

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**HAREKETLİ BİR NESNENİN HAREKETLİ KAMERA İLE GERÇEK  
ZAMANLITAKİBİ**

**Elek. Öğr. Sultan Süleyman KADIOĞLU**

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde  
"YÜKSEK LİSANS (ELEKTRİK MÜHENDİSLİĞİ)"  
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 23.09.2011  
Tezin Savunma Tarihi : 05.08.2011**

**Tez Danışmanı : Yrd.Doç. Dr. H.İbrahim OKUMUŞ**

**Trabzon 2011**

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü**  
**Elektrik-Elektronik Mühendisliği Anabilim Dalında**  
**Sultan Süleyman KADIOĞLU tarafından hazırlanan**

**HAREKETLİ BİR NESNENİN HAREKETLİ KAMERA İLE GERÇEK ZAMANLI  
TAKİBİ**

**başlıklı bu çalışma, Enstitü Yönetim Kurulunun 11 / 01 / 2011 gün ve 1387 sayılı  
kararıyla oluşturulan jüri tarafından yapılan sınavda**

**YÜKSEK LİSANS TEZİ**  
**olarak kabul edilmiştir.**

**Jüri Üyeleri**

**Başkan : Prof.Dr. A.Sefa AKPINAR** .....

**Üye : Yrd.Doç.Dr. İbrahim OKUMUŞ** .....

**Üye : Yrd.Doç.Dr. Hüseyin PEHLİVAN** .....

**Prof. Dr. Sadettin KORKMAZ**  
**Enstitü Müdürü**

## ÖNSÖZ

Bu tezde görüntü işleme teknikleri kullanılarak güvenlik amaçlı hareketli bir cismin hareketli bir kamera ile alınan görüntüsünü bilgisayar ortamında işleyen, cismi takip eden ve gerçek zamanlı video görüntülerini analiz edilerek işlenmesi, kamu güvenliği, önleyici hizmetler, olay tespit ve takip, trafik gözetleme ve sınır güvenliği gibi yüksek doğruluk ve güvenilirlik oranına ihtiyaç duyan sistemler için güvenilir teknikler geliştirilmiştir. Çalışmamın her aşamasında yol gösteren, fedakarlıkta bulunan, yardım ve desteğini esirgemeyen değerli hocam sayın Yrd. Doç. Dr. H.İbrahim OKUMUŞ'a sonsuz teşekkürlerimi sunarım.

Çalışmalarım süresince her konuda bana yardımcı olmaya çalışan, şube müdürüm sayın Özler ÖZER'e teşekkürlerimi sunarım.

Tez çalışmalarım sırasında maddi ve manevi destekleri ile beni hiçbir zaman yalnız bırakmayan eşim Sonay KADIOĞLU'na ve aileme de teşekkür ederim.

Sultan Süleyman KADIOĞLU  
Trabzon 2011

## TEZ BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “ Hareketli Bir Nesnenin Hareketli Kamera İle Gerçek Zamanlı Takibi ” başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd. Doç. Dr. H.İbrahim OKUMUŞ'un sorumluluğunda tamamladığımı, verileri kendim topladığımı, analizleri yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 08/07/2011

Sultan Süleyman KADIOĞLU

## İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	II
TEZ BEYANNAMESİ.....	III
İÇİNDEKİLER.....	IV
ÖZET .....	VIII
SUMMARY .....	VIII
ŞEKİLLER DİZİNİ .....	IX
TABLolar DİZİNİ.....	X
KISATMALAR DİZİNİ.....	XIV
SEMBOLLER DİZİNİ .....	XV
1. GİRİŞ.....	1
1.1. Adım Motorlar .....	2
1.1.1. Adım Motorların Kullanım Alanları.....	3
1.1.2. Adım Motorlarının Avantajları.....	4
1.1.3. Adım Motorlarının Dezavantajları .....	4
1.1.4. Adım Motorlarının Çalışma Prensipleri.....	5
1.1.5. Adım Motorlara Ait Bazı Önemli Kavramlar.....	6
1.1.5.1. Adım Tepkisi / Tek Adım Tepkisi / Cevap Süresi .....	6
1.1.5.2. Adım Oranı .....	7
1.1.5.3. Adım Açısı .....	7
1.1.5.4. Adım Açısı Doğruluğu .....	8
1.1.5.5. Çözünürlük .....	8
1.1.5.6. İleriye Atma (Overshoot ).....	8
1.1.5.7. Adım Motor Moment Parametreleri .....	9
1.1.5.7.1. Moment – Hız Grafikleri .....	11
1.1.5.8. Anma Geriliminin Anlamı .....	12
1.1.6. Adım Motorun Uçlarının Bulunması.....	12
1.1.7. Adım Motorların Çeşitleri .....	14
1.1.7.1. Sabit Mıknatıslı (Permanent Magnet, PM) Adım Motorlar.....	15
1.1.7.2. Değişken Relüktanslı (Variable Reluctance, VR) Adım Motorları.....	16

1.1.7.3.	Karma (Hibrid) Adım Motorları.....	17
1.1.8.	Bipolar ve Unipolar Adım Motor Ne Demektir?.....	19
1.1.8.1.	Çift Kutuplu ( Bipolar ) Adım Motor .....	19
1.1.8.2.	Tek Kutuplu ( Unipolar ) Adım Motor .....	20
1.1.9.	Adım Motorlarının Uyarımı .....	21
1.1.9.1.	Tek-faz uyarım (1 Fazlı Tam Adımlı Sürüş ) .....	21
1.1.9.2.	İki-faz Uyarım (2 Fazlı Tam Adımlı Sürüş).....	21
1.1.9.3.	Karma Uyarım (2 Fazlı Yarım Adımlı Sürüş).....	22
1.1.10.	Adım Motorlarının Denetimi .....	22
1.1.10.1.	Açık Döngü Denetim.....	22
1.1.10.2.	Kapalı Döngü Denetim .....	23
1.1.11.	Adım Motor Sürücü Sistemleri .....	24
1.1.11.1.	Temel Sürücüler.....	24
1.1.11.1.1.	Tek Kutuplu Sürme.....	25
1.1.11.1.2.	Çift Kutuplu Sürme.....	26
1.1.11.1.3.	İki Düzeyli Sürme.....	26
1.1.11.1.4.	L / R Sürme.....	27
1.1.11.1.5.	Kıyıcı Sürme.....	28
1.1.11.2.	NPN Transistörlü Unipolar Adım Motor Sürücü Devresi .....	29
1.1.11.3.	ULN2003 Entegresi ile Unipolar Adım Motor Sürücü Devresi.....	30
1.1.11.4.	UCN5804B Entegresi ile Tek Kutuplu Adım Motor Sürücü Devresi.....	31
2.	HAREKETLİ NESNE TAKİP YÖNTEMLERİ .....	32
2.1.	Dijital Görüntünün Temel Kavramları .....	33
2.1.1.	Görüntü Alımı.....	33
2.1.1.1.	Piksel .....	33
2.1.1.2.	Çözünürlük .....	34
2.1.2.	Dijital Görüntü.....	34
2.1.3.	Siyah – Beyaz Görüntü .....	35
2.1.4.	Renkli Görüntü .....	36
2.2.	Video Kodlama.....	37
2.3.	Görüntü Dosyaları .....	39

2.3.1.	Vektör ve Piksel Grafikler .....	39
2.5.	Görüntü Histogramı .....	39
2.6.	Nesne Takibinde Önemli Rol Oynayan Kriterler .....	41
2.6.1.	Nesne Gösterimleri .....	41
2.6.1.1.	Noktalar ile Gösterim .....	42
2.6.1.2.	Geometrik Şekil ile Gösterim .....	42
2.6.1.3.	Nesne Silueti ve Çevre Çizgisi ile Gösterim .....	42
2.6.1.4.	Eklemlili Şekil Modelleri ile Gösterim.....	43
2.6.1.5.	İskelet Model ile Gösterim .....	43
2.7.	Özellik Seçimi .....	43
2.8.	Nesne Takibi Metotları .....	45
2.8.1.	Tekli ve Çoklu Nesne Takibi.....	45
2.8.2.	Kalman Filtresi ile Nesne Takibi.....	45
2.8.3.	CONDENSATION Yöntemi ile Nesne Takibi.....	47
2.8.4.	Nitelik ve Modele Dayalı Nesne Takibi .....	48
2.8.4.1.	Niteliğe Dayalı Takip .....	48
2.8.4.2.	Modele Dayalı Takip .....	48
2.8.5.	İnsan Hareketlerinin Analizi Açısından Nesne Takibi .....	48
3.	PARALEL PORT İLE İLETİŞİM.....	51
3.1.	Paralel İletişim Genel .....	51
3.2.	Paralel Port Adresleri.....	53
3.3.	DATA Portu ile Programlama.....	54
4.	YAPILAN ÇALIŞMALAR VE İRDELEME .....	55
4.1.	Sistemin Yapısı.....	56
4.2.	Tasarım Aşaması .....	58
4.2.1.	Mekanik düzenek.....	59
4.2.2.	Sürücü Devresi .....	59
4.2.3.	Devrenin Bağlantı Şeması .....	60
4.3.	Devrenin Kurulumu ve çalışması .....	61
4.4.	MATLAB Programı .....	62
4.4.1.	Matlab Yazılımı .....	62

4.5.	Pozisyon Kontrol Sisteminin Tasarımı .....	63
4.5.1.	Görüntü Alımı.....	64
4.5.2.	Renk İndirgeme .....	64
4.5.3.	Nesne Bulma ve Takibi .....	64
4.5.3.1.	CamShift Yöntemi ile Nesne Bulma .....	67
4.5.3.2.	CamShift Yöntemi ile Nesne Takibi .....	68
4.6.	CamShift Yöntemi ile Nesne Takibi Adımları .....	70
5.	BULGULAR.....	76
6.	TARTIŞMALAR.....	77
7.	SONUÇLAR ve ÖNERİLER .....	78
7.1.	Geleceğe Yönelik Çalışmalar .....	80
8.	KAYNAKLAR .....	81
9.	EKLER .....	85

ÖZGEÇMİŞ



## ÖZET

### Hareketli Bir Nesnenin Hareketli Kamera ile Gerçek Zamanlı Takibi

Sultan Süleyman KADIOĞLU

Karadeniz Teknik Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. H. İbrahim OKUMUŞ

2011, 84 Sayfa Tez, 5 Sayfa Ek

Çalışmamızda, dinamik ortamlarda hareketli kamera ile hedef takibi yapan gerçek zamanlı bir sistem tasarımı gerçekleştirilmiştir. Hareketli görüntü alımı, hareketli bir platform üzerinde bulunan bir kamera ile yapılmaktadır. Sistem; görüntü alımı, hedef nesnenin tanınması ve hedef nesnenin takibi ana modüllerinden oluşmaktadır.

Hedef nesne kameranın görüş açısına girdiğinde nesnenin görüntü üzerindeki yeri bulunmakta ve nesne takibi işlemi başlamaktadır. Nesne takibi, kamera kullanılarak elde edilen ardışık video çerçevelerindeki hareketli bir nesnenin zaman boyunca çerçeve içerisindeki konum bilgisinin belirlenmesi işlemidir. Bunu gerçekleştirmek için MATLAB 7.9 ortamında Camshift algoritması kullanılmıştır.

Güvenlik amaçlı hareketli bir cismin kamera ile alınan görüntüsünü bilgisayar ortamında işleyen, cismin yörüngesini takip eden ve ileriki zamanki konumunu kestirebilen bir bilgisayar yazılımı geliştirilmiştir. İkinci aşamada ise bilgisayardan gelen veriye göre üzerine kamera yerleştirilen platformu hareket ettiren adım motoru mikrodenetleyici ile sürülmektedir.

Bu sistemde kullanılan teorik ve uygulamaya yönelik yöntemler anlatılmış ve karşılaşılan problemlere sunulan çözümler irdelenmiştir. Çalışma sonucunda elde edilen istatistiksel değerlerin incelenmesi sonucunda video çerçeveleri boyunca nesne takibinin başarı ile yapılabildiği gözlemlenmiştir.

**Anahtar Kelimeler:** Nesne Takibi, Nesne Bulma, Hareketli Nesne Takibi, Görüntü Tabanlı Güvenlik Sistemleri

## SUMMARY

Moving An Object In Real-Time Tracking Moving Camera

Sultan Süleyman KADIOĞLU

Karadeniz Technical University

The Graduate School of Natural and Applied Sciences

Electrical-Electronic Engineering Department

Supervisor: Yrd. Doç. Dr. H. İbrahim OKUMUŞ

2011, 84 Pages, 5 pages Appendix

In this study, a real time system design, which can track the target in dynamic environments with moving camera, was carried out. Taking moving image is being done by a camera which is located on a moving platform. The system consists of a main module like taking image, identifying target object and a tracking target object.

When the target object is seen on the camera, its location on the image is found and tracking the object starts. Tracking image is the process of determining the location information in the frame of a moving object gained by using camera in sequential video frames. To carry out this, Camshift Algorithm in MATLAB 7.9 was used.

For the study a software was developed. This software can process the image of an object on computer environment taken by a camera for security, can track the object's orbit and can guess the future location of the object. On the second stage, stepper motor which moves the platform located a camera on according to the data coming from computer is ran by microcontroller.

The methods used in this system for theoretical and implementation were stated and the solutions for the problems encountered were examined. The statistical values gained as a result of study show that tracking objects along the video frames could be done successfully..

**Key Words :** Tracking Objects, Finding Object, Tracking Moving Object, Image Based Security Systems

## ŞEKİLLER DİZİNİ

**Sayfa No**

Şekil 1.1. Adım motoru.....	2
Şekil 1.2. Adım motorunun çalışma prensibi.....	5
Şekil 1.3. Adım motoru tek adım tepkisi .....	7
Şekil 1.4. Adım motorun overshoot eğrisi .....	8
Şekil 1.5. Dönme momenti .....	9
Şekil 1.6. Adım motorunun basit bir şekli .....	10
Şekil 1.7. Adım motorun moment-hız grafiği.....	11
Şekil 1.8. Farklı yapılarda adım motor .....	13
Şekil 1.9. Adım motorda uç sırası bulma yöntemi .....	14
Şekil 1.10. Sabit mıknatıslı adım motor.....	15
Şekil 1.11. 4-fazlı sabit mıknatıslı adım motoru.....	15
Şekil 1.12. Değişken relüktanslı adım motorun yapısı .....	16
Şekil 1.13. Karma adım motoru şekli .....	17
Şekil 1.14. Karma adım motorunun rotoru .....	18
Şekil 1.15. Karma adım motorunun yapısı .....	18
Şekil 1.16. Çift kutuplu adım motoru .....	19
Şekil 1.17. Çift kutuplu adım motoru modeli .....	20
Şekil 1.18. Tek kutuplu adım motoru modeli .....	20
Şekil 1.19. Adım motoru açık döngü denetimi blok diyagramı.....	23
Şekil 1.20. Adım motoru kapalı döngü denetimi blok diyagramı .....	23
Şekil 1.21. Tek kutuplu sürücü devresi.....	25
Şekil 1.22. Çift kutuplu sürücü devresi.....	26
Şekil 1.23. İki seviyeli sürücü devresi .....	27
Şekil 1.24. Seri direnci transistör ile şöntlenmiş sürücü devresi .....	28
Şekil 1.25. Kıyıcı sürücü devresi.....	30

Şekil 1.26. NPN transistörlü unipolar adım motor sürücü devresi .....	31
Şekil 1.27. ULN2003 entegresi ile unipolar adım motoru sürücü devresi.....	31
Şekil 1.28. UCN5804B entegresi ile tek kutuplu adım motoru sürücü devresi.....	32
Şekil 2.1. Pikseller .....	33
Şekil 2.2. Bir fotoğraf çerçevesindeki bir pikselin koordinatları .....	34
Şekil 2.3. Görüntü ve piksel değerleri serisi .....	34
Şekil 2.4. Görüntünün Koordinat Sistemi.....	36
Şekil 2.5. Görüntünün Kırmızı(R), Yeşil(G), Mavi(B) renk kodları .....	38
Şekil 2.6. Video dizisinde uzamsal ve zamansal örnekleme .....	39
Şekil 2.7. Dijital görüntü.....	42
Şekil 2.8. Görüntünün histogramı .....	42
Şekil 2.9. Nokta takibi .....	43
Şekil 2.10. Çevre çizgisi oluşturma örnekleri[12]. .....	44
Şekil 2.11. Tek boyutlu uzaya indirgenmiş renk olasılık dağılımı .....	45
Şekil 2.12. Optik akış (a) 1. Görüntü (b) Yön doğrultuları (c) 2. Görüntü. ....	46
Şekil 3.1. Paralel Portlarda Kullanılan DB-25 Konnektörü.....	52
Şekil 3.2. Paralel portun basit olarak karakterize edilişi.....	53
Şekil 3.3. Paralel port soketi görünüşüŞekil 3.4. Paralel port uçlarının görünüşü .....	53
Şekil 4.1. Hareketli nesne takip sistemi .....	57
Şekil 4.2. Hareketli kamera sistemi .....	58
Şekil 4.3. ULN 2003 entegresi bağlantı şekli .....	61
Şekil 4.4. ULN2003 entegresinin iç yapısında darlington bağlı transistörün yapısı .....	61
Şekil 4.5. ULN 2003'ün Adım motoru bağlantısı .....	62
Şekil 4.6. Adım motorun sürücü devrenin bağlantı şekli .....	62
Şekil 4.7. Sistem blok diyagramı .....	65
Şekil 4.8. Bhattacharyya uzaklığı .....	68

Şekil 4.9	Hedef nesnenin görüntüsü .....	69
Şekil 4.10	Elin Mean-Shift yöntemi ile bulunması.....	69
Şekil 4.12.	Elin CamShift yöntemi ile takibi .....	73
Şekil 4.11.	CamShift yöntemi ile nesne takibi yarı kodlama [58].....	75
Şekil 4.13.	CamShift yöntemi ile nesne takibine ait akış diyagramı [58].....	76

## **TABLolar DİZİNİ**

**Sayfa No**

Tablo 1.1. Adım motorlarda kablo bağlantısı tablosu.....	13
Tablo 1.2. Adım motoru tek-faz uyartım tablosu .....	21
Tablo 1.3. Adım motoru iki-faz uyartım tablosu .....	22
Tablo 1.4. Adım motoru karma uyartım .....	22
Tablo 3.1. BIOS veri alanındaki LPT Adresleri .....	53

## KISALTMALAR DİZİNİ

<b>AA</b>	: Alternatif akım
<b>AR-GE</b>	: Araştırma-geliştirme
<b>B</b>	: Mavi
<b>CNC</b>	: Bilgisayarla sayısal kontrol
<b>CW</b>	: Saat ibresi yönünde
<b>DA</b>	: Doğru akım
<b>Emk</b>	: Elektro motor kuvvet
<b>G</b>	: Yeşil
<b>GND</b>	: Toprak
<b>HSV</b>	: Renk tonu, doygunluk, değer
<b>H</b>	: Yüksek
<b>HSM</b>	: Karma tip adım motoru
<b>IEEE</b>	: Elektrik-Elektronik Mühendisleri Enstitüsü
<b>L</b>	: Düşük
<b>MATLAB</b>	: MATrix LABoratory
<b>N</b>	: Kuzey
<b>NPN</b>	: Negatif-Pozitif-Negatif
<b>PC</b>	: Kişisel bilgisayar
<b>PNP</b>	: Pozitif-Negatif-Pozitif
<b>PM</b>	: Sabit mıknatıslı adım motoru
<b>R</b>	: Kırmızı
<b>SA</b>	: Adım açısı
<b>S</b>	: Güney
<b>s</b>	: Arama alanı boyutu
<b>TTL</b>	: Transistör-Transistör-Lojik
<b>T</b>	: Zaman
<b>VR</b>	: Değişken relüktanslı adım motoru

## SEMBOLLER DİZİNİ

$X_i$	: Piksel koordinatları
$X_c, Y_c$	: Hedef nesnenin merkez koordinatları
$J_M$	: Rotor eylemsizliği (kg.m <sup>2</sup> )
$J_L$	: Yükün eylemsizlik momenti (kg.m <sup>2</sup> )
$G$	: Rotor ağırlığı (kg)
$D$	: Rotor çapı (m <sup>2</sup> )
$m$	: Adım motorunun faz sayısı
$Z$	: Rotor dişlerinin veya manyetik kutup çiftlerinin sayısı
$f_{sL}$	: Yüklü başlama frekans. (darbe/s)
$f_s$	: Yüksüz başlama frekansı (darbe/s)
$E_p$	: Statik açısal hata (%)
$H_s$	: Teorik adım açısı (derece)
$T_d$	: Tutma torku (N.m) (motor uyarıldığı durumda)
$T_h$	: Tutma torku (N.m) (motor uyarılmadığı durumda)
$T_o$	: Çekme torku (N.m)
$f_p$	: Darbe oranı (darbe/s)
$v$	: Dakika başına devirdeki rotor hızı (r/min)
$f$	: Besleme sinyallerinin frekansı (Hz)
$w$	: Saniye başına radyandaki rotor hızı (rad/sn)
$l$	: Çözünürlük veya birim adım (cm/step)
$D$	: Çıkış katı kasnağının çapı (cm)
$i$	: Küçültme diş oranı
$F$	: Yüke etkiyen kuvvet (Newton)
$F_A$	: Tepki kuvveti (Newton)
$J$	: Eğim (derece)
$\mu$	: Yüzeyin sürtünme katsayısı
$W$	: Yük ağırlığı (Newton)
$\mu$	: Verim
$n$	: $3.6^\circ / T_s$



