun uzunluğu aynı olduğundan, saniye ve akrep kolunun çakışarak oluşturduğu kolun ve saniye kolunun 2. bölgede kapladıkları alanlar bulunur. Saniye ve akrep kolunun çakışarak oluşturduğu kolun ve yelkovan kolunun 1. bölgedeki alan değerleri, akrep ve yelkovan kolunun kalınlığı aynı olduğu için, birbirine yakın olacağından, 2. bölgedeki alan değerlerinin karşılaştırılması uygun görülmüştür.

**ii.** Bulunan alan değerleri karşılaştırılır. İki kol arasındaki alan değerlerinin mutlak farkı 10 pikselden büyük ve alan değeri küçük olan kol ile saniye kolu arasındaki açı $3°$'den küçük ise alan değeri büyük olan kolun yelkovan kolu ve diğer tek kolda akrep ve saniyenin çakışmasıyla oluşan kol olarak

tanınır. $\gamma_1$, $\gamma_2$ açılarından, alanı büyük olan kolun açısı yelkovan kolunun açısı $\theta_Y$ ve alanı küçük olan kolun açısı da akrep kolunun açısı $\theta_A$ olarak belirlenir ve sınıflandırma işlemi bitirilir. Eğer fark 10 pikselden küçük ise arayüzün bilgi penceresinde saat okunamadı bilgisi verilir ve program, otomatik olarak saat oku düğmesine 2. durumda basılmış gibi davranır.

Bölütlenen görüntüdeki kol sayısı bir ise, üç kolun kesiştiği tek bir olasılık vardır. Üç kolun kesiştiği saat görüntüsü Şekil 4-36'da gösterilmiştir.




**Şekil 4-36** Üç kolun kesiştiği saat görüntüsü

Üç kolun kesiştiği durumda, kolların tanınmasında aşağıdaki tek adım izlenir:

**1.** Bölütlenen tek kolun açısı $\gamma_1$ ile saniye kolunun açısı $\theta_S$ arasındaki fark $3°$'den küçük ise, akrep, yelkovan ve saniye kollarının çakışmış olduğu belirlenir. Akrep ve yelkovan kollarının açıları $\theta_A$ ve $\theta_Y$ belirlenir ki bu durumda iki açı değeri $\gamma_1$ açısına eşit olur. Eğer fark $3°$'den büyük ise arayüzün bilgi penceresinde saat okunamadı bilgisi verilir ve program, otomatik olarak saat oku düğmesine 2. durumda basılmış gibi davranır.

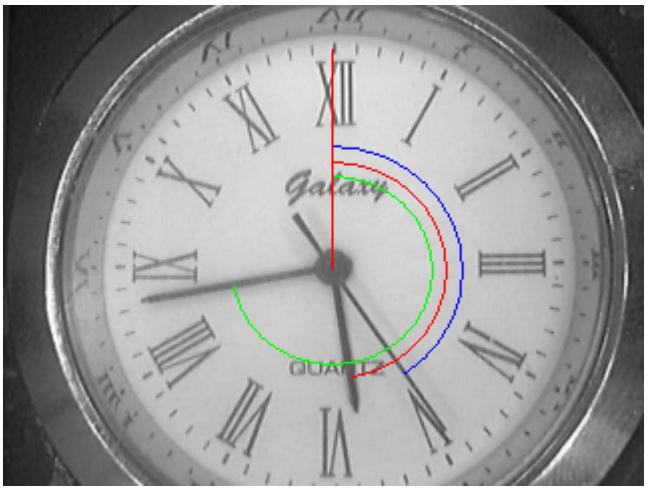
### 4.6.4. Saat, dakika ve saniye'nin hesaplanması

Akrep, yelkovan ve saniye kolları tanınıp sınıflandırıldıktan ve açıları belirlendikten sonra, aşağıdaki adımlar izlenerek saat, dakika ve saniye değerleri hesaplanmaktadır:

**1.** Saat dakika ve saniyenin hesaplanmasına geçildiği için arayüzün bilgi penceresinde saat okundu bilgisi verilir.

**2.** Kolların açıları $\theta_S$, $\theta_A$, *ve* $\theta_Y$ Eşitlik 4.19'daki denkleme uygulanarak kolların, $(x_o, y_o)$ noktası orijini olan ve yatay ekseni görüntü düzleminin yatay eksenine paralel olan koordinat sisteminin dikey eksenine göre saat yönündeki $\theta'_A$, $\theta'_Y$ ve $\theta'_S$ açıları elde edilir. $\theta_A=279°$, $\theta_Y=189°$ ve $\theta_S=304.69°$ açıları Eşitlik 4.29'daki denkleme uygulanarak $\theta'_A=171°$, $\theta'_Y=261°$ ve $\theta'_S=145.31°$ bulunmuştur. Şekil 4-10(d)'deki görüntüdeki akrep, yelkovan saniye kollarının dikey eksene göre açıları, Şekil 4-37'de $\theta'_S$ mavi, $\theta'_A$ kırmızı ve $\theta'_Y$ yeşil renkte gösterilmiştir.

$$\theta' = \left[360° - (\theta - 90°)\right] \bmod 360° \qquad (4.29)$$




**Şekil 4-37** Akrep, yelkovan ve saniye kollarının açıları

**3.** $\theta'_A$, $\theta'_Y$ ve $\theta'_S$ açıları Eşitlik 4.20'deki denkleme uygulanarak saat dakika ve saniye değerleri hesaplanır. $\theta'_A=171°$, $\theta'_Y=261°$ ve $\theta'_S=145.31°$ açıları Eşitlik

4.30'deki denkleme uygulanarak *Saat = 5.7, Dakika=43.5* ve *Saniye=24.2* bulunmuştur.

$$Saat = \frac{\theta'_A}{30} \quad Dakika = \frac{\theta'_Y}{6} \quad Saniye = \frac{\theta'_S}{6} \qquad (4.30)$$

**4.** Saati okumada kullanılan en son görüntünün yakalanmasından, saniye değerinin bulunmasına kadar geçen süre bulunan saniye değerine eklenir. Saniye süresinin değişmesi nedeniyle saat, dakika ve saniye değerleri güncellenir. Güncellenen değerlerin ondalık olma olasılığı olduğu için gerekli düzeltmeler yapılarak değerler tamsayıya çevrilir. Görüntünün yakalanmasından saniye değerinin bulunmasına kadar geçen süre *1.33s* olarak ölçülmüştür. Bu süre saniye değerine eklenerek, değerler üzerinde gerekli düzeltmeler yapılmıştır. Arayüzün saat penceresinde ilk gösterilecek değerler *Saat = 5, Dakika=43* ve *Saniye=25*'dir.

**5.** Arayüzün saat penceresinde zaman her saniye güncellenerek <saat><dakika><saniye> biçiminde gösterilir. Zamanın gösterilmeye başlanmasıyla birlikte program aynı anda önizleme yapmaya devam eder ve arayüzün bilgi penceresinde önizleme yapılıyor bilgisi verilir.

## 5. SONUÇ

Sayısal görüntü işlemenin uygulandığı alanlarda, birçok birim amaçlarını gerçekleştirebilmek için çevrelerindeki görsel bilgiye ihtiyaç duymaktadır. Bu tür birimler çevrelerindeki görsel bilgiyi sahne analizi yaparak elde etmektedir. Bugüne kadar yapılan çalışmalarda, kapalı ve açık alanlardaki sahnelerde farklı birçok amaç için değişik sahne analizi çalışmaları yapılmıştır [1-6]. Bahsedilen çalışmalarda birtakım belirlenmiş görevler başarı ile gerçekleştirilmiş ise de hiçbirinde uygulamadan bağımsız genel amaçlı bir sahne analizi sunulmamıştır. Bu çalışmada da bir sahne analizi olarak görülen analog göstergeli saatin optik görüntüsünden saatin kaç olduğunun anlaşılması için, bilgisayar tabanlı bir görüntü işlem sistemi olan saat okuma sistemi gerçekleştirilmiştir. Gerçekleştirilen sistem ile, saatin işaretçileri olan akrep, yelkovan ve saniye kolu olan bir saatin görüntüsü, karşısına yerleştirilen video kamera ile algılanmakta, algılanan görüntü bilgisayar ortamında gerekli görüntü işleme teknikleri işlenerek çözümlenmekte ve saatin kaç olduğu anlaşılmaktadır.