<b>g</b>	: Yerçekimi ivmesi (cm2/sn)
<b>w</b>	: Hedef nesnenin genişliği
<b><math>\lambda</math></b>	: Eşik değeri
<b>P</b>	: Güç
<b>I</b>	: Akım
<b>F</b>	: Kuvvet
<b><math>T_m</math></b>	: Moment
<b><math>W_n</math></b>	: Açısal hız
<b>r</b>	: Yol

## 1. GİRİŞ

Günümüzde şehirlerde suç oranları her geçen yıl artmaktadır ve bu olayların aydınlatılmasında çok çeşitli önlemler alınmaktadır. Bu önlemlerden en önemlilerden biri kamera güvenlik sistemleridir. Polisiye olaylarda olayın olma anı ile polisin olaya müdahalesi arasında geçen zaman ne kadar kısa olursa olayın aydınlatılma ihtimali o kadar yüksek olur. Bu nedenle kamera sistemlerinden alınan görüntü ne kadar sağlıklı olursa olaya müdahale ve olayın aydınlatılması o kadar kolay olur. Günümüzde kullandığımız güvenlik kameralarının birçoğu sabit kameralardır. Sabit kameraların bazı dezavantajları vardır. Bunlardan en önemlisi sabit kameranın görüntü açısı sabit olduğundan çok kısa bir sürede hareketli cisim kameranın görüntü açısından çıkabiliyor bunun önüne geçebilmek için hareketli kamera sistemi tasarımına ihtiyaç duyulmuştur. Bu çalışmada hareket eden bir nesnenin takibini yapabilmek için öncelikle hedeflenen nesnenin görüntüdeki yerinin bulunması gerekir. Hedeflenen nesne, herhangi bir hareketli nesne olabileceği gibi uygulamanın amacına göre önceden belirlenmiş bir nesne de olabilir. Eğer hedef nesne önceden belirlenmemişse, öncelikle görüntüde bir hareket olana kadar beklenir, hareket gerçekleştikten sonra hareket eden nesne, hedef olarak belirlenir ve bu nesne takip edilmeye başlanır. Eğer hedef nesne önceden belirlenmişse görüntüde bu nesne aranır ve nesne bulunduktan sonra takip aşamasına geçilir. Takip edilecek nesnenin şekil özelliklerine göre nesneyi temsil edecek bir şekil seçilir. Hedef nesne, kendisini temsil eden şekil içerisindeki özelliklerden yararlanılarak takip edilir. Bu özellikler takip aşamasında önemli rol oynadığı için takip edilecek nesneyi çevreden ayırt edici kılacak şekilde farklı olmalıdır. Nesne takibi için kameradan ardı ardına alınan fotoğraf karelerinin MATLAB ortamında bir yazılımı geliştirilmiş, paralelport üzerinden adım motorunun sürücüsü denetlenerek motorun konum kontrolü sağlanmıştır.

Birinci bölümde adım motorlarının tanıtılması, adım motoru çeşitleri, adım motorlarının uyarılması, adım motor terminolojisi gibi konular incelenmektedir.

İkinci bölümde adım motorlarının kontrol yöntemleri ele alınmıştır. Ayrıca bilgisayarlar yardımıyla adım motorların kontrolünün nasıl yapılacağı üzerinde de durulmuştur. Bunun yanında bir de adım motor kontrol devresi tasarımına yer verilmiştir.

Üçüncü bölümde görüntü işleme tekniklerine yer verilmiştir.

Dördüncü bölümde ise paralelport tan veri gönderilmesine yer verilmiştir. Çalışmanın son bölümünde yapılan sisteme ilişkin çıkarımlar yapılmıştır.

### 1.1. Adım Motorlar

Bobin uçlarına darbe uygulandığında hareket eden motorlara adım motor denir. Adım motorlar, ortada mıknatıs veya metalden oluşan rotor ile rotoru çevreleyen ve üzerinde elektromanyetik alan etkisi yaratarak gerilim indüklemesi meydana getiren bobinlerden oluşur. Bobin uçlarına belli bir sıraya göre gerilim uygulanarak motorun adım hareketi sağlanır.

Bu tip motorlar genellikle özel dijital devreler ile mikroişlemci kontrollü devrelerde kullanılır. Adım motor devresi bir sayıcı ile kontrol edilebilir. Devre girişine uygulanan darbe sayısı kadar adım hareketi oluşur. Bu darbeler, adım motorun uçlarına belirli bir sırayla uygulanır. Adım motora gönderilen darbe dizisi değiştirilmeden devamlı uygulandığı sürece rotor sabitlenir ve motor durur.

Adım motorları, sürekli bir dönme hareketi yerine eşit büyüklükte ayırık açısal hareketler yapabilen bir elektromekanik sistemdir. Kontrol işaretleri analog gerilim ve akımlar yerine darbe işaretleridir. Adım motorunun sürücüsüne uygulanan bir sayısal darbe işareti motorun belirli bir açı kadar hareket yapmasına neden olur ve bu olay her darbe işareti için tekrarlanır. Böylece dönme olayı elde edilir. Şekil 1.1’de örnek bir adım motorunun şekli görülmektedir. Projede bir adet adım motoru kullanılmıştır[1].



Şekil 1.1. Adım motoru

Adım motorlar, sabit mıknatıslı (PM- permanent magnet), değişken relüktanslı (VR – variable reluctance) ve hybrid olmak üzere temelde üç sınıfa ayrılır. Bunların dışında değişik yapı ve özelliğe sahip adım motorlar bulunur.

Önceleri yaygın olarak kullanılan analog sistemlerde, mikroişlemcilerden alınan sayısal işaretleri analog a çevirebilmek için ara cihazların kullanılma zorunluluğu vardı. Ancak adım motorlarının doğrudan doğruya sayısal işaretlerle çalışması bu zorunluluğu

ortadan kaldırdı. Bu yüzden adım motorları aşağıdaki sayılan sebeplerden dolayı günümüzde çok yaygın olarak kullanılmaktadır.

- Adım motorlarında veriler sayısal olarak ele alınmaktadır. Sayısal tüm devreler hem ucuz olmakta hem de çıkış hareketinin sayısal olması istenmesi halinde bu motorlar ideal bir çözüm olarak görülmektedir.
- Ucuz mikroşlemciler sayesinde kontrol ve işlem rahatlıkla otomatize edilebilir. Bu yüzden adım motorları ideal hareket edebilen kontrol cihazlarındandır.
- Adım motorları basit konstrüksiyonları yüzünden kullanışlıdır.
- Daha güçlü, daha kullanışlı ve aynı zamanda ucuz katı hal güç düzenleri kullanılması bu motorlarının kullanım alanlarını bir hayli arttırmıştır.
- Lineer güç kuvvetlendiricisi adım motorları için gerekmez .
- Geri beslemesiz olarak da çalıştırılabilirler.
- Sayısal kontrol verilerine doğrudan cevap verirler.
- Çok geniş bir hız aralığında çalışabilirler.
- Problemsiz ve hızlı olarak devreye giriş çıkış özellikleri vardır [2].

### **1.1.1. Adım Motorların Kullanım Alanları**

Adım motorlar uzun yıllardır var olmalarına rağmen ticari olarak kullanılmaları ancak 1960'lı yıllarda yüksek seviyeli doğru akımları anahtarlayabilen transistörlerin üretimine başlanmasıyla yaygınlaşmıştır. 1970'li yıllardan beri dijital elektronikteki ve mikroşlemci teknolojisindeki gelişmelerle birlikte adım motorlarının kullanımı giderek cazipleşmekte ve tüm dünyada bu motorların üretim ve uygulamalarıyla ilgili geliştirme çalışmaları yapılmaktadır.

Günümüzde adım motorları endüstride birçok kontrol sistemlerinde, hassas konum kontrolü yapmak amacıyla kullanılmaktadır. En çok yazıcılar, çiziciler, disket sürücüler, harddisk sürücüler, kart okuyucular... vb gibi bilgisayar çevre cihazlarında bu elemanlardan yararlanılmaktadır. Ayrıca sayısal kontrol sistemlerinde, CNC( Bilgisayarla Sayısal Kontrol) tezgahlarda, proses kontrol sistemlerinde, robot teknolojisinde (milimetrik hareketlerin kontrolünde) ve uzay endüstrisine ait bir çok sistemde adım motorları tahrir elemanı olarak yer almaktadır [3].

### 1.1.2. Adım Motorlarının Avantajları

Adım motorların bu kadar çok kullanım alanı bulmasının nedeni bu motorların bazı avantajlara sahip olmasıdır. Bu avantajlar aşağıdaki gibi sıralanabilir.

- Adım motorlar dijital giriş işaretlerine cevap verirler, bu nedenle mikroişlemci veya bilgisayarlarla kontrol için ideal elemanlardır.
- Adım motorların hangi yöne döneceği, devir sayısı, dönüş hızı gibi değerlerin mikroişlemci veya bilgisayar yardımı ile kontrol edilebilmesi, her an bu motorların dönüş yönü, hızı ve konumunun bilinebilmesini sağlamaktadır. Bu özelliklerinden dolayı adım motorlar ile çok hassas konum kontrolü yapılabilir. Adım motorların dijital girişlere cevap vermesi, geri beslemeye ihtiyaç duyulmaksızın açık çevrim çalıştırılabilmesini sağlamaktadır. Yani açık çevrim çalıştırılan bir adım motoru ile hız, ivme ve konum kontrolü daha basit ve daha az maliyetle gerçekleştirilebilir. Böylece alışılmış kararsızlık problemlerinin de önüne geçilmiş olur.
- Adım motorlar, giriş işaretlerinin frekansına bağlı olarak çok geniş bir hız aralığında sürülebilirler.
- Adım motorlar, herhangi bir hasara yol açmadan defalarca durdurulup çalıştırılabilirler. (Sürerken aniden durdurma ya da aniden ters yönde sürme isteğine karşı mükemmel cevap verebilirler.)
- Aşırı yüklenmeden hasar görmezler, oldukça dayanıklıdırlar.
- Her yeni adımla artan (kümülatif) konum hataları yoktur.
- Mekanik yapısı basit olduğundan bakım gerektirmezler.
- Yağlanma ve kirlenme problemleri yoktur[3].

### 1.1.3. Adım Motorlarının Dezavantajları

Adım motorların bütün bu avantajlarına karşılık bazı dezavantajları da aşağıdaki gibi sıralanabilir.

- Adım açıları sabit olduğundan rotordan alınan hareket sürekli değil darbelidir.
- Klasik sürücülerle kullanıldıklarında verimleri düşüktür.
- Adım cevapları nispeten büyük aşım ve salınımlıdır.
- Yüksek eylemsizlikli yüklerde yetenekleri sınırlıdır.
- Sürtünme kaynaklı yükler, hata kümülatif olmasa dahi açık çevrim çalışmada

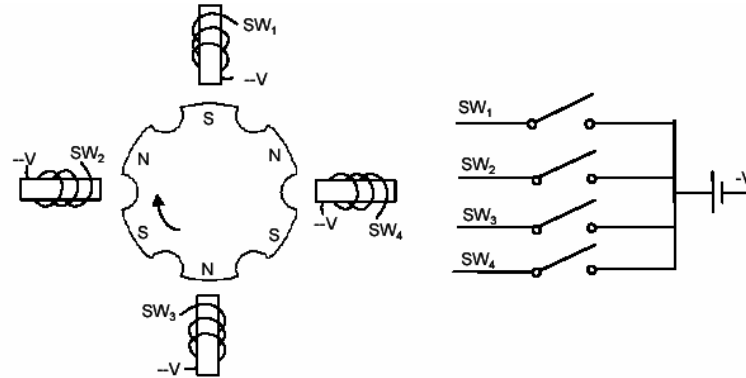
konum hatası meydana getirebilirler.

- Elde edilebilecek çıkış gücü ve momentı sınırlıdır.
- İyi kontrol edilmezse rezonans meydana gelebilir.
- Oldukça yüksek hızlarda çalıştırmak pek kolay değildir[4].

#### 1.1.4. Adım Motorlarının Çalışma Prensibi

Adım motorlar, girişlerine uygulanan lojik sinyalleri dönme hareketine çevirirler. İstedığınız yönde ve derecede döndürebileceğiniz adım motorlar, hassas hareketleri sayesinde, birçok cihazda konum kontrolü amacıyla kullanılmaktadır. Dijital çalışma ise H (Yüksek) ve L (Düşük) [1 ve 0] bilgileriyle çalışma demektir.

Diğer elektrik motorlarında, motora enerji verildiği zaman sürekli olarak rotorları dönmektedir. Bu dönme olayının son bulması için enerjinin kesilmesi gerekmektedir. Halbuki adım motorlarında durum bundan farklıdır. Bu motora giriş işareti uygulandığı zaman belirli bir miktar döner ve durur. Bu dönme miktarı motorun yapısına göre belirli bir açı ile sınırlandırılmaktadır, rotorun dönmesi girişe uygulanan sinyal sayısına bağlı olarak değişir. Girişe bir tek sinyal verildiğinde rotor tek bir adım hareket ederek dururken, daha fazla sinyal uygulanınca sinyal adedi kadar adım hareket eder ve durur. Örnek olarak  $1,8^{\circ}$  dönen bir motorumuz olsun. İlk işaretin uygulanmasıyla motorun rotoru  $1,8^{\circ}$  döner ve ikinci darbenin gelmesini bekler. Darbe uygulanmazsa dönme olmaz. İşaretlerin arka arkaya uygulanmasıyla da rotor devamlı dönme yapar. Anlatılan bu durum bütün adım motorlarının çalışma prensibini oluşturur. Tipik bir adım motorunun çalışma prensibi aşağıdaki Şekil 1.1’de gösterilmiştir[5].



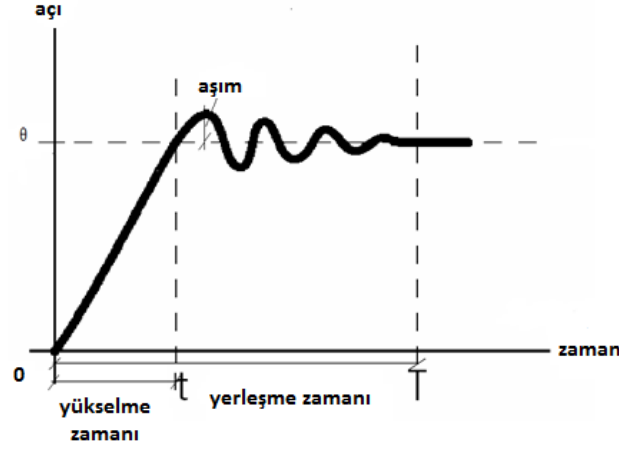
Şekil 1.2. Adım motorunun çalışma prensibi

Yukarıdaki Şekil 1.2'de adım motoru, bir daire içinde elektromanyetik alanların dönüşü ile ifade edilebilir. 1 numaralı anahtar kapandığı zaman sabit mıknatıs kendiliğinden birinci elektromanyetik alan ile aynı hizaya gelecektir. Bundan sonra 1 numaralı anahtar açılıp 2 numaralı anahtar kapatıldığı zaman sabit mıknatıs ikinci elektromanyetik alanın karşısına gelecektir. Bu olaylar sırasıyla tekrarlanırsa sabit mıknatıs yani rotor, bir daire içinde düzgün şekilde dönme prensibine göre çalışır. Adım motoru dinamik açıdan incelendiğinde ise, her bir adımda mekaniki olarak bir denge noktası olduğu görülmektedir. Motor hareket halinde kendisinin ve yükün atalet momentlerinin toplamı olan bir moment ile hareketi gerçekleştirir ve yeni bir denge noktasına ulaşır. Diğer bir ifade ile "x" denge noktasından "x" denge noktasına ulaşmış olur. Motor bu hareketi sırasında boşta ise veya yük ataletinde bir düşüş meydana gelirse rotor yeni denge noktası civarında salınım yapar. Bu olay yüksek hızlarda adım motorunun adım kaymasına veya stabil olmayan kontrol dışı hareket etmesine neden olur. Uygulamalarda adım motoru kullanılırken motor sargılarının endüktansına ve momentine dikkat edilmesi gerekir. Aşırı moment motorun kontrol dışı hareket etmesine neden olacağından moment değeri belirlenirken motorun maksimum momentinin %70 'i ve hatta bu değerden biraz daha küçük bir moment değeri alınmalıdır [6-11].

### **1.1.5. Adım Motorlara Ait Bazı Önemli Kavramlar**

#### **1.1.5.1. Adım Tepkisi / Tek Adım Tepkisi / Cevap Süresi**

Motor fazlarından biri uyarılmış durumdaysa, motor kararlı bir adım konumundadır. Bu fazın uyarıtımı kesilip yeni bir faz uyarılırsa rotor bir adım atacaktır. Rotor konumunun zamana göre değişimi "tek adım tepkisi" olarak tanımlanır. Bir başka deyişle motorun girişine bir komut sinyali uygulandıktan sonra motorun adımlara cevap vermek için gerekli olarak aldığı zamana (T) "tek adım tepkisi", "adım tepkisi" ya da "cevap süresi" denir. Bu süre hem motor parametrelerine hem de motorun sürücü devresine bağlıdır ve yaklaşık olarak milisaniye civarlarındadır.



Şekil 1.3. Adım motoru tek adım tepkisi

Tek adım tepkisi, motorun adım hareketinin hızını, tepkinin aşım ve salınım miktarını, adım açısının hassaslığını veren önemli bir parametredir. Adım motorlarından maksimum performans elde edebilmek için tek adım tepkisindeki aşım ve salınımların azaltılması ve yerleşme zamanının kısaltılması gerekmektedir. Bu nedenle tek adım tepkisinin iyileştirilmesi adım motorlarının kontrolünde çok büyük öneme sahiptir. Motora uygulanan giriş darbeleri ile çıkış hareketi arasındaki senkronizasyonu bozmamak için, sürme esnasında iki darbe arasındaki süre, cevap süresinden daha kısa olmamalıdır. Dolayısıyla adım motorunun cevap süresinin kısaltılabilmesi, motorun daha hızlı adım atabilmesini sağlayacaktır. Adım motorlarının cevap davranışlarında dikkat edilmesi gereken diğer bir nokta da yaptıkları aşımın ve salınımın miktarıdır. Adım motorları, bilgisayar sistemlerinde veya bilgisayarla kontrolü gerektiren hassas sistemlerde kullanıldığında, aşım ve salınım sistemi kötü yönde etkileyen ve önemli boyutta hatalara sebep olan bir davranıştır. Şekil 1.3'te adım motoru tek adım tepkisi gösterilmiştir[12].

#### 1.1.5.2. Adım Oranı

Bir saniyede rotorun yapabildiği adım sayısıdır. Bu adım sayıları, tipik olarak saniyede 300 ila 800 arasındadır. Tezimizde kullanacağımız adım motorun adım sayısı 400 olarak seçilmiştir.

#### 1.1.5.3. Adım Açısı

Motora bir tek sinyal uygulandığında rotorun döndüğü açıdır. Yapımcı firmalar, değişen sayılarda rotor ve stator sargıları ile çeşitli açılarda adım motorları yapmaktadırlar. Bu tezde kullanılan motorun adım açısı  $0.9^{\circ}$  dir.



#### 1.1.5.4. Adım Açısı Doğruluğu

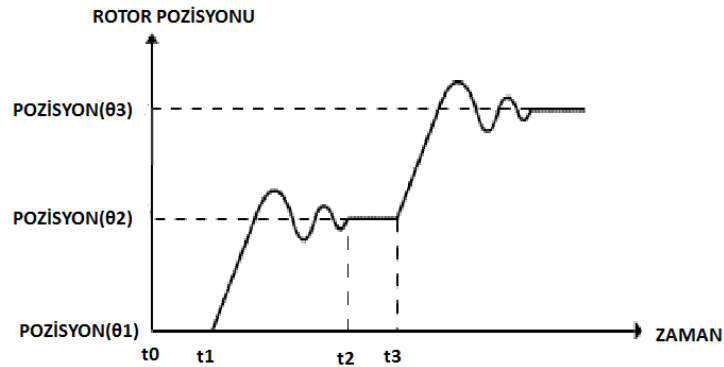
Rotorun yaptığı her bir adımdaki hata miktarını gösterir. Bu parametre genellikle bir yüzde ile verilir. Rotor bir adım döndüğünde meydana gelen toplam hata oranını gösteren bu parametrenin yazılması önemlidir. Bu hata değeri kümülatif (birikimli) değildir. Yani rotorun yaptığı her adım ile bu hata miktarı toplanarak gitmez. Rotorun gerçek hızı adım açısı ve adım hızına bağlıdır.

#### 1.1.5.5. Çözünürlük

Çözünürlük, dönen motorlar için adım açısı(derece), lineer motorlar için ise adım uzunluğu (mm) olarak tanımlanır. Bu sabit değer, üretim sırasında tespit edilen bir büyüklüktür. Bir adım motorunun adım büyüklüğü, çeşitli uyartım metotları ile değiştirilebilir. Mesela yarım adım çalışmada, adım büyüklüğü normal değerinin yarısına indirilir.

#### 1.1.5.6. İleriye Atma (Overshoot )

Adım motora gönderilen sinyallerin frekansı arttırılırsa, adım motorun adımları hızlanır ancak adımlarının hızlanmasıyla adım motor istenmeyen bir karakteristik gösterir. Buna “İleriye atma ” denir. Bu durum aşağıda Şekil 1.4'te görülmektedir. Motorun rotoru “Pozisyon1” ve “ $t_0$ ” anında durmaktadır. “ $t_1$ ” anında uygun stator sargısına güç uygulanmıştır. Bu uygulama rotorun “Pozisyon2” ye dönmesini sağlar. Rotor “Pozisyon2” de tam olarak durmaz çünkü bir dönme momenti vardır. Bundan dolayı dönmek isteyecektir. Rotor biraz daha döner. Buna “İleri atma” denir[13].

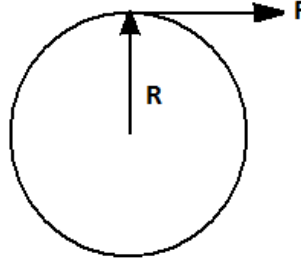


Şekil 1.4. Adım motorun overshoot eğrisi

İki kutup arasındaki magnetik ilişkiden dolayı rotor tekrar geri döner ama yine bir miktar “İleriye atma ” meydana gelir. Rotorun hareketi kesin ve net olmayacağından tam durması istenen yerde duramayacaktır. Kutupların manyetik etkisinden dolayı geri gelip yerinde durması gerekirken fazla dönerek tekrar istenmeyen bir durum meydana gelir. Yani rotor basit harmonic hareket ya da sönümlü salınım yapar. Neticede motora sinyal uygulandıktan sonra rotorun durması bir miktar zaman alır. Rotorun durması için geçen zamana “oturma (yerleşme) zamanı” denir. Bazı devrelerde rotorun durmasının çabuk olması istenir. Bunun yapılabilmesi için bazı işlemlere ihtiyaç vardır. Bu işleme sönüm denir.

#### 1.1.5.7. Adım Motor Moment Parametreleri

$$T_m = F \times r \quad (1.1)$$



Şekil 1.5. Dönme momenti

Moment, elektrik devrelerindeki elektrik potansiyeline, milin açısal hızı da elektrik akımına benzetilebilir. Nasıl elektrik devrelerindeki direnç potansiyel etkisinde akımı sınırlarsa, mekanik devrelerde de sürtünme kuvvetleri milin moment etkisinde açısal hızına sınır getirir. Bir elektrik devresinde akü, potansiyel kaynağıdır aynı şekilde mekanik sistemlerde de motor, moment kaynağıdır. Motor mili, moment etkisinde dönmeye zorlanır. Bir aküden çekilen akımı sıfırdan itibaren arttırsak, akünün iç direncinden dolayı akım arttıkça akü voltajı düşer. Benzer olarak bir motorun milinden alınan devir arttıkça motorun verdiği moment düşer. Elektrik devrelerinde akım ve gerilimin çarpımı, düzgün harekette ise bir nesneye etkiyen kuvvet ile nesnenin hızının çarpımı gücü verirken, motorda moment ile açısal hızın çarpımı mekanik gücü verir.

$$P = I \times V \text{ ( Güç = Akım x Gerilim )} \quad (1.2)$$

$$P = F \times V \text{ ( Güç = Kuvvet x Hız )} \quad (1.3)$$

$$P = T_m \times \omega_n \text{ (Mekanik Güç = Tork x Açısal Hız )} \quad (1.4)$$

$$T_m = F \times r \text{ (Dönme Momenti [Tork] = Kuvvet x Yol) } \quad (1.5)$$

Bir motordan, herhangi bir devirde beklenen momentin büyüklüğü, motorun gücü ile orantılıdır. Sabit güç altında itme kuvveti artarken hız düşer, hız artarken ise itme kuvveti düşer ve dolayısıyla momentte düşer.

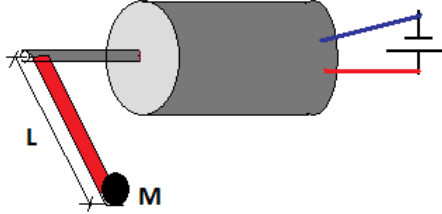
Adım motor kataloglarında dört çeşit moment parametresine rastlanılır. Bu parametreler genellikle moment-hız grafikleriyle birlikte kataloglarda sunulur[14].

#### a. Yüksüz ve Enerjisiz Tutma Momenti

Yüksüz ve enerjisiz tutma moment, motor enerjisiz iken rotor stator dışlarındaki etkileşimden dolayı oluşan moment demektir. Diğer bir ifadeyle sargılara besleme gerilimi uygulanmıyorken, adım motor şaftını döndürmek için gerekli olan maksimum Moment miktarı demektir. Bu parametre adım motor tipine göre değişir.

#### b. Yüksüz ve Enerjili Tutma Momenti

Tutma moment, ya da statik moment olarak da bilinir. Sargılardan sadece birisinden nominal değerde akım akıtılıyorken milden alınan moment demektir. Diğer bir ifadeyle enerjili halde motor duruyorken üretilen maksimum moment miktarıdır.



Şekil 1.6. Adım motorunun basit bir şekli

Tutma moment, bir adım motorunun en temel moment karakteristiğidir ve pratik olarak şu şekilde ölçülebilir. Motor miline Şekil 1.6'da görüldüğü gibi ağırlıksız bir çubuk bağlanır ve çubuğun yatayla sıfır derece açı yapması sağlanır. Motor sargısından nominal değerde akım akıtılır. Şekildeki kırmızı ağırlık çubuğun en ucuna oturtulur ve değişik ağırlıktaki kütlelerle denenerek hangi ağırlıktan sonra motor mili sapma yapıyor tespit edilir.

Çubuğun boyu L ve Kütlenin ağırlığı M olması durumunda;

Tutma Momenti=  $L \times M \times 9,81$  (Yerçekimi ivmesi)

$$(N.m) = (m) \times (Kg) \times (N/Kg) \quad (1.6)$$

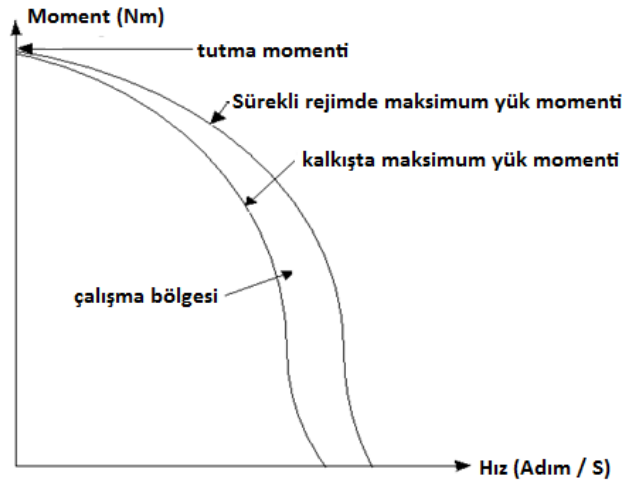
#### c. Kalkıştaki Maksimum Yük Momenti

Özellikle açık döngülü sistemlerde duran bir sistemi istenen pozisyona getirebilmek için motora uygulanan uyartım darbelerinin motor tarafından hiç kaçırılmadan takip edilmesini sağlamak çok önemlidir. Fakat uygulanan uyartım sinyallerin sıklığı, motorun miline bağlı yükü sıfır hızından itibaren kaldırıp hızlandırmasına izin vermeyebilir. Bu yüzden adım motorları için, kalkışta maksimum yük momenti eğrileri tanımlanır.

#### d. Sürekli Rejimdeki Maksimum Yük Momenti

Bu parametre de motor hızına bağlı olarak değişeceğinden sürekli rejimde maksimum yük momenti eğrisinden bahsedilir. Sürekli rejimde maksimum yük momenti/ hız eğrisi herhangi bir sabit dönüş hızında, rotor hareketinin giriş darbe dizisiyle olan senkronizasyonunu bozmadan ve rotorun durmasına neden olmadan sürekli halde motor miline uygulanabilecek maksimum yük momentini verir. Bu moment aynı zamanda, söz konusu hızda motorda meydana gelecek maksimum moment anlamına da gelmektedir[14].

#### 1.1.5.7.1. Moment – Hız Grafikleri



Şekil 1.7. Adım motorun moment-hız grafiği

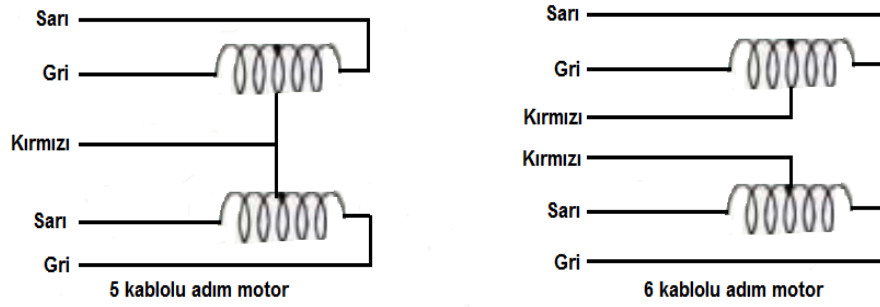
Klasik motorlarda bu eğriye karşılık gelebilecek bir karakteristik yoktur. Maksimum yük momenti eğrisi çalışma noktalarını göstermediği gibi bir transfer fonksiyonu eğrisi de değildir. Sadece, çalışma bölgesini sınırlar. Bu eğrinin sınırladığı bölge içinde herhangi bir noktada motor giriş darbe dizilerini kaybetmeden ve durma tehlikesi olmadan ilgili hız ve yük momenti ile çalışır. Sınırların dışına çıkıldığında bu durum değişebilir. Şekil 1.7'de adım motorun moment-hız grafiği çizilmiştir.

#### **1.1.5.8. Anma Geriliminin Anlamı**

Etikette belirtilen voltaj rotor hareketsizken tutma momentinin oluşturulması için gerekli voltaj değeridir. Bu voltaj değeri etiket akımına bölünürse ilgilenilen fazın sıcak sargı direnci bulunmuş olur. Etiket akımının üstüne çıkılmaması gerekir.

#### **1.1.6. Adım Motorun Uçlarının Bulunması**

Adım motorlarda genellikle 5 veya 6 kablo bulunur. 5 kablolu adım motorlarda bir, 6 kablolu adım motorlarda iki kablo ortak uçtur ve bu uçlar kaynağın pozitif (+) kutbuna bağlanırlar. Kaynağın pozitif (+) kutbuna bağlanacak ortak uçları ölçü aletinin ohm kademesini kullanarak bulmak mümkündür. Ölçü aleti ohm kademesinde iken adım motorun bobin uçlarına bağlı kablolar arasındaki direnç ölçülür. Adım motorlar ister 5, ister 6 kablolu olsun tüm uçlar arasında eşit dirence sahip olan uç ortak uçtur. 6 Kablolu adım motorlarda kablolar üçerli olarak iki gurup halindedir. Her guruptaki bir kablo ortak ucu temsil eder. Ölçüm yapılırken her iki gurup kendi aralarında ölçülerek ortak uç tespit edilir. Bu işlem için ölçü aleti ohm kademesinde iken ilk guruba ait üç kablo ayrı ayrı kendi aralarında ölçülür. Tüm uçlar arasında eşit direnç gösteren uç, ortak uçtur. Aynı işlem ikinci gurup içinde tekrarlanır. Ölçüm sonucunda her iki ortak uca göre iki gurupta da eşit direnç değerleri elde edilir. Bu dirençlerin değerleri her adım motor için farklı olabilir. Adım motorlara ait bobin kabloları farklı renklerle temsil edilir. Bu renkler 6 kablolu adım motorlarda genellikle her gurup için aynı şekilde tekrarlanır.



Şekil 1.8. Farklı yapılarda adım motor

Aşağıdaki Tablo 1.1'de tezimizde kullanmış olduğumuz adım motor için ortak uca göre bobin uçları arasındaki direnç değerleri verilmiştir. Tabloda, “1.Sarı – 1.Kırmızı” arasında karşılıklı olarak 84.5  $\Omega$  (Ohm) ölçülmüştür. Aynı resimde “1.Kırmızı – 1.Gri” arasında da karşılıklı olarak 84.5  $\Omega$  (Ohm) ölçülmüştür. Buna göre 1. Kırmızı ortak uç olarak tespit edilmiş olur. Aynı ölçümler 2. grup uçlar için de tekrarlandığında 2. grubun da ortak ucunun kırmızı olduğu görülür. Gri ve sarı renkli kabloların kendi aralarındaki ölçüm değerleri ise ortak uca göre iki kat yüksektir. Bunun nedeni yukarıdaki resimlerden de anlaşılacağı gibi sarı ve gri renkli kabloların adım motor bobininin iki dış ucuna bağlı olmasıdır.

Tablo 1.1. Adım motorlarda kablo bağlantısı tablosu

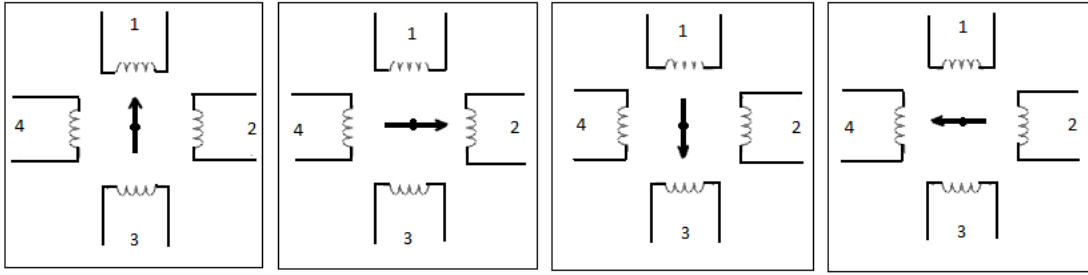
	1.Sarı	1.Kırmızı	1.Gri		2.Sarı	2.Kırmızı	2.Gri
1.Sarı		84.5	116.23	2.Sarı		84.5	116.23
1.Kırmızı	84.5		84.5	2.Kırmızı	84.5		84.5
1.Gri	116.23	84.5		2.Gri	116.23	84.5	

Bir devre üzerinde adım motoru düzgün çalıştırmak için kabloları doğru sıralamada bağlamak gerekir. Ortak uç dışında kullanılan diğer dört ucun kendi arasında bir sırası vardır. Kablo sıralaması bobin uçlarına enerji uygulanarak deneme yanılma şeklinde tespit edilebilir. Eğer kablo bağlantısı yanlış yapılırsa adım motorda bir titreme oluşur ve motor dönmez. Motora adım attırmak için ortak uca motorun cinsine göre pozitif (+) 5V ile 12V arası sabit gerilim uygulanır. Diğer dört uca ise belirli sırada şase (-) uygulanır. Eğer şase potansiyeli bobin uçlarına uygun sıralamada uygulanırsa adım motorda dönme hareketi başlar.

Bobin uçlarının uygun sıralaması Şekil 1.9'daki gibi tespit edilir.

- Ortak uçlara ait kablolar kaynağın pozitif (+) kutbuna bağlanır.
- Adım motorun diğer herhangi bir ucu seçilerek şase uygulanır. Örneğin 1 numaralı bobin ucuna şase uygulanarak sabit bırakılır.
- Başka bir bobin ucu seçilerek şase potansiyeli uygulanır. Eğer adım motor saat yönünde bir adım atıyor ise bu 2 numaralı kablodur.
- Motor saat yönünün tersinde bir adım atıyorsa bu 4 numaralı kablodur.
- Motor hiç hareket etmiyor ise bu 3 numaralı kablodur.

Sonuç olarak adım motoru saat yönünde döndürmek için ortak uca pozitif (+) ; 1, 2, 3 ve 4 numaralı kabloların bağlı olduğu bobinlere sırasıyla şase (-) potansiyeli uygulanmalıdır. Adım motoru saat yönünün tersine hareket ettirmek için ise bobinlere 4, 3,2 ve 1 sıralamasında enerji uygulanmalıdır[15].



1 Şase 1-2 Şase 1-4 Şase 1-3 Şase  
adım saat yönünde adım saat yönünün tersi motor hareketsiz  
Şekil 1.9. Adım motorda uç sırası bulma yöntemi

### 1.1.7. Adım Motorların Çeşitleri

Kullanımda olan birçok elektrik motorunda olduğu gibi adım motorlar da makinanın yapısına ve çalışmasına göre sınıflandırılabilir.

Adım motorlar için ilk günlerden bu yana birçok değişik tasarım yapılmıştır. Bu motorların kayda değer bir moment üretebilmeleri için hem rotor hem de statorları önemli miktarda magnetik akı taşıyabilecek yapıda olmalıdır. Bu nedenle de olabildiğince çok sayıda demir diş ile çevrili olmaları gerekmektedir. Günümüzde hem bu özelliği taşıyıp, hem de ticari bakımdan kullanılabilir nitelikteki adım motorların başlıcaları Değişken Relüktanslı, Sabit Miknatıslı ve Hibrid adım motorlardır. Temelde bunlar aynı türden cihazlardır, ama farklı yapım yöntemlerinden kaynaklanan farklı karakteristikleri belli bir uygulama için adım motor seçilirken önem kazanır. Diğer tür adım motorlar, bu üç tip

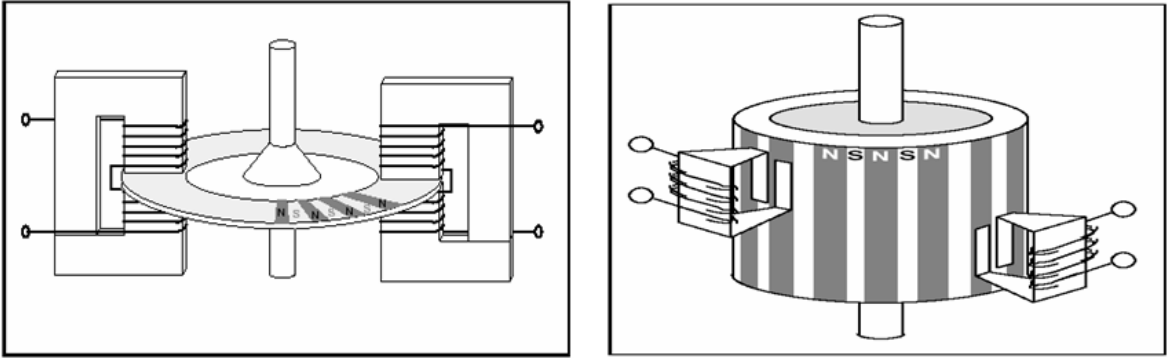
adım motorların varyasyonları olarak tanımlanabilirler.

Adım motorların en belirgin özelliği, girişine uygulanan uyarma darbelerinin, “adım” olarak tam tanımlanmış rotor konumlarına dönüştürmesidir. Rotorun hassas konumlama yapması, hareketli ve sabit parçalar üzerindeki demir dişlerin magnetik olarak karşılıklı gelmesi, dizilmesi ile mümkündür.

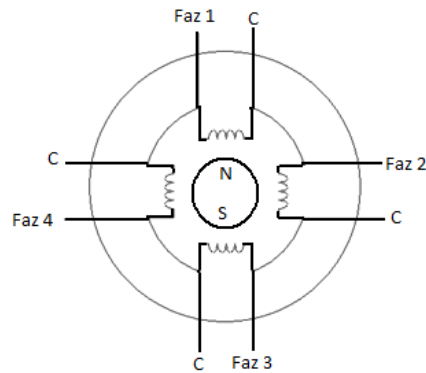
Adım motorlar rotorlarının yapıldığı malzemeye göre sınıflandırılmaktadırlar. Adım motorlar yapılarına göre 3 çeşittir[16]. Bunlar;

### 1.1.7.1. Sabit Mıknatıslı (Permanent Magnet, PM) Adım Motorlar

Sabit mıknatıslı adım motorlardaki rotor, sabit mıknatıstan oluşur. Bu tip motorların rotorlarında diş bulunmaz. Aşağıda Şekil 1.10'da görüldüğü gibi iki farklı yapıda (“disk” ya da “tin-can” yapısı) rotor yapıları mevcuttur. Stator yapıları, değişken relüktanslı adım motorların stator yapılarıyla aynıdır.