Görüntü işleme teknikleri ile analog göstergesi olan saatin okunmasıyla ilgili çeşitli çalışmalar yapılmıştır. Bu çalışmalarda analog göstergeli olan saatlerin okunmasında kullanılan görüntü işleme teknikleri Hough dönüşümü ve doğru algılama yöntemleridir. Saatin göstergesi içindeki dairesel ölçeklerin algılanmasında, Hough dönüşümüne dayanan dairesel yay algılama yöntemleri kullanılmıştır [7,8]. İşaretçiler ise, dairesel ölçeklerin veya dikdörtgen ölçeklerin içinde kalan görüntü düzlemindeki gri seviye profiller kullanılarak algılanmıştır [6].

Bu çalışmada analog göstergeli olan saatin okunmasında, karşıtlık arttırma ve yumuşatma gibi düşük seviyeli görüntü işleme teknikleri ve yüksek seviyeli görüntü işleme tekniklerinden oluşan nesne tanıma analizi kullanılmıştır. Saatin okunmasında nesne tanıma analizinin kullanılmasının getirdiği en büyük avantaj, Şekil 5-35 ve Şekil 5-36'da gösterilen saat göstergesinde kolların çakıştığı durumlarda da saatin başarı ile okunması olmuştur. Buna ek olarak önkoşullarda belirtilen özelliklere sahip bütün saatler bu çalışmada gerçekleştirilen saat okuma sistemi ile okunabilmektedir. Saat okuma sistemi aşağıda belirtilen adımları içermektedir.

- Saat görüntülerinin yakalanması, renkli görüntülerin gri seviyeye dönüşü ve gri seviye görüntüleri içinde sadece saniye kolunun hareket edip etmediğinin algılanması,

- Görüntü içindeki saatin konumunun bulunması ve sistem baştan başlatılmadıysa, her saat okuma işlemi yapılacağı zaman saatin konumunun değişip değişmediğinin kontrolü,

- Saniye kolu (hareketli nesne), akrep ve yelkovan kollarının (hareketsiz nesneler) görüntüden bölütlenmesi. Bölütlenen kolları dönme, ötelenme (görüntü içindeki konumları) ve büyüklük değişimlerinden (ağ kamerasının saate yakınlaşıp uzaklaşmasından) bağımsız bir şekilde ifade eden özniteliklerin çıkarılması. Kollara ait öznitelikler karşılaştırılarak akrep yelkovan ve saniye kollarının tanınması,

- Akrep, saniye ve yelkovan koluna ait öznitelikler kullanılarak saatin gösterdiği saat, dakika ve saniye değerlerinin bulunması.

Saat okuma sisteminde, saat oku düğmesine basılmasıyla saatin doğru olarak okunması amaçlanmıştır. Sistemin başarılı olmadığı durumlar da söz konusudur. Bu gibi durumlarda sistem durmamakta, saat oku düğmesine basıldıktan sonra kontrollerinin yapıldığı noktalarda hata algılanırsa, sistem, otomatik olarak yeni görüntüler yakalayarak saati okumaya çalışmaktadır. Sistemin hata algılamasının temel nedenleri, sistemin görüntü aldığı ortamda saniye kolundan başka hareket eden nesne olması ve saniye kolunun yelkovan koluyla tamamen çakışmasıdır.

Kollardan çıkarılan özniteliklerin bozulması da saatin doğru okunmasını engelleyen bir faktördür. Saat okuma işlemi sırasında görüntüden bölütlenmiş kol sayıları bulunurken, saat göstergesindeki kolların tamamen çakışmadığı fakat birbirlerine yakın olduğu durumlarda kolların çakışıyormuş gibi algılanması, kolların gösterdiği değerlerin çok az bir sapma ile algılanmasına yol açabilir. Özellikle bu durum saniye ve yelkovan kolları arasında gerçekleşmektedir. Böyle bir durum

gerçekleşirse sistemin okuduğu zaman ile saatin gösterdiği zaman arasında 1 saniye veya 1 dakikalık sapma gerçekleşebilir.