Şekil 1.10. Sabit mıknatıslı adım motor



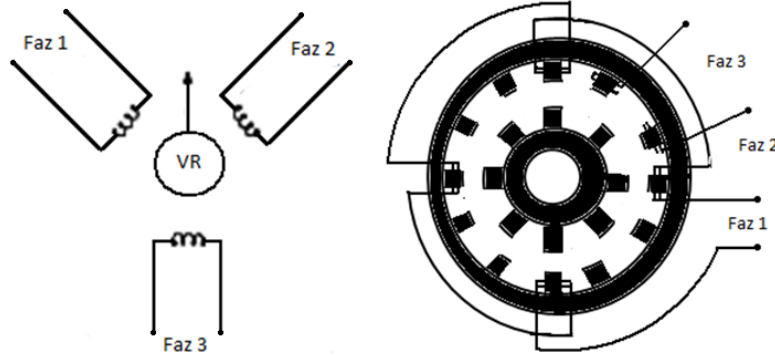
Şekil 1.11. 4-fazlı sabit mıknatıslı adım motoru



Bu motorun statoru çok faz sargılı, rotoru ise daimi mıknatıslıdır. Arasında daimi mıknatısın bulunduğu iki rotor parçası birbirine göre yarım rotor dış mesafesi kadar kaydırılmıştır. Uyarılmamış PM adım motorunun ürettiği moment uyarılmışinkinden daha küçüktür. Motor örneği üstteki Şekil 1.11'de gösterilmiştir. İki kutuplu sabit mıknatıslı rotor, oyuklu 4 kutuplu stator içinde döner. Burada C ile adlandırılan terminal, her bir fazın birer uçlarının birleştirilerek güç kaynağının pozitif ucuna bağlandığı ortak uçtur. Eğer fazlar Faz1, Faz2, Faz3, Faz4 sırasıyla uyarılırsa; rotor saat ibresi yönünde (CW) hareket edecektir.

### 1.1.7.2. Değişken Relüktanslı (Variable Reluctance, VR) Adım Motorları

Değişken relüktanslı adım motorlarında sabit mıknatıslı adım motorlarda olduğu gibi en az dört kutuplu stator bulunur. Sabit mıknatıslı adım motorlarından tek farkı ise rotorun, sabit mıknatıs yerine artık mıknatıslık özelliği göstermeyen olması ve dişler açılmış yumuşak demirden imal edilmesidir. Dişler, silindir eksenine paralel olarak açılmış oluklarla şekillendirilmiştir. Şekil 1.12'de üç fazlı değişken relüktanslı adım motorunun yapısı görülmektedir. Statordaki diş sayısının rotordaki diş sayısından fazla olduğu şekilden görülmektedir.



Şekil 1.12. Değişken relüktanslı adım motorun yapısı

Faz 1'e ait seri bağlı dört sargıya DC gerilim uygulandığında bu sargıların etrafında oluşan manyetik alanlar rotor kutuplarını mıknatıslar ve rotoru bu sargıların karşısına getirecek kadar hareket ettirir. Bu anda diğer kutuplar ise stator ve rotordaki diş sayısı eşit olmadığından stator kutupları karşısında değildir. Bu durum şekilde görülmektedir.

Faz 1 enerjisini kesip faz 2'ye uygularsak bu kez statorda faz 2 bobinleri etrafında meydana gelen manyetik alan kutupları, rotorun faz 1 karşısındaki kutuplarını kendine

çeker. Böylece rotorun dönmesi sağlanır.

Üç fazlı (üç sargılı) sistemlerde rotorun devamlı dönmesi için stator sargıları ard arda enerjilendirilmelidir. Faz 2 enerjisi kesilip faz 3'e uygulandığında bu kez rotor kutupları statordaki faz 3 sargılarının bulunduğu kutupların karşısına gelecek şekilde döner ve durur. Rotorun dönme yönü (saat ibresi yönü ve ya tersi) fazlara uygulanacak gerilimlerin yönüne bağlıdır.

Değişken relüktanslı motorlarda rotor, hafif ve küçük boyutlu yapılıdır. Rotor ölçülerinin küçük olması eylemsizlik momentinin de küçük olmasını sağlar. Bunun sonucu fazlara uygulanan gerilim meydana getireceği moment sebebiyle rotor çok hızlı hareket eder. Değişken relüktanslı motorların harekete başlama, durma ve dönme adımları sabit relüktanslı adım motorlarından daha hızlıdır[17].

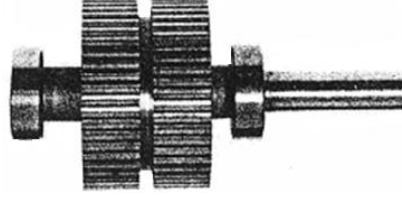
### 1.1.7.3. Karma (Hibrid) Adım Motorları

Karma adım motorlar. (HSM) ; sabit mıknatıslı (PM) ve değişken relüktanslı (VR) adım motorlarının birleştirilip geliştirilmesiyle ortaya çıkmış bir motor türüdür. HSM'ler kullanılacakları sistem ve yük özelliklerine bağlı olarak farklı yapılarda imal edilebilirler. Aşağıda farklı hibrid motor yapıları görülmektedir.



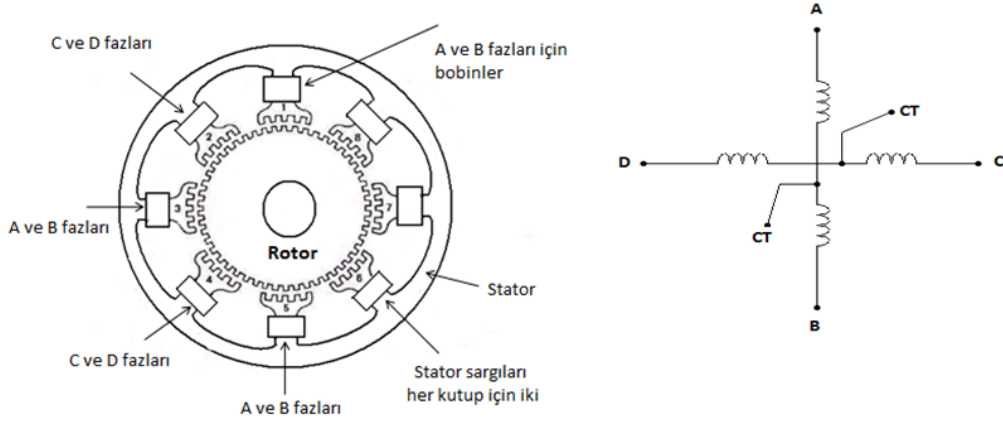
Şekil 1.13. Karma adım motoru şekli

Karma adım motorların rotorunda sabit mıknatıs bulunur. Bu tip motorlara karma denmesinin sebebi, motorun değişken relüktanslı ve sabit mıknatıslı motorların çalışma prensiplerinin bir kombinasyonu ile çalışmasıdır. Yukarıda Şekil 1.13'te karma adım motoru şekli gösterilmiştir.



Şekil 1.14. Karma adım motorunun rotoru

Rotorun her bir ucuna, tipik olarak üzerinde elli diş bulunan dişli türü göbekler vardır. Göbekler amaçlı olarak ayarsız yapılır, böylece mıknatısın kuzey ucundaki dişler, güney ucundakiyle  $180^\circ$  faz dışı kalır. Şekil 1.14'te rotor yapısını net olarak göstermektedir. Karma adım motorlarında tipik olarak 8 stator kutbu bulunur. Her kutupta bulunan diş sayısı iki ila altı arasındadır. Ayrıca rotorun istenen konuma gelmesini sağlamak üzere mıknatıs akısının ilgili kutuplar üzerinden akışını desteklemek veya engellemek amacıyla stator kutuplarına sargılar ilave edilmiştir.



Şekil 1.15. Karma adım motorunun yapısı

Karma adım motorlarında iki farklı sargı kullanılır. Her bir sargı (faz), sekiz stator kutbundan dördünü dolaşır. Şekil 1.15'te A ve B sargıları 1, 3, 5, 7 kutupları üzerinde ise, C ve D sargıları 3, 4, 6, 8 kutuplarındadır. Her faza ait yakın kutuplar birbirleriyle zıt yönde sarılmışlardır. Öncekilerde olduğu gibi, karma adımlı bir motorun rotoru da, stator tahrik akımlarının uygun bir şekilde sıralanmasıyla adım adım hareket ettirilir.

Bu tip motorların rotoru sabit mıknatıslı olduğundan, her zaman için bir tutma veya kalıntı momenti vardır. Motor, düşük hızla ivmelendiği takdirde saniyede 30.000 adıma kadar hızlarda çalışabilir.

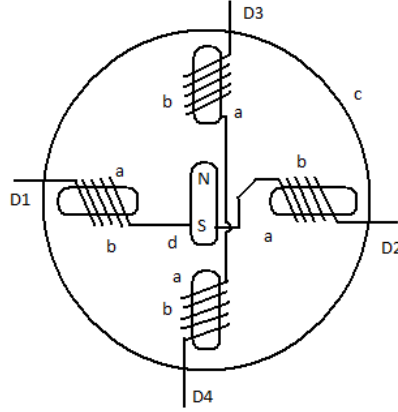
Tüm adım motorları içinde karma tip, her türlü doğrusal ve açısal konumlama

sisteminde en yaygın kullanılan tiptir[18].

### 1.1.8. Bipolar ve Unipolar Adım Motor Ne Demektir?

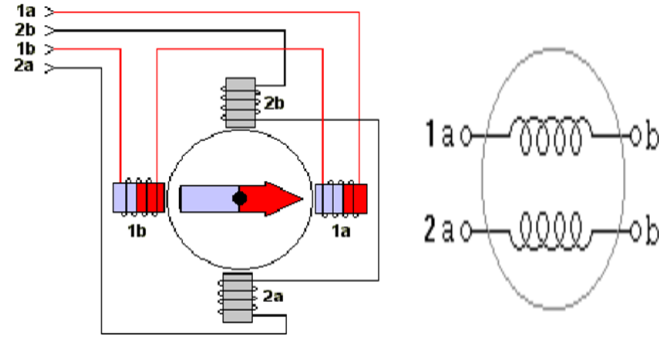
#### 1.1.8.1. Çift Kutuplu ( Bipolar ) Adım Motor

Daha önce söylediğimiz gibi adım motorlarda harekete neden olan, statordaki sargılardaki manyetik alandır. Bu manyetik alan akımın yönüne bağlı olarak değişir. Çift kutuplu adım motorlarının yapısı aşağıdaki gibidir.



Şekil 1.16. Çift kutuplu adım motoru

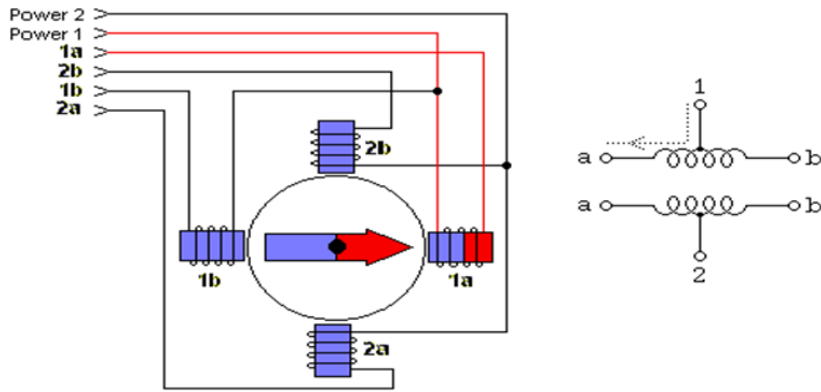
Çift kutuplu adım motorlarında 2 sargıya sırayla ve akım yönleri değiştirilecek güç verilir. Şekil 1.17'yi göz önüne alırsak, 1a (+), 1b (-) kutuplara bağlandığında sargılarda manyetik alan oluşacak ve rotor kendini bu manyetik alana göre pozisyonlayacaktır. 2. durumda 2a (+), 2b (-) kutuplara bağlanırsa bu sefer rotor değişen manyetik alanı takip edecek ve ok şeklindeki uç kısım 2a'ya, arka kısım 2b'ye dönerek yeni pozisyonunu alacak ve motor şaftı 90 derece dönmüş olacaktır. 3. durumda 1b (+), 1a (-) kutba bağlanacak ve hareketin devamı sağlanacaktır. 4. durumda da 2b (+), 2a (-) kutba bağlanınca tam bir tur hareket tamamlanmış olacaktır. Ancak çift kutuplu adım motorlarındaki akım yönünü değiştirme zarureti ve bu sebepten kontrol devrelerinin nispeten karışıklığı tek kutuplu adım motorlarının ortaya çıkmasına neden olmuştur[19].



Şekil 1.17. Çift kutuplu adım motoru modeli

### 1.1.8.2. Tek Kutuplu ( Unipolar ) Adım Motor

Çift kutuplu adım motorlardaki akım yönü değiştirme zaruretine alternatif olarak geliştirilen tek kutuplu adım motorların temel şematik görüntüsü aşağıdaki gibidir.



Şekil 1.18. Tek kutuplu adım motoru modeli

Şekil 1.18'de tipik bir 6 kablolu tek kutuplu adım motoru şeklindedir. Tek kutuplu yapıda bütün sargının üzerinden geçen akımın yönünü değiştirmek yerine sargının ortasından alınan bir uç ile sargıda 2 yönlü akım akışı sağlanmıştır. Yukarıdaki resimde 1 ve 2 numaralı uçlar ortak uçlardır. 5 kablolu tek kutuplu motorların 6 kabloludan tek farkı 1 ve 2 numaralı ortak uçların içeride birleşerek dışarıya tek kablo olarak çıkmasıdır.

Yukarıdaki Şekil 1.18 üzerinden adım adım ilerlersek; 1 ve 2 ortak uçları sürekli pozitif voltaja bağlı olsun. Bu durumda 1a'yı 0 volta çekersek, yukarıdaki Şekil 1.18'de 1a'ya doğru akım geçer ve sargıda manyetik alan oluşur. Bu manyetik alandan etkilenen rotor kendini konumlandırır. 2. durumda 2a'yı 0 volta çekersek bu kez 2a'nın sarılı olduğu sargıda manyetik alan oluşur ve rotor saat yönünde 90 derece döner. 3. durumda 1b'yi ve 4. durumda 2b'yi 0 volta çekersek motor shaftımız 1 tam tur atmış olur.

Tek kutuplu motorlar aynı boyutlardaki çift kutuplu motorlara göre daha düşük momente sahiptir. Tek kutuplu adım motorda çift kutuplu ile aynı alana çift kutupluda kinin aksine 2 sargı sığdırmak zorunda olduğumuz için sargıları oluşturan iletkenler daha incedir. Daha ince iletken daha düşük akım demektir. Daha düşük akım da daha düşük moment anlamına gelir[19].

### 1.1.9. Adım Motorlarının Uyartımı

Tablolarda kullanılan “1” rakamının anlamı o sargının enerjili olduğu, “0” ise sargının enerjisiz olduğudur. Oluşturulan tablolarda adım motorun sargı uçları Bobin4, Bobin3, Bobin2 ve Bobin1 kullanılmıştır.

#### 1.1.9.1. Tek-faz uyartım (1 Fazlı Tam Adımlı Sürüş)

Motor sargılarının sadece birinin uyartıldığı uyartım cinsine tek-faz uyartımı adı verilir. Aşağıdaki Tablo 1.2'de 4-fazlı adım motoru için tek-faz uyartım sırasındaki fazların durumu görülmektedir. Bu uyartım metodunda rotor her bir uyartım işareti için tam adımlık bir hareket yapmaktadır. Uyartım dönüş yönüne bağlı olarak sıra ile yapılır.

Tablo 1.2. Adım motoru tek-faz uyartım tablosu

Adım	Bobin4	Bobin3	Bobin2	Bobin1
a.1	1	0	0	0
a.2	0	1	0	0
a.3	0	0	1	0
a.4	0	0	0	1

#### 1.1.9.2. İki-faz Uyartım (2 Fazlı Tam Adımlı Sürüş)

Motor sargılarının ikisinin sıra ile aynı anda uyartıldığı uyartım cinsine iki-faz uyartımı adı verilir. Aşağıdaki Tablo 1.3'te 4-fazlı adım motoru için iki-faz uyartım sırasındaki fazların durumu görülmektedir. İki faz uyartımda rotorun geçici durum tepkisi tek-faz uyartımlıya göre daha hızlıdır. Fakat burada güç kaynağından çekilen güç iki katına çıkmaktadır.

Tablo 1.3. Adım motoru iki-faz uyartım tablosu

Adım	Bobin4	Bobin3	Bobin2	Bobin1
b.1	1	1	0	0
b.2	0	1	1	0
b.3	0	0	1	1
b.4	1	0	0	1

### 1.1.9.3. Karma Uyartım (2 Fazlı Yarım Adımlı Sürüş)

Bu uyartım yönteminde tek-faz uyartımı ile iki-faz uyartımı ard arda uygulanır. Burada rotor her bir uyartım sinyali için yarım adımlık bir hareket yapmaktadır. Aşağıdaki Tablo 1.4'te fazların uyartım sırası görülmektedir. Bu uyartım metodunda adım açısı yarıya düştüğünden adım sayısı iki katına çıkmaktadır.

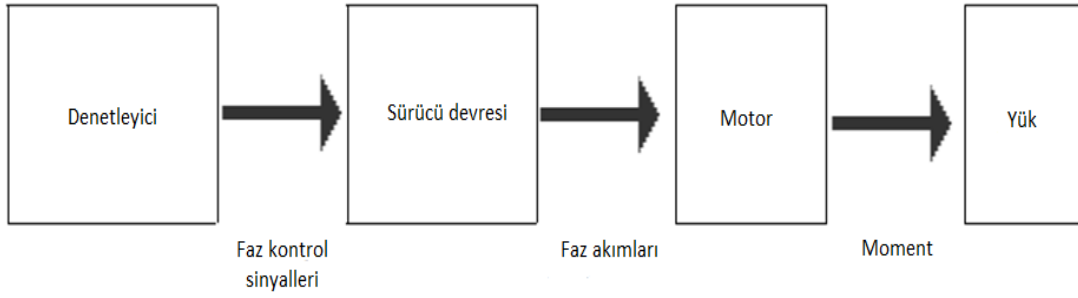
Tablo 1.4. Adım motoru karma uyartım

Adım	Bobin4	Bobin3	Bobin2	Bobin1
a.1	1	0	0	0
b.1	1	1	0	0
a.2	0	1	0	0
b.2	0	1	1	0
a.3	0	0	1	0
b.3	0	0	1	1
a.4	0	0	0	1
b.4	1	0	0	1

### 1.1.10. Adım Motorlarının Denetimi

#### 1.1.10.1. Açık Döngü Denetim

Şekil 1.19'da açık döngü denetim için blok diyagramı görülmektedir. Sayısal kontrol işaretleri denetleyici tarafından üretilir ve sürücü devre tarafından yükseltilip adım motorunun sargularına uygulanır. Eğer denetleyici olarak mikroişlemci veya bilgisayar kullanılırsa bu elemanların getirdiği esnekliklerden dolayı aynı denetleyici ile farklı adım motorları kontrol edilebilir. Kontrol edilecek adım motorları 3, 4 veya daha farklı faz sayısına sahip olabilir. Ayrıca kullanılacak uyartım metodu için tek-fazlı, iki-fazlı veya yarım adım uyartımlarından herhangi biri seçilebilir. Bu uyartım metodlarından hangisinin kullanılacağı daha önce de açıklandığı gibi motorun kullanılacağı sisteme bağlıdır.

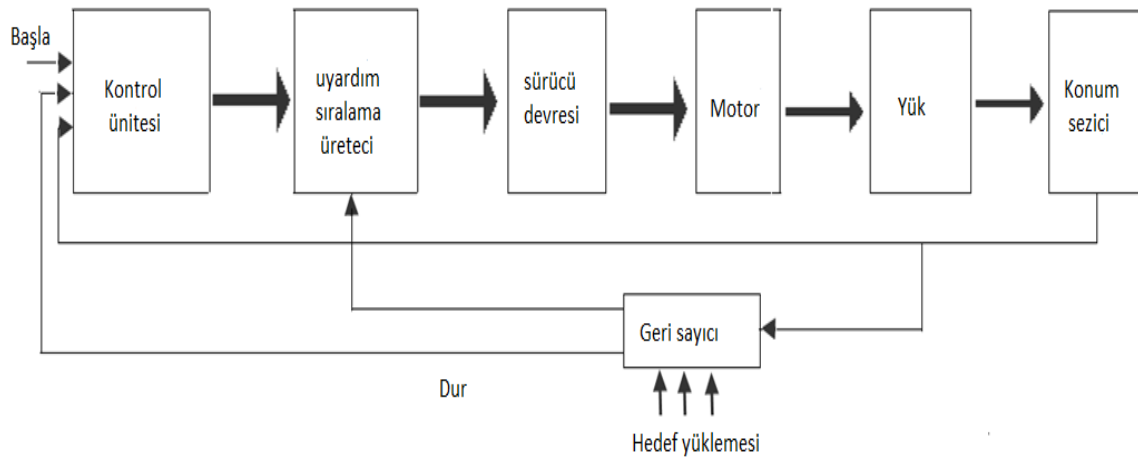


Şekil 1.19. Adım motoru açık döngü denetimi blok diyagramı

Denetleyici tasarlanırken motorun cinsi ve yükün durumu göz önünde bulundurulmalıdır. Bu sırada meydana gelen sınırlamalar kalıcı veya geçici durum sınırlamaları olabilir. Açık döngülü denetimde motorun konumu bilinmediğinden dolayı motorun gönderilen bütün adım komutlarını yerine getirdiği varsayılmaktadır. Eğer uyarım hızı çok yüksek ise, motor adım komutlarından bir kısmını yerine getiremeyebilir. Bu durumda kalıcı bir hata meydana gelir. Bu tür hataların meydana gelmemesi için motor yükünün en büyük olduğu durum göz önüne alınarak hata yapılmayan en yüksek hız belirlenip, bu hızın üzerindeki hızlarda uyarım yapılmamalıdır.

### 1.1.10.2. Kapalı Döngü Denetim

Kapalı döngü sistemlerde ani rotor konumu sezilerek denetim birimine iletilir. Her adım komutu için bir önceki komutun gerçekleştirildiği adım bilgisi alınarak uygulanır. Bu nedenle motor ile denetleyici arasında herhangi bir adım kaybı olmaz. Kapalı döngü denetime bir örnek aşağıdaki Şekil 1.20'de gösterilmiştir.



Şekil 1.20. Adım motoru kapalı döngü denetimi blok diyagramı



İlk olarak geri sayıcıya hedef konum yüklenir. Daha sonra başla komutu verilerek adım komutlarının sıralayıcıya uygulanması sağlanır. Adım komutlarına bağlı olarak motor adım hareketi yapmaya başlar. İlk adım tamamlanınca, konum sezici geri sayıcıyı ve denetim birimlerini uyarır ve geri sayıcı değeri bir azalır. Eğer bu denetim açık döngülü yapılırsa, geri sayıcı adım komutlarının sayısını yine saklar fakat komutun uygulanıp uygulanmadığı bilinmez. Konum sezici, denetim birimine yeni adım komutu üretimi için işaret gönderir. Ağır yükler için adım komutları arası sürenin daha büyük olması nedeniyle adım komutlarının ard arda gelmesi istenmez.

Yüke göre hız ayarlaması yapılır ve motor hedef konuma gelene kadar bu olaylar tekrarlanır. Adım motoru hedef konuma gelince denetim birimi dur komutu ile uyarılarak yeni adım komutu üretilmesi engellenir. Kapalı döngü sistemi, adım motorunun yük durumunu da göz önüne alarak uyarım sürelerini ayarlar ve en uygun hız profilinde çalıştırılır. Şekil 1.20'de kapalı döngü denetim için blok diyagramı görülmektedir[20].

#### **1.1.11. Adım Motor Sürücü Sistemleri**

Bir adım motorunun adım atabilmesi için motor fazlarına uygulanacak olan darbeler genelde bir kontrol devresi tarafından üretilmektedir. Böyle bir lojik devre TTL (Transistör-Transistör-Lojik ) elemanlarından oluşmaktaysa devreden en fazla 5V ve 18mA 'lık çıkışlar alınabilmektedir. Bununla birlikte tipik bir adım motorunun uyarım devresi 5V ve 3A'dır. Buradan da görülebileceği gibi kontrol katı ile motor katı arasında kuvvetlendirme yapabilecek bir başka katın daha bulunması gerekmektedir. Klasik çift kutuplu transistörleri kullanarak birkaç sürücü katı ile ya da alan etkili güç transistörleri kullanarak bir kat için sürücü devresi tasarlamak mümkün olabilir.

Bu durumda adım motorlarının çeşidine göre sürücü devresi tasarlamak gerekmektedir. Bu yüzden adım motoru sürücü devreleri sabit olmayıp motorun yapısı ve çeşidine göre değişkenlik göstereceğinden temel mantıklarıyla günümüzde çok kullanılan sürücü devrelerine burada değinilecektir[21].

##### **1.1.11.1. Temel Sürücüler**

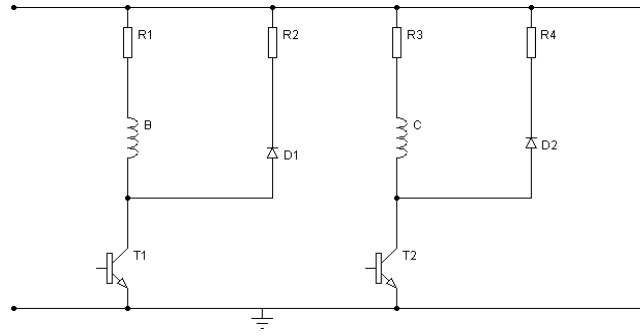
Değişken relüktanslı adım motorlarının uyarılması için fazlardan sadece tek yönde akım akitilir. Dolayısıyla tek yönlü sürücü devreleri yeterli olmaktadır. Ancak karma ya da sabit mıknatıslı adım motorlarında fazlardan her iki yönde de akım akmalıdır. Bu amaçla iki yönlü sürücü devreleri kullanılmalıdır. Bazı tür adım motorlarında ise basitlik sağlamak

amacıyla fazlar birbirine ters yönde sarılmış ikişer sargıdan oluşur. Bu sargıların her birinden yine değişken relüktanslı adım motorlarında olduğu gibi tek yönde akım akıtılır. Bu yüzden bu motorlarda her iki yönde tek yönlü akım akıtan sürücü devreleri kullanılır. Görüldüğü gibi her motor yapısı için çeşitli sürücü devreleri kullanılmaktadır.

Buradan hareketle temel sürücü devreleri genel olarak;

#### 1.1.11.1.1. Tek Kutuplu Sürme

Bu tip sürücü devreleri yalnızca değişken relüktanslı adım motorlarını sürmek için kullanılır. Aşağıda ise yalnızca bir fazı içeren sürücü devre şekli görülmektedir. Şekil 1.21 incelenecek olursa, anahtarlama transistörü ya doyumda ya da kesimdedir. Doyumda olduğu zaman kaynak geriliminin tamamı faz sargısı ve zorlama direnci üzerine düşeceğinden, direnç yeterli sargı akımını verecek büyüklükte seçilmelidir.

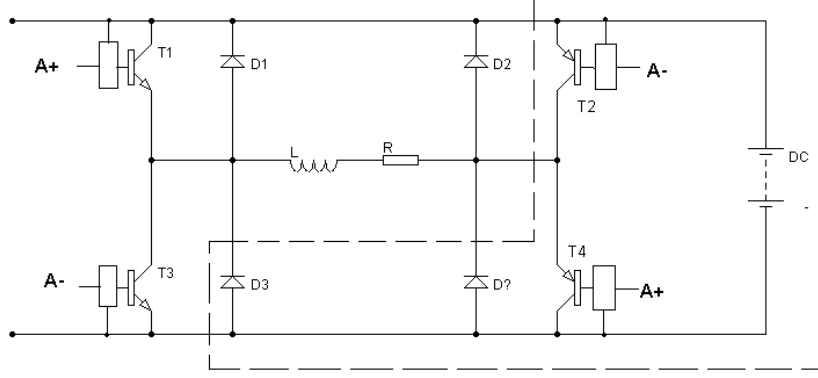


Şekil 1.21. Tek kutuplu sürücü devresi

Burada  $R$  zorlama direnci,  $r$  sargı direnci ve  $I$  sargı akımıdır. Zorlama direncinin görevi ise; Genelde faz sargıları önemli büyüklükte indüktansa sahiptir. Bu nedenle zaman sabiti büyük olacak ve geniş hız bölgesinde çalışmaya engel olacaktır. Çünkü faz sargılarına gerilim uygulandığında sargıdan geçen akımın yeterli düzeye erişmesi zaman alacaktır. Bu zamanı düşürmek için sargıya seri direnç bağlanır. Göz önünde tutulması gereken diğer bir nokta ise anahtarlama transistörü tetiklenirken emiter ve kolektör uçları arasında sürücü devreye zarar verecek nitelikte gerilimler oluşur. Bunu önlemek için söndürme direnci ve söndürme diyotu devreye eklenir. Buna ek olarak sargılardan akım akarken faz sargılarında biriken enerji bu iki öğeyi de içine alan bir yol izleyerek eritilir [21].

### 1.1.11.1.2. Çift Kutuplu Sürme

Karma ya da sabit mıknatıslı adım motorlarının sürülmesinde kullanılan bu devrenin bir fazı için gerekli olan donanım Şekil 1.22’de verilmiştir.



Şekil 1.22. Çift kutuplu sürücü devresi

Buradaki zorlama direncinin ( $R$ ) görevi tek kutupluda olduğu gibi zaman sabitini küçültmek içindir. Sargıdan geçen akımın yönünü değiştirmek için dört adet transistör kullanılır. Aynı anda transistörlerden yalnız ikisi doyumdadır. Akım yönü değiştirilmek istendiğinde doyumdaki transistörler kesime, kesimde olan transistörler ise doyuma ulaşır. Bu arada sargılarda biriken enerjinin bir kısmı besleme kaynağına geri döner.

Çift kutuplu sürmede söndürme akımı tek kutupluya göre daha hızlı akar ve bu akım için ayrıca söndürme direnci bağlanmasına gerek yoktur [21].

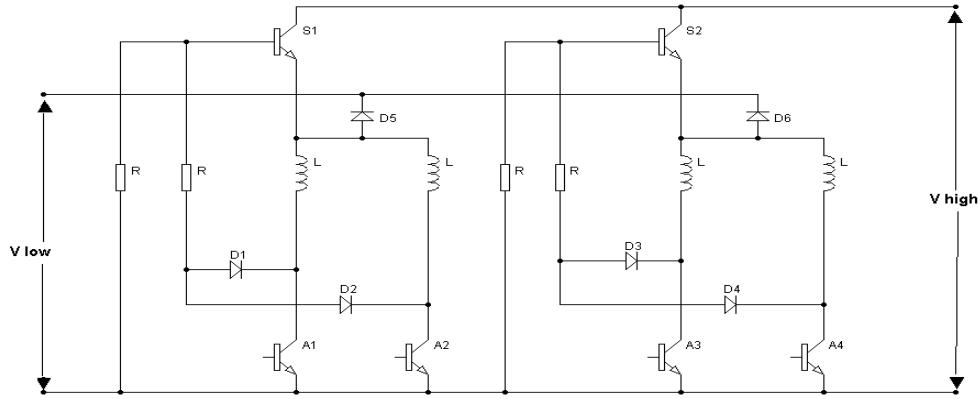
### 1.1.11.1.3. İki Düzeyli Sürme

Daha verimli bir çalışma yöntemi, başlangıçta yüksek gerilimli bir kaynağı yalnız sargılarda akım oluşturmak için kullanmak ve ardından da düşük bir gerilimi devreye sokmaktır. Bu çalışmaya çift gerilim veya iki seviyeli çalışma denir. Adım motoru çift gerilim çalışmada seri dirençli çalışmaya göre daha fazla performans göstermektedir.

Adım motoruna yol verildiği zaman hem faz hem de yüksek gerilim anahtarı kapanır. Akım istenilen bir değere ulaştıktan sonra veya sabit bir zaman aralığından sonra yüksek gerilim anahtarı açılır ve akım alçak gerilim anahtarı üzerinden sağlanır.

Akım, motorun hızı arttıkça yüksek gerilim kaynağının devrede kalma süresi de artar. Bir noktadan sonra yüksek gerilim kaynağı sürekli devrede kalır. Eğer sargı akımının kaynağı yeteri kadar sönmeyeceği hızlara erişirse demir nüvenin doyması akımın sargıda kalmasına neden olur. Böylece indüktansın değeri düşer. Bu durumda akım daha düşük

empedans göreceğinden nominal akımın birkaç katına çıkar. Böyle bir durumdan bir akım sezicinin kullanılmasıyla yüksek gerilimin devrede kalma süresi kontrol edilerek kurtulunur. Böylece indüktans değişimleri ve zıt emk yüzünden oluşan akım dalgalanmalarının ortaya çıkaracağı zararlar önlenmiş olur. Şekil 1.23'te iki seviyeli sürücü devresi gösterilmektedir [21].



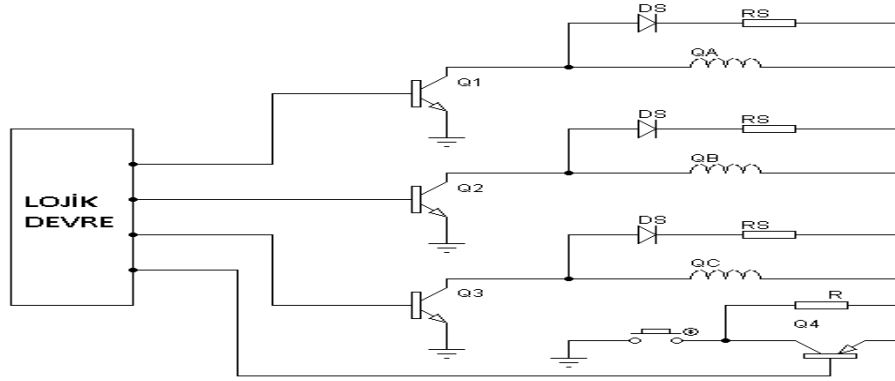
Şekil 1.23. İki seviyeli sürücü devresi

#### 1.1.11.1.4. L / R Sürme

Çok yaygın kullanılan bir sürücü tipi olup belli bir sabit mıknatıslı adım motorundan daha yüksek çıkış elde etmek için ucuz bir çözümdür. Bu tip sürücüde sargılara seri bir direnç eklenir. Eklenen direnç ise kontrol gerilimini artırarak sargılardan daha yüksek değerde akım elde etmek için kullanılır. Seri dirençler istenilen  $L / R$  oranını verecek şekilde seçilmeli ve dirençler sargı dirençlerinin üç katı veya  $(L / 4 \times R)$  oranını verecek şekilde olmalıdır ve besleme voltajı sabit akım oranını vermesi için dört kat arttırılmalıdır. Eğer arttırılmış olan güç yoğunluğu devre için sınırlanıyorsa iki seviyeli güç veya kıyıcı güç kaynağı kullanılmalıdır [21].

Seri dirençli sürücülerin bir dezavantajı verimliliğinin bir hayli düşük olmasıdır. Çünkü gücün bir kısmı dirençler üzerinde ısıya dönüşmektedir. Bunun için gerekli direnç sayısı, kaynağa giden ortak dönüş yoluna tek bir direnç daha bağlanarak en aza indirilebilir.

Seri dirençler kondansatörlerle şöntlenerek geçici rejimden direnç etrafında alçak empedanslı yol oluşturulabilir. Bunun neticesinde momentte artma sağlanır. Verim ise direncin bir transistor ile paralel bağlanması sonucu arttırılabilir. Şekil 1.24'te seri direnci transistör ile şöntlenmiş sürücü devresi gösterilmektedir.



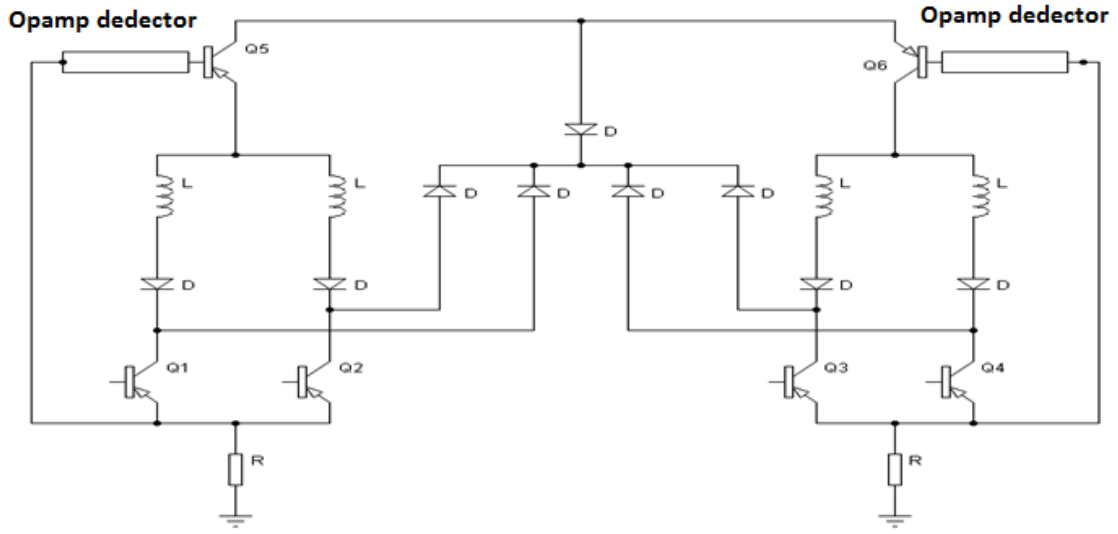
Şekil 1.24. Seri direnci transistör ile şöntlenmiş sürücü devresi

#### 1.1.11.1.5. Kıyıcı Sürme

Bir başka akım sınırlayıcı sürücü Şekil 1.25'te adım motorunun sargısının ortalama akımını, darbe genişlik veya frekans modülasyonunu kullanarak kontrol eden tiptir. Bu yöntemde yüksek gerilim anahtarlayan transistör akım sezici devre tarafından kontrol edilmektedir. En basit bir kontrol şekli istenilen bir akım değerine ulaşıldığında transistörün kesime gitmesi ve akımın düştüğü sırada transistörün iletme geçmesidir. Böylece istenilen ortalama akım sağlanabilecektir. Bu tip kıyıcıların frekansı sargı endüktansı ve zıt emk'sı ile değişmektedir.

Başka bir kıyıcı ile kontrol sistemi de kıyıcı frekansının veya gerilimin kontrol edilmesiyle gerilimin devrede kalma süresinin değiştirilmesi ve akımının kontrol edilmesidir. Frekans ise  $50\text{ Hz}$  ile  $25\text{ KHz}$  arasında herhangi bir değerde olabilir.

Ses gürültülerinin önemli olduğu yerlerde en yüksek frekans tercih edilmelidir. Normal olarak akımı kıymak için akım sezici ile kontrol edilen ayrı bir transistor kullanılır. Kıyıcı kontrolün üstünlüklerinden biri de gerilim kaynağında meydana gelebilecek bazı değişmeler veya sargı direncinin değişmesi, kompanze edilebilir. Kıyıcıyı düşük akımlara programlayarak motor dururken güç tasarrufu yapılmış ve ısı kaybı önlenmiş olur. Şekil 1.25'te kıyıcı sürme devresi gösterilmektedir [21].

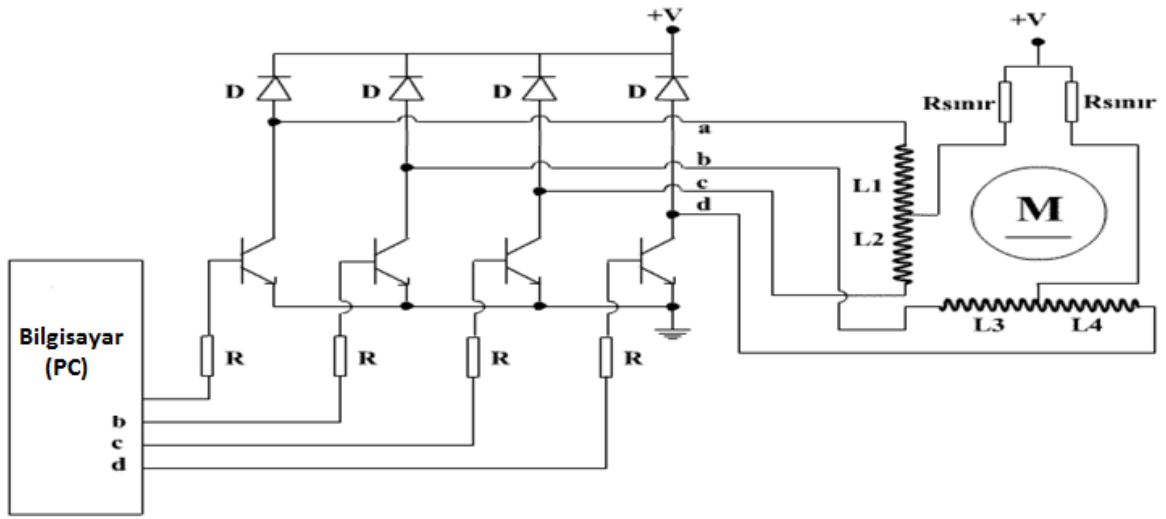


Şekil 1.25. Kıyıcı sürücü devresi

Gerek kıyıcı gerekse çift gerilimli sürücüde seri dirençte ısı kaybı olmaz. Çünkü seri direnç yerine anahtarlama transistörü kullanılmıştır. Kıyma sonucu moment büyür, bunun yanında yüksek hızlarda anahtarlama kayıplarından dolayı güç kaybı diğer sürücülere nazaran yüksek olur.