Bu çalışmada çözümleme sürecinin hızı, gerçek zamanlıdır. Saatin okunmasında gerçek zamanlı çalışmadan söz edebilmek için çözümleme sürecinin en fazla 1 saniye olması gerekmektedir. Çizelge 5-1'de, 5 tanesi saatin konumun bilindiği durum, 5 tanesinin de saatin konumunun bilinmediği durum olmak üzere 10 kez yapılan saat okuma işleminin çözümleme süreleri verilmiştir. Sürelerde 1.'den 5.'e gidildikçe, saatin kameraya uzaklığı azaltılmıştır. Her iki durumda da sürelerin gerçek zamanlı çalışmaya uygun olduğu ve saatin kameraya farklı uzaklıklarda olduğu durumlarda da okuma işleminin başarılı olduğu görülmektedir. Önceden beklendiği gibi okunacak saat ağ kamerasından uzaklaştıkça çözümleme süresinin düştüğü görülmektedir.

**Çizelge 5-1** Saatin okuma sisteminin çözümleme süreleri

| | **Süreler (saniye)** | |
|---|---|---|
| | **Saatin konumu biliniyor** | **Saatin konumu bilinmiyor** |
| **1.** | *0.906* | *1.125* |
| **2.** | *1.093* | *1.418* |
| **3.** | *1.094* | *1.39* |
| **4.** | *1.312* | *1.64* |
| **5.** | *1.421* | *1.671* |

## KAYNAKLAR LİSTESİ


[1] Ho, A.T.S., Zhu, X., Vrusias, B., Armstrong, J., **"**Digital Watermarking and Authentication for Crime Scene Analysis"**,** The Institution of Engineering and Technology Conference on 13-14 June 2006 Page(s):479 - 485

[2] Blaschof, W.F. and Caelli, T., **"**Scene Understanding By Rule Evaluation**",** Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 19, Issue 11, Nov. 1997 Page(s):1284 – 1288

[3] Chang, Jyh-Yeong, Cho, Chien-Wen, "Vision-Based Forward-Looking Traffic Scene Analysis Scheme", 2007 IEEE Intelligent Vehicles Symposium on 13-15 June 2007 Page(s):56 - 61

[4] T. Matsuyama and V. Hwang. SIGMA, a Knowledge-based Aerial Image Understanding System. Advances in Computer Vision and Machine Intelligence. Plenum, New York, 1990.

[5] D.M. McKeown, Jr. Wilson, W.A. Harvey, and L.E. Wixson. Automating Knowledge Acquisition for Aerial Image Interpretation. Comp. Vision Graphics and Image Proc, 46:37-81, 1989.

[6] Sablatnig, R. and Kropatsch, W. G., "Automatic Reading of Analog Display Instruments", Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on, Vol.1, 9-13 Oct 1994 Pages: 794-797

[7] H.K. Yuen, J. Princen, J. Illingworth, J. Kittler, "Comparative Study of Hough Transform Methods for Circle Finding", Image and Vision Computing, Vol. 8(1), pp. 71-77, 1990.

[8] P. Kierkegaard, "A Method for Detection of Circular Arcs Based on the Hough Transform", Machine Vision and Applications, Vol. 5 , pp. 249-263, 1992.

[9] GONZALES, C. Rafael and WOODS, Richard E., Digital Image Processing, 2nd Edition, Prentice Hall, 2001.

[10] Sengur, A.; Turkoglu, I.; Cevdet Ince, M., "Image segmentation applications with unsupervised neural networks," Signal Processing and Communications Applications Conference, 2005. Proceedings of the IEEE 13th , vol., no.pp. 271- 274, 16-18 May 2005

[11] Karaboga, D.; Basturk, B. "Image segmentation using differential evolution algorithm", Signal Processing and Communications Applications Conference, 2005. Proceedings of the IEEE 13th, 16-18 May 2005, Page(s): 33- 36.

[12] Pun T., Entropic thresholding, a new method, Computer Graphics Image Process. CGIP 16, 210-239, 1981.

[13] Kapur J.N., Sahoo P.K. and Wong A.K.C., A new method of gray level picture thresholding using the entropy of the histogram. Comput. Vision Graphics Image Process.,29, 273-285, 1985.

[14] Pal, N.R., Pal, S.K., A review on image segmentation techniques. Pattern Recognition. 26, 1277–1294, 1993.