#### 1.1.11.2. NPN Transistörlü Unipolar Adım Motor Sürücü Devresi

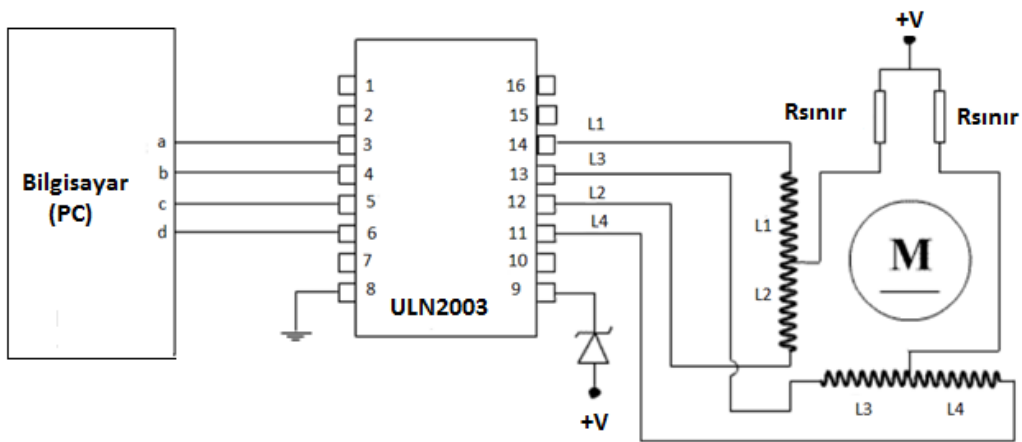
Devrenin amacı, motorun ortak kablolarına, motor çalışma gerilimini uyguladıktan sonra, motorun diğer uçlarını, belli bir sırada toprağa göndererek, bir adım hareketi elde etmektir. Transistörün base ucuna, bilgisayardan +5V geldiğinde, transistör tetiklenir ve Collector-Emetor iletme geçer. Dolayısıyla transistörün collector ucuna bağlı kablolar toprağa ulaşır.



Şekil 1.26. NPN transistörlü unipolar adım motor sürücü devresi

Şekil 1.26 devresindeki diyotlar, motorun üretmiş olduğu ters emk'dan, transistörleri korumak için kullanılmışlardır. Yani, motor adım attıktan sonra oluşacak ters emk, bu diyotlar sayesinde absorbe edilir. Kullanılan diyotun değerinin herhangi bir önemi yoktur. Devredeki NPN transistörler anahtarlama elemanı olarak kullanılmışlardır ve bu transistörlerin modelleri, motor için gerekli olan akıma bağlıdır. Yaklaşık 5 Amper civarı akımlar için TIP141 modeli, daha küçük akımlar için ise BC108 veya BC547B tipi transistörler kullanılabilir. Transistörler BC547 seçilirse  $R=4.7K$  ve  $D=1N4001$  seçilebilir. Motor akımını sınırlamak için,  $R_{sınır}$  direnci motor sargılarına seri olarak bağlanabilir[22].

### 1.1.11.3. ULN2003 Entegresi ile Unipolar Adım Motor Sürücü Devresi

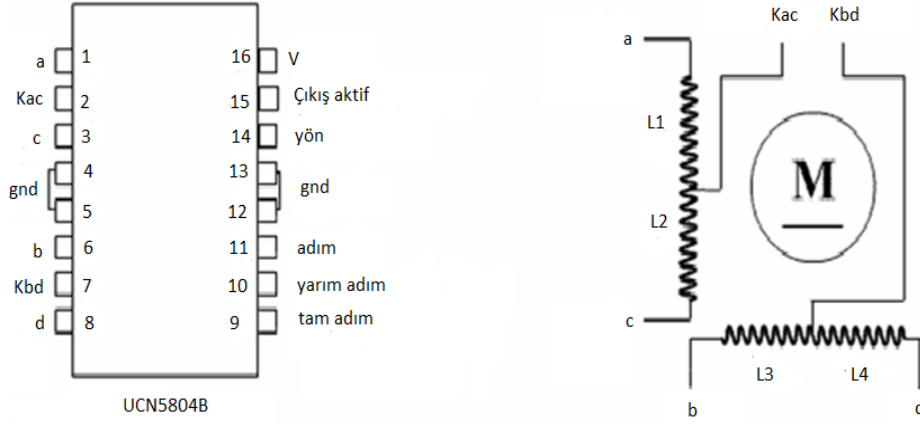


Şekil 1.27. ULN2003 entegresi ile unipolar adım motoru sürücü devresi

Adım motor sürücü devresi olarak kullanılan ULN2003 entegresi içerisinde 7 adet NPN transistor ve dâhili diyot bulunmaktadır. Bir önceki devrede transistör bacaklarıyla uğraşmak yerine, bu entegreyi kullanmak oldukça kolaylık sağlayacaktır. Şekil 1.27'deki devreyi kurduktan sonra, bilgisayardan herhangi bir bacağa +5V uyguladığımızda, karşısındaki bacak toprağa ulaşacaktır. ULN2003 entegresi ile 500mA kadar akım çeken motorlar çalıştırılabilir. +12V'a kadar entegreye besleme gerilimi verilebilir.

#### 1.1.11.4. UCN5804B Entegresi ile Tek Kutuplu Adım Motor Sürücü Devresi

UCN5804B entegre devresi küçük tek kutuplu adım motorları sürmek için üretilmiştir. Şekil 1.28'deki entegre ile +35V gerilime kadar çalışan ve 1.25A akım çeken adım motorlar kontrol edilebilir. Entegre devre çıkışları adım motora direk olarak bağlanabilir. Buna ilaveten entegre devrenin adım girişi, yarı adım girişi ve yön girişleri de bulunmaktadır. Motorun dönmesi için entegrenin adım girişine sinyal verilmesi gerekmektedir. Yön girişi lojik 0 olunca motor bir yöne, lojik 1 olunca ise motor diğer yöne dönmektedir[22].



Şekil 1.28. UCN5804B entegresi ile tek kutuplu adım motoru sürücü devresi



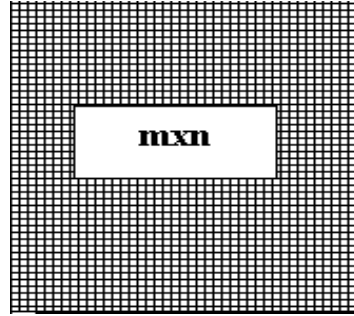
## 2. HAREKETLİ NESNE TAKİP YÖNTEMLERİ

Literatürde çok değişik nesne takip yöntemleri mevcuttur. Güvenilir ve verimli bir takip işlemi gerçekleştirmek açısından farklı uygulama alanları için nesneyi en iyi şekilde temsil edecek modelin seçilmesi gerekir. Takip uygulaması için nesne temsili, takip edilecek parametreler hakkında bilgi sahibi olduğu için önemlidir.

### 2.1. Dijital Görüntünün Temel Kavramları

#### 2.1.1. Görüntü Alımı

Sayısal bir görüntü, nesnelere tarafından yansıtılan ışık enerjisinin bir algılayıcı tarafından öngörülen elektromanyetik aralıkta algılanarak sayısal işaret haline dönüştürülmesi ile oluşur. Bir görüntünün temel bileşeni pikseldir. Sayısal bir görüntü Şekil 2.1'de olduğu gibi  $m \times n$  boyutlu piksellerden oluşan bir matris ile ifade edilir. Gri tonlu görüntülerde, görüntü farklı gri ton değerlerinden oluşur.

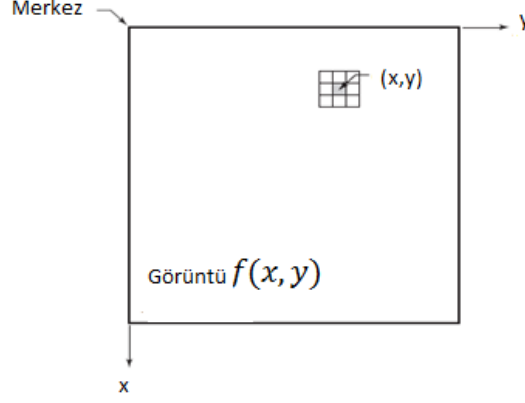


Şekil 2.1. Pikseller

Gri değer aralıkları  $\{0,1,2, \dots,255\}$  şeklinde ifade edilir. Bunun anlamı, bir gri tonlu görüntüde 256 tane farklı gri ton değeri daha doğrusu gri değerler bulunabilir. Sıfır gri değeri kural olarak siyah renge, 255 gri değeri beyaza karşılık gelir. Bu değerler arasında ise gri tonlar oluşur. Elektromanyetik spektrumda  $0,4-0,5\mu\text{m}$  dalga boyu mavi renge,  $0,5-0,6\mu\text{m}$  dalga boyu yeşil renge,  $0,6-0,7\mu\text{m}$  dalga boyu kırmızı renge karşılık gelir. Bu dalga boylarında elde edilmiş üç gri düzeyli görüntü bilgisayar ekranında sırası ile kırmızı-yeşil-mavi (RGB) kombinasyonunda üst üste düşürülecek olursa renkli görüntü elde edilmiş olur. Bu sistemlerde kamera ile görüntü dış ortamdan alınır, görüntü işleme teknikleri ile işlenir, yorumlanır ve yapılması gereken işleme karar verilir[23].

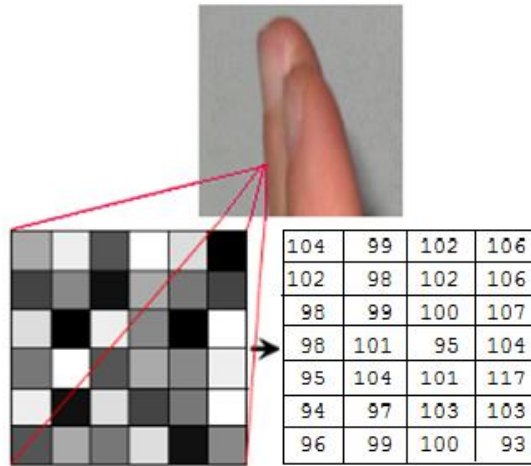
### 2.1.1.1. Piksel

Piksel kare şeklinde görüntünün en küçük birimidir. Aşağıda Şekil 2.2'de bir fotoğraf çerçevesindeki bir pikselin koordinatları gösterilmiştir.



Şekil 2.2. Bir fotoğraf çerçevesindeki bir pikselin koordinatları

Sayısal görüntüler yan yana gelen pikseller topluluğundan oluşmaktadır. Aşağıda Şekil 2.3'te görüntü ve piksel değerleri serisi gösterilmiştir. İki boyutlu uzayda sayısal görüntü satır ve sütunların kullanıldığı matris şeklinde tanımlanır. Matrisin her bir elemanına satır ve sütun kesişim noktalarında  $a(x, y)$  şeklinde ulaşılır. Pikselin kendi başına en ve boy değerleri yoktur. Dikdörtgen biçimindeki tek bir piksel 1x1 mm, 1x1 cm ya da 3x2 m olabilir. Tersi belirtilmedikçe piksellerin en ve boy oranı eşittir.



Şekil 2.3. Görüntü ve piksel değerleri serisi

Kendi başlarına boyut sahibi olmayan piksellerin boyutu belirlendiğinde uzunluk biriminde kaç piksel bulunacağı da belirlenmiş olur. Kapladığı alan ne olursa olsun görüntü birimi olan her piksel sadece tek bir renk değeri içerebilir. Piksel tabanlı sayısal görüntü işleme programları görüntü üzerinde işlem yaparken inç ve santimetre değerini kullanmazlar. Tüm kesme, yapıştırma, yer değiştirme işlemlerinin tamamı piksellere uygulanır. Vektörsel görüntü dosyalarından farklı olarak noktasal görüntü dosyaları için çözünürlük ve içerdiği

piksel sayısı çok önemlidir. Dokümandaki işlemlerin hepsi piksellerin rengi veya konumunu değiştirerek etki yapar. İşlemlerin ne kadar süreceği ise üzerinde işlem yapılan piksel sayısına ve işlemin karmaşıklığına bağlıdır. Sayısal bir görüntü aşağıdaki şekilde ifade edilmektedir[23].

$$g(x, y) = T(f(x, y)) \quad (2.1)$$

#### 2.1.1.2. Çözünürlük

Çözünürlük boyut tanımlamalarında ek olarak gereken bir kavramdır. Çözünürlük, uzunluk biriminde birbirinden ayırt edebilen nokta sayısıdır. Çözünürlük hesaplarında uzunluk birimi olarak inç veya santimetre kullanılır.

Çözünürlük; bir devirdeki adım sayısı veya dönen motorlar için adım açısı (derece), lineer motorlar için ise adım uzunluğu  $\square mm \square \square$  olarak tanımlanır. Bu sabit değer, üretim sırasında tespit edilen bir büyüklüktür.

Bir adım motorunun adım büyüklüğü, çeşitli kontrol düzenleri ile değiştirilebilir. Yarım adım çalışmada adım büyüklüğü normal değerinin yarısına indirilir. Bu yöntem sırasıyla iki faz ve tek faz uyarımlarla uygulanabilir. Ayrıca motorun sahip olduğu faz sayısına göre yarım adım çalışma için değişik uyarma şekilleri de saptanır. Bir diğer yöntem ise bir faz uyarılmış durumda iken diğer fazlardan akıtılacak akımların seviyelerinin kontrol edilmesiyle adım büyüklüğünün çeşitli değerlerde tutulmasıdır[24].

#### 2.1.2. Dijital Görüntü

Resim üzerinde bilgisayarla işlem yapabilmek için  $f(x, y)$  görüntü fonksiyonun koordinat sisteminin ve genliğinin dijitize edilmesi gerekir.  $(x, y)$  düzlemsel koordinat sisteminin sayısallaştırılmasına görüntü örnekleme, genliğin sayısallaştırılmasına gri ton ayıklaştırılması adı verilir. Bu durumda dijital resim hem düzlemsel koordinatların, hem de parlaklığın ayrık biçimde olduğu bir görüntü fonksiyonudur.

$$f(x, y) \cong \begin{pmatrix} f(0,0) & \dots & f(0, M-1) \\ \vdots & \ddots & \vdots \\ f(N-1,0) & \dots & f(N-1, M-1) \end{pmatrix} \quad (2.2)$$

(2.2) denkleminde  $f(x, y)$  fonksiyonu, sürekli bir resmi göstermektedir. Denklemin sağ tarafı, bu resmin yaklaşığıdır ve dijital resim olarak adlandırılır. Bu dijital resmi, satır ve sütun indislerinin resimdeki bir noktanın koordinatlarını, bu noktadaki elemanın da gri ton değerini gösterdiği bir matris olarak ( $i = 0, 1, \dots, N-1; j = 0, 1, \dots, M-1$ ) aşağıda Şekil 2.4'te olduğu gibi düşünebiliriz. Bu şekildeki bir görüntü düzleminin elemanlarına piksel adı verilir [25]. Sayılaştırma işleminde  $N, M$  değerlerini ve ayrık gri ton sayısını belirlemek gerekir.

	1	2	3	...	M
1					
2		(x-1,y-1)	(x-1,y)	(x-1,y+1)	
3		(x,y-1)	(x,y)	(x,y+1)	
⋮		(x+1,y-1)	(x+1,y)	(x+1,y+1)	
N					

Şekil 2.4. Görüntünün Koordinat Sistemi

Gri tonlarının sayısına  $G$  diyelim. Bilgisayar uygulamalarında bu değerle genellikle  $2^n$ 'nin pozitif tamsayı kuvvetleri olarak seçilir:

$$N = 2^n, M = 2^m, G = 2^k$$

Buradaki gri ton değerlerinin  $[0, L]$  aralığında ve eşit aralıklı ayrık değerler olduğunu kabul ediyoruz. Resimler bilgisayar ortamında işlem gördüğüne ve saklandığına göre, bir resmi saklamak için  $N * M * k$  bit hafızaya ihtiyaç vardır. Görüntü matrisi sürekli bir fonksiyonun yaklaşığı olduğundan iyi bir benzerlik için matrisin boyutlarının ve gri ton sayısının ne olması gerektiği sorunuyla karşılarız. Bu parametreler büyüdükçe dijital resim orijinal resme 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.

### 2.1.3. Siyah – Beyaz Görüntü

Siyah beyaz bir görüntü, ışık yoğunluğunu gösteren iki boyutlu bir fonksiyon olarak düşünülebilir. Bu  $f(x, y)$  fonksiyonunda  $x$  ve  $y$  düzlemsel koordinatları, fonksiyonun

$(x, y)$  noktasındaki değeri resmin o noktadaki parlaklığını gösterir. Siyah beyaz bir resmin  $(x, y)$  noktasındaki parlaklığına resmin bu noktadaki gri ton değeri adı verilir. Teoride  $x, y \in R$ 'dir ve  $f(x, y)$  sıfırdan büyük, sonlu bir sayıdır.  $f(x, y)$ 'nin iki bileşeni vardır:

- 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:

$$f(x, y) = i(x, y).r(x, y) \quad (2.3)$$

Burada teorik olarak

$$0 < i(x, y) < \infty \quad (2.4)$$

ve

$$0 < r(x, y) < \infty \quad (2.5)$$

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 absorbe edilmesi, 1 limit değeri ışığın tamamen yansıtması anlamına gelir. Aydınlanma ortalama olarak açık bir günde 9000, bulutlu günde 1000, tipik bir ofiste 100 foot – kandeladır [54].  $f(x, y)$  fonksiyonun  $(x, y)$  noktasındaki değerine  $\lambda$  dersek (2.3), (2.4) ve (2.5) denklemlerinden  $\lambda$ 'nın

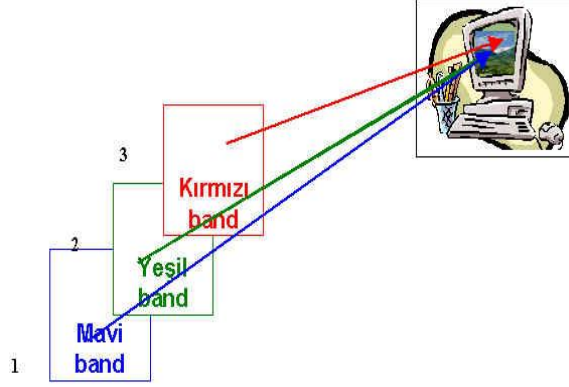
$$L_{min} < \lambda < L_{max} \quad (2.6)$$

(2.6) denkleminde olduğu gibi belirli bir aralıkta değiştiğini görürüz. Teoride tek gereklilik  $L_{min}$ 'in pozitif olması ve  $L_{max}$ 'in sonlu olmasıdır. Uygulamalarda,  $L_{min} = i_{min}.r_{min}$  ve  $L_{max} = i_{max}.r_{max}$  yazarsak ve yukarıdaki değerleri göz önünde alırsak, kapalı mekanlar için  $0.005 L_{min} \cong$  ve  $100 L_{max} \cong$  bulunur.  $[L_{min}, L_{max}]$  aralığına gri ton ölçeği adı verilir. Pratikte bu aralık  $[0, L]$  aralığına kaydırılır. Burada  $\lambda = 0$  siyah ve  $\lambda = L$  beyaz olarak kabul edilir. Aradaki değerler grinin tonlarına karşılık gelir [26].

#### 2.1.4. Renkli Görüntü

Görüntü işlemede renkli resim kullanımının çeşitli avantajları vardır. Otomatik görüntü analizinde renk, nesnelerin tanımlanmasını ve çevrelerinden izole edilmesini kolaylaştıran güçlü bir araçtır. Ayrıca, insan gözü yaklaşık iki düzine gri tonu ayırt edebilmesine karşılık binlerce renk tonu ve yoğunluğu ayırt edebilir.

Renkli görüntü;

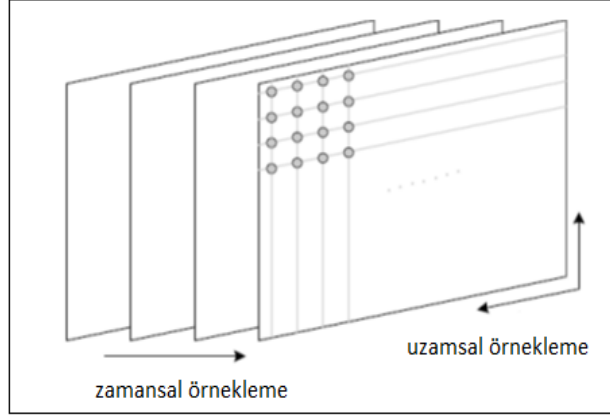


Şekil 2.5. Görüntünün Kırmızı(R), Yeşil(G), Mavi(B) renk kodları

Renkli görüntüler bilgisayar ekranlarında 24 bit lik veri olarak görüntülenir. Görüntüleme yukarıda ki Şekil 2.5'te görüldüğü gibi 3- R(Kırmızı), 2- G(Yeşil), 1- B(Mavi) kodlanmış aynı objeye ait üç adet gri düzeyli görüntünün üst üstle ekrana iletilmesi ile oluşur[27].

## 2.2. Video Kodlama

Video kodlama işlemi sayısal video sinyalinin sıkıştırılması ve çözülmesi işlemidir. Diğer bir ifadeyle sayısal video doğal görsel sahnelerinin uzamsal ve zamansal olarak örneklenip gösterilmesi işlemidir. Örnekleme sonrasında ya bir çerçeve ya da bir alan oluşturulmaktadır. Uzamsal örnekleme sonrasında durağan tek bir çerçeve oluşmaktadır. Çerçeve ise nokta adını verdiğimiz resim bileşenlerinin/piksel bir araya gelmesiyle oluşur. Zamansal örnekleme ise uzamsal örnekleme bileşenlerinin periyodik olarak tekrar etmesidir. Uzamsal örnekleme için  $t$  anındaki resmin örneklenmesi, zamansal örnekleme içinse farklı  $t$  anlarındaki resimlerin örneklenmesi tanımlanabilir[28]. Zamansal ve uzamsal örnekleme ile oluşturulan çerçeveler Şekil 2.6'da gösterilmektedir.



Şekil 2.6. Video dizisinde uzamsal ve zamansal örnekleme

Sayısal bir görüntü  $f(x,y)$  şeklinde tanımlanmaktadır  $y= f(x, y)$  Yukarıdaki ifade de  $x, y$  görüntü üzerindeki konumsal koordinatlar ve  $f$  fonksiyonunun değeri ise parlaklık değeri olarak isimlendirilmektedir.

Bir görüntü  $x-y$  koordinatları boyunca sürekli olabilir. Aynı durum  $x, y$  koordinatlarında olduğu gibi grilik seviyesinde de olabilir.  $1 \times N$  boyutlu matris satır vektörü,  $M \times 1$  boyutlu matris ise sütun vektörüdür.  $1 \times 1$  matris ise scalardır.

```
>> f = imread('frame1.jpg');
```

 (4.6)

```
>> size(f)
```

 (4.7)

```
ans=
```

```
1024 1024
```

Bir pikselin iki temel özelliği söz konusudur:

```
>> [M,N] = size(f);
```

 (4.8)

```
>> M = size(f,1);
```

 (4.9)

Matlab ortamında görüntüyü ekranda görüntülemek için aşağıdaki fonksiyon kullanılır. Syntax'ı ;

```
>> imshow(f)
```

 (4.10)

Low ile belirtilen değerinden küçük ve eşit olan parlaklık değerlerini 0'a; high ile belirtilen değerden büyük veya eşit olan parlaklık değerlerini ise 1'e eşitler. Arada kalan değerlerde ise değişiklik olmaz.

```
>> imshow(f, [low high])
```

 (4.11)

imtool, görüntü üzerinde daha fazla etkileşim kurma imkanı sağlar.

```
>> imwrite(f, 'filename')
```

 (4.12)

Matlab ortamında görüntüyü saklamak için aşağıdaki fonksiyon kullanılır. Syntax'ı

;

Veri türleri arasındaki dönüşümün gerçekleştirilmesi için aşağıdaki yazım şekli (syntax) kullanılmaktadır:

$$B = \text{class\_name}(A) \quad (4.13)$$

Toolbox fonksiyonu olan `mat2gray` herhangi bir veri türünde olan görüntüyü  $[0 \ 1]$  aralığında olacak şekilde `double` türünde olan bir görüntüye dönüştürür. Kullanımı `im2bw` fonksiyonu `binary` bir görüntü oluşturur. Kullanım şekli aşağıdaki gibidir:

$$\gg g = \text{im2bw}(f, T) \quad (4.14)$$

Eşik değeri  $[0,1]$  aralığındadır.

Görüntü matrisinde sahip olduğu matris koordinatlar.

### 2.3. Görüntü Dosyaları

Görüntü işlemenin ilk adımı, görüntülerin alınıp sayısallaştırılmasıdır. Sayısallaştırılmış görüntüler bilgisayar ortamında hareketli görüntü (video) ya da resim dosyası biçiminde saklanır.

#### 2.3.1. Vektör ve Piksel Grafikler

Görüntü dosyaları iki şekilde depolanabilir. Eğriler, alanlar ve dolduruldukları renkler olarak saklanıyorsa bu tür görüntü dosyalarına vektör tabanlı ya da vektörel dosya denilir. Vektörel programlarla bir çizgi veya eğim oluşturulduğunda, başlangıç ve son nokta arasında matematiksel bir ifade oluşturulur. Açılar ve uzaklıklar bu eşitlikte değişkenlikler olarak yer alır. Çizilen şekil yeni bir boyuta getirildiğinde, değişkenlere büyütme oranı girilir ve şekil kayıp olmaksızın yeniden oluşturulur. Resim noktalar topluluğu olarak oluşturulmuşsa bu tür resimlere piksel tabanlı görüntü denilmektedir. Piksel tabanlı görüntülerin dosya boyutu büyütme oranı arttıkça artmaktadır.

### 2.5. Görüntü Histogramı

Histogram, görüntü üzerindeki piksellerin değerlerinin grafiksel ifadesidir. Buna görüntü histogramı veya gri-düzey histogramı denir. Görüntü histogramı, görüntünün her bir noktasındaki piksellerin tespiti ile bu piksellerin sayısının ne olduğunu gösterir. Bu sayede histogram üzerinden görüntü ile ilgili çeşitli bilgilerin çıkartılması sağlanır. Görüntü üzerindeki piksellerin nerede yerleştiği tam olarak çıkartılamaz. Fakat görüntünün aydınlık-karanlık bölge değerlerinden görüntü hakkında genel bilgiler elde edilebilir.



Uygulanmak istenen eşik değerleri tahmin edilebilir[29].

Gri ton değerleri  $[0, L - 1]$  aralığında değişen bir dijital görüntünün histogramı,

$$P(r_k) = n_k / N \quad (4.16)$$

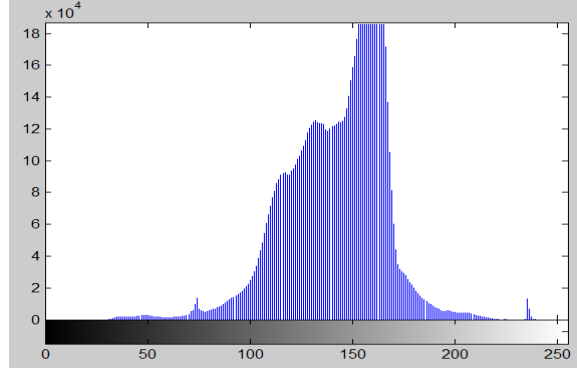
Ayrık fonksiyonudur. Burada  $r_k$  gri ton değerini,  $n_k$  görüntüde bu ton değerinden piksellerin sayısını,  $N$  görüntüdeki toplam piksel sayısını gösterir ( $k=0, 1, 2, \dots, L-1$ ) [54]. Bu şekilde tanımlanmış  $p(r_k)$  fonksiyonu, gri ton değerlerinin görüntüde bulunma olasılıklarının hesabını verir. Uygulamalarda histogramın tanımı çeşitli şekillerde basitleştirilebilir. Örneğin  $r_k$  değeri hemen her zaman doğrudan  $k$  tamsayısı olarak alınır. Ayrıca görüntü histogramı gösteren tablolarda çoğu zaman normalize edilmiş değerler (olasılıklar) yerine her bir gri ton değerinden piksellerin sayısı gösterilir. Şekil 2.7'de örnek bir görüntü ve Şekil 2.8'de bu görüntünün histogramı görülmektedir. Görüntü histogramı görüntünün genel görünümü hakkında bir bilgi verir. Bu görünüm, histogramda piksellerin yoğunlaştığı bölgelere bakarak çeşitli şekillerde yorumlanabilir. Bir görüntünün histogramının oluşturulması için kullanılan fonksiyon, Yazım şekli;

$$h = imhist(f, b) \quad (4.17)$$

Burada  $f$ , orijinal görüntü;  $b$  ise parlaklık ölçeğinin bir alt bölümüdür. Eğer  $b$  değeri açık bir şekilde belirtilmezse 256 varsayılır.



Şekil 2.7. Dijital görüntü



Şekil 2.8. Görüntünün histogramı

Histogram, görüntünün içerdiği piksel değerlerinin ağırlığını belirten grafiksel bir gösterimdir. Eğer pikseller histogramın küçük gri ton sayılı değerlerini gösteren bölgelerinde yoğunlaşıyorsa görüntünün genel olarak karanlık olduğu sonucuna varılabilir. Pikseller yüksek gri ton değerlerinde yoğunlaşıyorsa çok aydınlık ve beyazın ağırlıklı olduğu bir görüntü söz konusudur. Pikseller histogramın orta bölgelerinde dar bir alanda yoğunlaşıyorsa görüntüye gri ton hakimdir. Bu durumlar, kontrast geliştirme veya histogram işleme tekniklerinin uygulanabilirliği konusunda fikir verir. Ancak, histogramın verdiği bilgilerin sınırlarına dikkat etmek gerekir. Örneğin histogram, görüntünün içeriğine ait herhangi bilgi taşımaz. Ayrıca histogramın verdiği bilginin, görüntünün bütününe ait olduğunu unutmamak gerekir. Histogram, her bir gri ton değerinden piksellerin sayısı hakkında bilgi veremez. Bu yüzden görüntünün bütünüyle değil de sadece belli bir bölgesiyle ilgileniyorsak, bu bölgeye ait histogramı hesaplamak gerekecektir [29-32].

## 2.6. Nesne Takibinde Önemli Rol Oynayan Kriterler

Nesne takibi işlemine başlamadan önce hedef nesneye, ortama ve uygulamaya göre belirlenmesi gereken bazı kriterler vardır. Bunlar temel olarak nesne gösterimi, takip amacıyla kullanılan özellik ve kullanılacak nesne takibi metodudur.

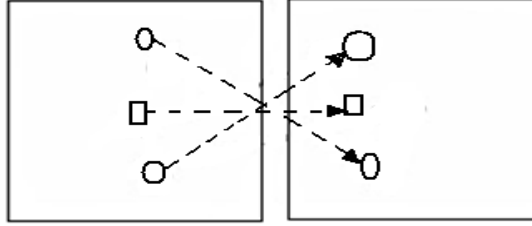
### 2.6.1. Nesne Gösterimleri

Nesne takibi aşamasında önemli rol oynayan seçimlerden biri takip edilecek nesnenin sembolize edileceği şekildir. Uygulamanın amacına göre takip edilecek nesne değişiklik göstermekte, takip edilecek nesneye göre de ilgili gösterim seçilmektedir. Örneğin trafikte bir araç, denizde bir gemi, havada bir uçak, yolda bir insan, sahada bir top takip edilebilir.

Nesneyi sembolize etmek için, takip edilecek nesnenin şekil ve hareket yapısına uygun olan bir şekil seçilir [33].

### 2.6.1.1. Noktalar ile Gösterim

Şekil 2.9'daki gibi arka arkaya gelen görüntülerde, bulunan nesnelere noktalar ile temsil edilir. Nesne, merkezinde yer alan bir nokta ile veya noktalar kümesi ile gösterilir. Nokta gösterimi görüntüde az yer kaplayan nesnelere takip etmek için uygundur [33].



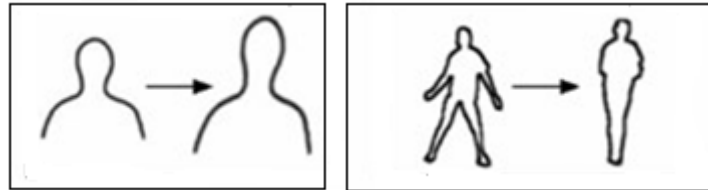
Şekil 2.9. Nokta takibi

### 2.6.1.2. Geometrik Şekil ile Gösterim

Nesne dikdörtgen, elips gibi bir geometrik şekil ile gösterilir. Bu şekillerle gösterilen nesnenin hareketi genellikle ötelemeli, doğrusal veya izdüşümsel olmaktadır. Geometrik şekil ile gösterim genellikle katı, esnek olmayan nesnelere uygun olmakla birlikte, esnek nesnelere takip etmek için de kullanılır [33-35].

### 2.6.1.3. Nesne Silueti ve Çevre Çizgisi ile Gösterim

Çevre çizgisi nesnenin sınır çizgilerini temsil eder. Çevre çizgisinin içinde kalan bölge ise nesnenin silueti olarak tanımlanır. Siluet tabanlı metotlar nesnelere kesin şekil tanımlamaları sunar. Aşağıda Şekil 2.10'da çevre çizgisi oluşturma örneği gösterilmiştir. Çevre çizgisi ve siluet gösterimlerinin karmaşık ve esnek yapıda nesnelere takibinde kullanımı uygundur [36].



Şekil 2.10. Çevre çizgisi oluşturma örnekleri[12].

#### 2.6.1.4. Eklemlı Şekil Modelleri ile Gösterim

Eklemlı nesnelere birbirine eklem ile bağlanan parçalardan oluşur. Örneğın insan; gövdesi, kolları, bacakları, kafası, elleri ve ayaklarının eklemlerle birbirine bağlanmasından oluşan bir canlıdır. Bu gösterimde parçalar arası ilişkiler kinematik hareket denklemleri ile çıkartılabilir. Eklemlı bir nesnenin gösteriminde, nesnenin eklemler ile bağlanan her bir parçası silindir veya elips ile ifade edilebilir [36].

#### 2.6.1.5. İskelet Model ile Gösterim

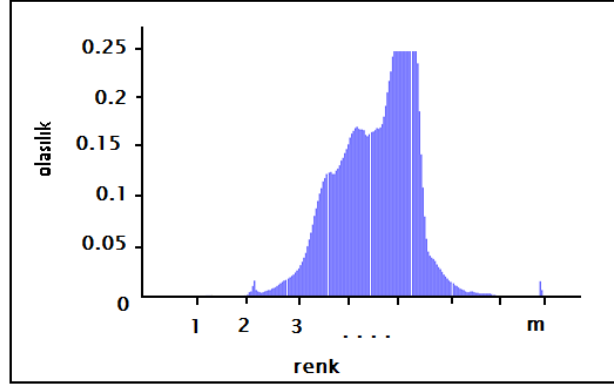
Nesnenin iskeleti, nesnenin silüetine orta eksen dönüşümü uygulanarak çıkarılabilir. Bu model genellikle nesne tanımada şekil gösterimi olarak kullanılır. İskelet gösterimi, hem eklemlı hem de esnek yapılı nesnelere gösterimi için kullanılabilir.

### 2.7. Özellik Seçimi

Nesne takibinde özellik seçimi kritik bir rol oynar. Seçilecek özelliğın veya özelliklerin takip edilecek nesneyi ayırt edici kılması gerekir. Görsel özelliklerin kullanılmasındaki en önemli sebep de her bir nesnenin kendine özel görsel özellikleri olmasıdır. Özellik seçimi, nesne gösterimi ile yakından alakalıdır. Örneğın histogram tabanlı gösterimlerde renk özelliğı kullanılırken, çevre çizgisi tabanlı gösterimlerde kenar özelliğı kullanılmaktadır. Çoğru nesne izleme algoritması bu tarz kombinasyonlar kullanmaktadır [36-37].

#### Renk:

Bir nesnenin rengi ışık kaynağının ışık tayfının güç dağılımı ve nesnenin yüzey yansıma özellikleri olmak üzere iki esas fiziksel faktörden etkilenir. Şekil 2.11'de tek boyutlu uzaya indirgenmiş renk olasılık dağılımı görülmektedir. Görüntü işleme çalışmalarında, rengi temsil etme amacıyla genellikle RGB (Red, Green, Blue – Kırmızı, Yeşil, Mavi) renk uzayı kullanılır. Bunun yanında  $L^*u^*v$  ve HSV (Hue, Saturation, Value – Renk tonu, Doygunluk, Değere) uzayları da kullanılmaktadır. Nesne takibinde hangi renk uzayının kullanılmasının daha verimli olduğuna dair kesin bir karar yoktur.



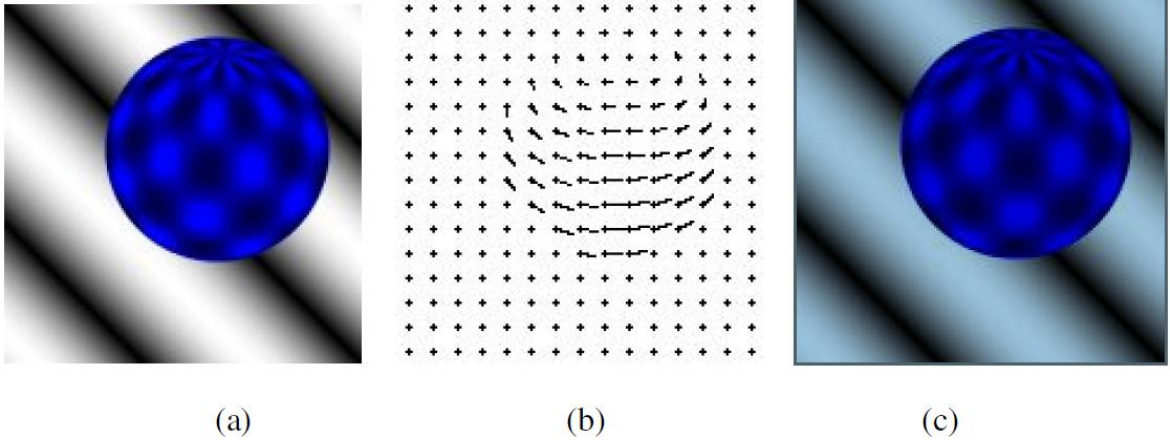
Şekil 2.11. Tek boyutlu uzaya indirgenmiş renk olasılık dağılımı

#### Kenar:

Nesne dış çizgileri görüntü yoğunluğunda genellikle büyük değişimlere yol açar. Kenarların en önemli özelliği renk özellikleri ile kıyaslandığında ışık değişimlerine daha az duyarlı olmasıdır. Nesnelerin dış çizgisini takip eden algoritmalar özellik olarak genellikle kenar özelliğini kullanırlar. Sonuçlarının basitliği ve kesinliği sebebiyle en çok tercih edilen kenar bulma yaklaşımı Canny Kenar Bulma yöntemidir [38].

#### Optik Akış:

Optik akış, bir bölgedeki her bir pikselin ötelemesini tanımlayan yer değiştirme vektörlerinin yoğunluk alanıdır. Arka arkaya gelen görüntüdeki eş piksellerin parlaklıkları kullanılarak hesaplanır. Optik akış, hareket tabanlı bölütleme ve takip uygulamalarında sıklıkla kullanılan bir özelliktir. Yoğunluk optik akışı hesaplamak için kullanılan popüler yöntemlerden bazıları: Horn ve Schunck , Lucas ve Kanade , Black ve Anandan , Szeliski ve Coughlan . Şekil 2.12’de (a) resmindeki topun (c) resmindeki yere gelirken hareket değişimi (b) resminde görüldüğü gibi olmuştur.



Şekil 2.12. Optik akış (a) 1. Görüntü (b) Yön doğrultuları (c) 2. Görüntü.

Doku:

Doku, pürüzsüzlük ve düzenlilik gibi özellikleri ölçen yüzeyin yoğunluk değişim ölçüsüdür. Renk ile karşılaştırıldığında doku tanımlayıcılarını oluşturabilmek için filtre uygulama gibi ön işlemlere ihtiyaç duyar. Kenar özellikleri gibi doku özellikleri de ışık değişimlerine daha az duyarlıdır[38].

## **2.8. Nesne Takibi Metotları**

### **2.8.1. Tekli ve Çoklu Nesne Takibi**

Nesne takibinin amacı ilgilenilen nesnelere içeren ortamdan alınan video verilerini toplayarak bu nesnenin ortamdaki sonraki konumunu tahmin etmek için kullanılmaktadır. Bir nesnenin kapalı döngü takibini yapmak için tasarlanmış tekli nesne takibi ve çoklu nesne takip sistemleri arasında temel bir fark vardır. Tekli nesne takibi sistemlerinde gözlenen veri doğrudan dinamik sistemin doğrudan aktif durumunu tahmin eden takip filtrelerine, örneğin Kalman filtresi, doğrudan giriş olarak verilebilir. Tekli nesne takibi algoritmasının geliştirilmiş şekli olan çoklu nesne takibi algoritmalarında ilgilenilen nesneye ait gözlemleri kategorize etmek için karmaşık veri işleme sistemlerine ihtiyaç vardır. Alınan veriler doğrudan takip filtresine verilmez bir çok ön işlemden geçirilir.

Görsel takipteki en belirgin özellik durumun olasılık yoğunluğunu ya da şeklin belirsizliğini modellemeye dayanır. Algoritmaların çoğu Kalman filtresine dayalı durum uzayı tanımını kullanır [39].