[15] P. J. Besl and R. C. Jain, “Segmentation through variable-order surface fitting,” IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-IO, pp167-192, 1988.

[16] R. M. Haralick and L. G. Shapiro, “Image segmentation techniques,” Comput. Vis. Graph. Image Process., vol. 29, pp. 100-132, 1985.

[17] P. K. Sahoo, S. Soltani, and A. K. C. Wong, “A survey of thresholding techniques,” Comput. Vis.. Graph. Image Process., vol. 41, pp. 233-260,1988.

[18] Adams, R. and Bischof, L. 1994. Seeded Region Growing. IEEE Trans. Pattern Anal. Mach. Intell. 16, 6 (Jun. 1994), 641-647.

[19] D. H. Ballard and C. M. Brown, Computer Vision. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[20] F. Meyer and S. Beucher, “Morphological segmentation,” J. Vis. Commun. Image Represent., vol. 1, pp. 2146, 1990.

[21] PITAS, Ioannis, Digital Image Processing Algorithms, 1st Edition, Prentice Hall, 1993.

[22] L. Li and M. Leung, “Integrating intensity and texture differences for robust change detection,” IEEE Trans. Image Processing, vol. 11, pp. 105–112, Feb. 2002.

[23] P. Rosin, “Thresholding for change detection,” in Proc. Int. Conf. Computer Vision, 1998, pp. 274–279.

[24] Foley, D.H. and Sammon, J.W., Jr, “An Optimal Set of Discriminant Vectors Computers,” IEEE Transactions on, Vol.C-24, Iss.3, Pages: 281- 289, March 1975

[25] Y. Kurozumi and W. A. Davis, “Polygonal Approximation of Minimax Method”, Computer Graphics and Image Processing, Vol. 19, 1982, pp. 248-264.

[26] ERÜNVER, M. Ö., Gerçek zamanlı görsel nesne tanıma, Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir, 125s, 2000.

[27] Türkoğlu, İ., Yapay sinir ağları ile nesne tanıma, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 93s, 1996.

[28] BİLDİRİCİ Ö., Mekansal Veri Analizinde Point In Polygon Testi, TUJK 2003 yılı Bilimsel Toplantısı: CBS ve Jeodezik Ağlar Çalıştayı, 24-26 Eylül 2003, s209 - 214, 2003.

## EKLER

### EK 1 PCVC 840K/20 Philips Ağ Kamerası Teknik Özellikleri

| | |
|---|---|
| **Duyucu** | 1/3'' VGA CCD |
| **Çözünürlük** | VGA Video Çözünürlüğü (640x480) |
| **Video** | Saniyede 60 çerçeve video |
| **Görüntü Çözünürlüğü** | 1,2 Mega Piksel (1280x960) Maks. |
| **Ses** | Yerleşik Mikrofon |

### EK 2 HP NC6120 Diz Üstü Bilgisayarı Teknik Özellikleri

| | |
|---|---|
| **İşlemci / İşlemci Hızı** | Mobil Intel® Centrino 1.6 GHz, 399MHz FSB |
| **Ön Bellek** | 2MB |
| **Ekran Kartı** | 128 MB Paylaşımlı |
| **Sistem Belleği** | 512 MB, 333Mhz |
| **İşletim Sistemi** | Ms Xp Home Service Pack2 |

### EK 3 C# Kod Parçası

```
// Create a new bitmap
Bitmap görüntü = new Bitmap(this.pictureBox1.Image);

// Lock the bitmap's bits.
Rectangle rect = new Rectangle(0, 0, görüntü.width, görüntü.height);
System.Drawing.Imaging.BitmapData görüntüData =
        görüntü.LockBits(rect,
System.Drawing.Imaging.ImageLockMode.ReadWrite,
        görüntü.PixelFormat);

// Get the address of the first line.
IntPtr ptr = görüntüData.Scan0;

// Declare an array to hold the bytes of the bitmap.
int bytes = görüntü.width * görüntü.height * 4;
byte[] rgbValues = new byte[bytes];

 // Copy the RGB values into the array.
System.Runtime.InteropServices.Marshal.Copy(ptr, rgbValues, 0, bytes);

// Unlock the bits.
görüntü.UnlockBits(görüntüData);
```