Kalman filtresi gürültülü gözlem bilgisi ile önceki duruma bağlı olarak yapılan tahmini birleştiren bir mekanizmaya sahiptir. Kalman filtresi gözlemler ve durumların kovaryans matrisleri ile durum belirsizliğini giderir ve gürültü gaussian dağılımlı olduğunda iyi çalışır. Eğer sistem çok dağılımlı ve gürültüde bazı engeller varsa Kalman Filtreleme algoritması kullanılamaz. Başka bir nesne takibi yöntemi olan Condensation algoritması da bir simülasyon yöntemidir ve görüntüde büyük görsel engeller olduğunda da iyi çalışır [39]. Çoklu hedef takibi için birçok algoritma geliştirilmiştir [40-44]. Bu algoritmaların çoğu çoklu hedef takibi söz konusu olduğunda Çok Hipotezli Takip algoritması kullanılır [44-45].

### **2.8.2. Kalman Filtresi ile Nesne Takibi**

Nesne takibi amacıyla kullanılan yöntemlerden en bilineni Kalman filtresidir [45]. Bu yöntemin uygulamasının basit olmasından ve gerçek zamanlı çalışabilmesinden dolayı

literatürde birçok çalışmada kullanılmıştır. Kalman filtresi, nesne takibinde hedef nesnenin doğrusal hareket ettiğini ve sistemde Gauss paraziti bulunduğunu kabul eder. Nesne takibi amacıyla kameralardan elde edilen ölçümler çoğu zaman parazit içermektedir. Bunun yanında nesne hareketleri de rastgele düzensizlikler içerebilir. Kalman filtresinin de dahil olduğu istatistiksel eşleme metotları, nesne durum tahmini boyunca ölçüm ve model belirsizliklerini de hesaba katarak bu gibi nesne takibi problemlerine çözüm üretir. Kalman Filtresi de diğer istatistiksel eşleme metotları gibi pozisyon, hız ve ivme gibi nesne özelliklerini belirleyebilmek için durum alan yaklaşımını kullanır [46]. Kalman filtresi, Gauss dağılımı olan durumlarda doğrusal bir sistemin durumunu tahmin etmek için kullanılır. Kalman filtrelemesi üç adımdan oluşur: önerme, doğrulama ve asimilasyon. Bu algorithmada amaç,  $t-1$  anındaki sistem model önermesi ile  $t$  anındaki ölçümler kullanılarak  $t$  anındaki sistemin durumunu bulmaktır. Önerme aşamasında,  $t-1$  anındaki sistem modeline ve nesne durumuna dayanılarak  $t$  anındaki nesnenin durumu önerilir. Ölçüm aşamasında,  $t$  anındaki görüntü üzerindeki özellikler çıkarılır. Bu özellikler, nesne durumunun doğrusal dönüşümü olarak kabul edilir. Asimilasyon aşamasında ise, önerilmiş durumlar ile ölçülmüş durumlar kombine edilerek nesnenin yeni durumu çıkarılır. Kalman filtresinin, tek şekilli ve Gauss olasılık dağılımının olduğu durumlarda çalışması bu algoritmanın önemli bir sınırlamasıdır. Çünkü genellikle nesnelere Gauss dağılımlı değildir. Kalman filtresinin başka bir dezavantajı da karmaşık arka planlara karşı hassas olmasıdır. Kalman Filtresi, bir sonraki görüntü üzerinde nesnenin yeri hakkında tek bir tahmin yapar. Bu sebepten ötürü Kalman Filtresi ile yalnızca tek bir nesnenin takibi yapılabilir. Bilindiği gibi nesneyi temsil etmesi için seçilen model, nesnenin hareketini kısıtlar. Örneğin nesne yalnızca bir nokta ile temsil ediliyorsa, noktanın dolayısıyla nesnenin yapabileceği hareket yalnızca ötelemeli bir harekettir. Elips gibi bir geometrik şekil gösterimi kullanılması durumunda ise ilginç ve izdüşümsel dönüşümler gibi parametrik hareket modelleri uygundur [47]. Kalman Filtresi yönteminde hedef nesne, nokta ile temsil edilmektedir. Eğer hedef nesne geometrik bir şekil ile temsil edilebilseydi takip işlemi sırasında nesnenin boyutlarındaki değişiklik ölçülebilirdi. Bu nedenle Kalman Filtresi yöntemi ile yapılan nesne takibinde hedef nesnenin boyutları, kapladığı alan ölçülemez yalnızca görüntü üzerindeki yeri bulunabilir.

### 2.8.3. CONDENSATION Yöntemi ile Nesne Takibi

CONDENSATION (Kosullu Yoğunluk Yayılımı - Conditional Density Propagation) algoritmasının temel kullanım alanı karmaşık bir ortamda hareket eden nesnelerin dış çizgilerinin bulunması ve takip edilmesidir. Görüntü üzerindeki piksellerin hangilerinin nesnenin kenarlarına ait olduğunu bulmak önemli bir problemdir. CONDENSATION bu problemi çözmeye çalışan olasılıksal bir algoritmadır [47]. Bu algoritma, parçacık filtresinin bir varyasyonu olup doğrusal olmayan, Gauss olmayan takip problemlerinin de üstesinden gelmektedir. Algoritmanın farklı yönlerinden biri görüntüdeki her bir piksel üzerinde işlem yapmıyor olmasıdır. Bunun yerine, işlenecek pikseller rastgele seçilmekte ve yalnızca bu piksellerin bir alt kümesi işleme sokulmaktadır. CONDENSATION her bir iterasyonunda üç tane adım içeren iteratif bir algoritmadır. Bu adımlar sırayla önerme, güncelleme ve yeniden örneklemedir. CONDENSATION, sıralı MonteCarlo önem örneklemesini kullanan bir bayes özyinelemeli tahmincidir [27]. Kısaca anlatılacak olursa, sıralı gözlemler kullanılarak o anki çok boyutlu durum tahmin edilir. Gözlemler, durum ile olasılıksal gözlem modeli üzerinden ilişkilidir. Durum dinamik olabilir, bu dinamizm hareket modeli içerisinde yakalanır. Algoritmanın sıralı doğası iteratif eşitlikler sağlar, örnekleme doğası ise N parçacıktan oluşan kümeyi yönetme yetisi sağlar. Her bir parçacık bir durum tahminini temsil eder ve buna ilişkilendirilmiş bir ağırlığa sahiptir.

Global tahminler parçacık kümesinin tamamına dayanarak yapılabilir. CONDENSATION'ın her iterasyonundaki önerme adımında, yeni bir parçacık elde edilerek ve parçacık kümesi oluşturularak her bir parçacığın hareket modeli örneklenir. Güncelleme adımında her bir parçanın ağırlığı gözlem modeli kullanılarak hesaplanır. O anki gözlemlerle aynı sonucu üreten parçacıkların ağırlıkları arttırılır. Yeniden örnekleme adımında, o anki parçacıkların ağırlıklı dağılımının örneklemeinden yararlanılarak yeni bir parçacık kümesi oluşturulur. Ağırlığın yüksek olması, parçacığın bir sonraki kümede yer alma olasılığının yüksek olması demektir [47]. Avantajlarının yanında, CONDENSATION algoritmasının dezavantajları da bulunmaktadır. Olasılık tavan değerine çıktığında veya yeni ölçümler öncekilerin kuyruğunda yer aldığı anda örneğin başarısız olduğu görülebilir. Bu dezavantajların üstesinden gelmek amacıyla birçok çalışma yapılmaktadır. Ayrıca, önceki ve sonraki dağılımların örneklerinin sayısını azaltmak için de birçok çaba harcanmaktadır [48].



#### **2.8.4. Nitelik ve Modele Dayalı Nesne Takibi**

Literatürde iki ana yaklaşım vardır. İlk yaklaşım görüntü dizisindeki optik akış alanlarını ya da yoğun hareket alanlarını belirleyerek görüntüdeki nesnelere için yapıyı anlamak amacıyla bu akış veya hareket alanlarını ya da her ikisini de analiz eder. İkinci yaklaşım bir görüntüdeki bir nesneye ait kesikli niteliklerin sonraki görüntüdeki niteliklere benzerliğe dayanır [49]. Bu yaklaşımda genellikle görüntüdeki niteliklerin konumları şablon eşleme ya da diğer arama yöntemleri kullanılarak aranır. Uygun eşleme bulunduğu zaman nesne hareketi bu benzerlikten bulunur.

##### **2.8.4.1. Niteliğe Dayalı Takip**

Nitelik takibindeki asıl amaç ardışık görüntülerdeki niteliklerin görüntüdeki konumlarını tahmin etmeyi sağlamaktır. En çok kullanılan ortak nitelikler kenarlar, köşeler ve fazla karşılaştırılmış görsel bölgelerdir [49]. Buna ek olarak aynı özelliklere sahip piksel grupları da nitelik olarak kullanılabilir. Son yıllarda nitelik takibi için değişmezlerin nitelik olarak kullanımını mümkün kılmak üzere yeni değişmezlerin hesaplanmasına başlanmıştır. Köşeler ve hatlar gibi standart niteliklerin yerine projektif değişmezler kullanılmaktadır [50]. Projektif değişmezler kamera pozisyonundan bağımsız olan görüntü nitelikleridir.

##### **2.8.4.2. Modele Dayalı Takip**

Modele dayalı takip yaklaşımlarının çoğunda ön bir bilgi olarak modelin bilindiği varsayılır. Amaç model ve görüntü nitelikleri arasındaki maliyet fonksiyonunu minimize eden uygun bir eşleştirme yapmaktır. Eklemlenmiş nesnelerin modele dayalı takibine ait yeni bir çalışma sunulmuştur [50]. Nesnenin önceki konfigürasyonunun bilgisi ve görüntüden alınan gözlemler Genişletilmiş Kalman Filtresi ile nesnenin yeni konfigürasyonunu tahmin etmek için birleştirilmiştir. Diğer yeni çalışma ise çok yüzlü nesnelerin 2-3 boyutlu modelleri kullanmıştır [51]. Önemli bir problem olan kısmen engelli görülebilen nesnelerin model tabanlı takibine ilişkin bir çalışmada sunulmuştur [52]. Bu yaklaşım, görüntüde nesne takibi yerine takip açısından nesne tanımayaya dayanır.

#### **2.8.5. İnsan Hareketlerinin Analizi Açısından Nesne Takibi**

İnsan hareketlerinin bilgisayarla analizi geniş olarak incelenmiş bir konudur [53]. Vücut parçalarının takibine dayalı insan hareketi analizinde modele dayalı ve görünüme

dayalı olmak üzere iki çeşit yaklaşım vardır. Modele dayalı yaklaşımda küçük figürler, ya da iki boyutlu kenarlar ya da hacimsel şekiller olarak temsil edilen insan vücut parçalarının modelleri kullanılır. Görünüme dayalı yaklaşımda insan hareketi alt seviyede ve iki boyutlu nitelikler kullanılarak tanımlanır. Nitelik çıkarımından sonra görüntü kareleri arasındaki nitelik benzerlikleri, pozisyon, hız, doku ve renge dayalı, örneğin Kalman Filtresi gibi istatistiksel tahminciler kullanılarak elde edilir.

Kapalı devre izleme uygulamaları gibi örneklerde ise vücudun tamamının takip edilmesi bağımsız vücut parçalarının takip edilmesine göre daha verimli bir şekilde yapılmıştır. Burada takip edilecek nitelikler, noktalar, iki boyutlu öbekler, piksel grupları ya da hacimler olarak temsil edilmişlerdir. Her piksel grubu, grup üzerindeki piksellerin konum ve renklerinin gaussian dağılımı ile temsil edilmişlerdir. Piksel grupları kişinin baş, el ve ayaklarına karşılık gelir. Bunların takibi ise sonraki görüntüde kişinin görünüşünü tahmin etmeyi piksel grubuna ya da arka plan modellerine ait pikseller arasındaki benzerliği belirlemeyi, bu modellerden birisi ile ilişkilendirmeyi ve istatistiksel modelleri güncellemeyi gerektirir. Ancak bu çalışma bir kişiyi takip etmekle sınırlı kalmıştır. Başka bir çalışmada W4 ile isimlendirilen bir sistem insanların vücut parçalarını sürekli takip etmek üzere geliştirilmiştir[55]. Sistem istatistiksel olarak arka planı modeller ve ön plandaki nesnelere algılayarak insan takibi için şekil ile periyodik hareket bilgisini kullanır. Bu sistemde vücut parçalarını vücut hizası dışında iken takip edebilmekle sınırlıdır.

Ortalama kayma algoritması örnek dağılım noktası için en yakın modu aramayı sağlayan parametrik olmayan istatistiksel bir yöntemdir [54-56]. Algoritma son zamanlarda, görünüme dayalı, piksel gruplarının takibi için adapte edilmiştir. Bu piksel gruplarının takibi senaryosunda, örnek noktalar, o noktada ki örnek ağırlıkları olacak olan piksel değerleri, görüntü koordinatları boyunca düzgün olarak dağıtılmışlardır. Bu ağırlıklar nesneye ait olanların yüksek arka plana ait olanların ise düşük olacak şekilde histograma dayalı tekniklerle seçilirler.

Ortalama kayma hızlı bir takip yöntemi olduğundan hem tekli hem de çoklu nesne takibinde kullanılmaktadır. Bradski [55-56] geliştirdiği insan bilgisayar etkileşimi uygulamasında ortalama kayma yöntemini adaptif pencere kernel boyutları ile kullanmış ve yöntemi Camshift (Continuously Adaptive Mean Sift) olarak adlandırmıştır. Yöntemde kullanıcı yüzünden takip edilmesi ile sanal gerçeklik ve video oyunu kumandaları gerçekleşmiştir. Adaptif pencere büyüklüğünün nasıl gerçekleştirileceğine dair örnekler sunulmuştur. Ancak takip esnasında boyutu değişen nesnelere için pencere boyutlarının

nasıl adapte edileceğine dair en ciddi çalışma Collins [56] tarafından yapılmıştır. Collins Adaptif pencere boyutu için Lindeberg teorisini [57] pencere boyutu fonksiyonuna uygulamış aynı zamanda standart ortalama kayma takip algoritmasında negatif piksel ağırlıklarını da kullanabilir hale gelmiştir.

Bilindiği gibi video görüntüleri çerçevelerden oluşmaktadır. Ardışık çerçeveler arası ise global olarak bir yer değişim söz konusudur. Bu yer değişim hareket vektörünü oluşturmaktadır. Hareket vektörü başka bir ifadeyle T anındaki çerçevenin (T+1) anında genel olarak ne kadar yer değiştirdiğidir.

### 3. PARALEL PORT İLE İLETİŞİM

Kontrol tekniğinin son yıllarda büyük bir gelişme göstermesiyle bilgisayar teknolojisi sayesinde elde edilen elektriksel sinyaller ile birçok mekanizmayı denetlemek mümkün hale gelmiştir. Bu sayede bilgisayar arabirimi olan paralel portun kullanım alanları da artmış ve tanımlanan sistemler için farklı kontrol ve yazılım teknikleri geliştirilmiştir. Bu bölümde bilgisayar ile kontrol edilen mekanizma arasında iletişimi sağlayan paralel port arabirimi hakkında detaylı bilgi verilecektir.

Paralel port iletişimde ise port konektörünün bölümleri ile bölümlerin işlevleri üzerinde durulacak ve kontrol edilmek istenen sistemde yazılımların nasıl yapıldığı ve işlendiği konularına da değinilecektir.

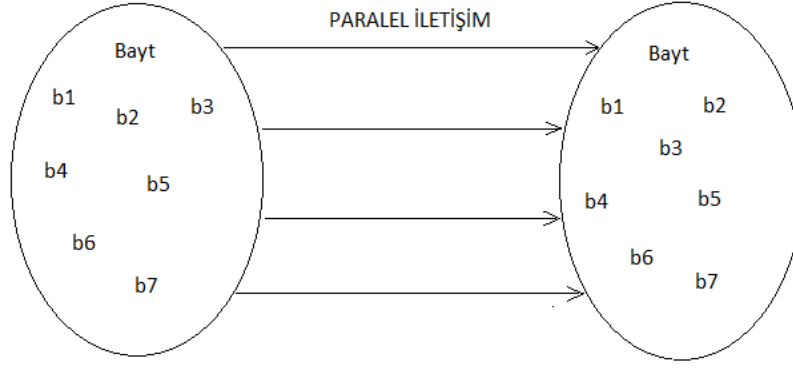
İletişim sisteminde MATLAB programlama dili kullanılmış olup bu konuyla ilgili ayrıca bilgi verilecektir. Aşağıda Şekil 3.1'de paralel port gösterilmektedir.



Şekil 3.1. Paralel Portlarda Kullanılan DB-25 Konektörü

#### 3.1. Paralel İletişim Genel

Paralelport bilgisayarın en kolay programlanabilen portudur [58]. 25 çıkış ucuna (pin) sahip olup 5 volt ile çalışmaktadır. Paralel port üzerinden aynı anda 8 bitlik veri aktarımı sağlanmaktadır. Porttan veri çıkışı için 2-9 pinleri arasındaki pinler kullanılır ve bu pinler “DATA Pinleri” veya “DATA Portu” adını almaktadır. Aşağıda verilen şekilde paralel portun temel çalışma şeması tanımlanmıştır.

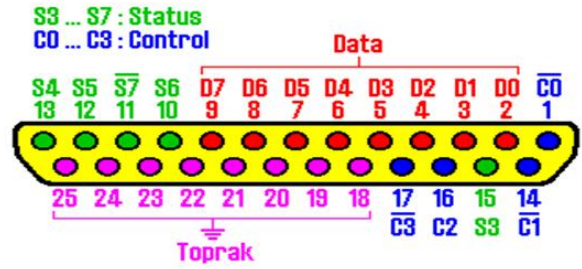


Şekil 3.2. Paralel portun basit olarak karakterize edilişi

Bilgisayar mikroişlemci teknolojisi üzerine kurulu olduğu için verileri “1” veya “0” lojik değerlerine göre işler. Normal koşullarda yani paralel port çalışmadığı sürece DATA pinleri üzerinde “0” lojik değeri vardır. Fakat herhangi bir program yardımı veya bilgisayarın normal çalışma düzeninden faydalanılarak bu değerler “1” lojik değerine dönüştürülebilir. Bir DATA pininin lojik “1” değerine sahip olması demek o pinin 5 voltluk gerilim almış olması anlamına gelmektedir. Bu gerilim değeri belki çok büyük veya güçlü devreleri çalıştırmak için yeterli değildir fakat en azından transistör, tristor gibi elemanlar tetikleyecek güce sahiptir. Aşağıda verilen Tablo 3.1'de ise bilgisayarın paralel portunun (yazıcı portu-LPT1) pin çıkışlar gösterilmiştir. Aşağıda Şekil 3.3'de paralel port soketi ve Şekil 3.4'te paralel portun uçları gösterilmektedir[58].



Şekil 3.3. Paralel port soketi görünüşü



Şekil 3.4. Paralel port uçlarının görünüşü

Paralel port üzerinde yukarıda Şekil 3.4'te görüldüğü gibi DATA portundan başka, STATUS ve CONTROL portları da bulunmaktadır. Hangi pinlerin hangi porta ait olduğu yine aşağıda verilen Tablo 3.1'de ifade edilmektedir. Tablo incelendiğinde seri porttakinden farklı olarak 8 tane toprak hattının var olduğu göze çarpmaktadır. Birkaç portun veya tüm portların kullanıldığı düşünüldüğünde her devre akımının aynı yolu takip

ederek geri dönmesi istenmediğinden ve çok sayıdaki toprak hattının toprak dönüşlerinin genel empedansını düşürmesi söz konusu olduğundan hassas uygulamalar için 8 ayrı pinle topraklamayı tercih etme durumu olacaktır[58].

### 3.2. Paralel Port Adresleri

Paralel Port genel olarak kullanılan 3 adet temel adrese sahiptir. Bu adresler aşağıdaki Tablo 3.1'de listelenmiştir. 3BCh temel adresi; ilk olarak video kartlar üzerinde paralel portlar için kullanılmıştır. Bu adres; daha sonra paralel port video kartlardan kaldırıldıktan sonra, belirli bir süre kullanılmamıştır. Şimdi ise; konfigürasyonlar. BIOS (Temel Giriş/Çıkış Sistemi) ile değiştirilebilecek şekilde ana kartlar üzerine entegre biçiminde konmuş paralel portlara seçenek olarak yeniden ortaya çıkmıştır. LPT1 normal olarak 378h adresini, LPT2 ise 278h adresini kullanmaktadır. 378h & 278h adresleri daima genel olarak paralel portlar için kullanılmıştır. "h" durumu adresin hexadecimal düzende olduğunu göstermektedir. Bu adresler makineden makineye de değişebilir özellikler taşımaktadır.

Tablo 3.1. BIOS veri alanındaki LPT Adresleri

Adres	Açıklama
3BCh - 3BFh	Video kartları üzerine yerleştirilmiş paralel portlar için kullanılmıştır. ECP adreslerini desteklemez.
378h - 37Fh	LPT 1 için genel adres
278h - 27Fh	LPT 2 için genel adres

Bilgisayar ilk açıldığında, BIOS bilgisayarın port numaralarını belirleyip onlara LPT1, LPT2 ve LPT3 aygıt etiketlerini vermektedir. BIOS ilk olarak 3BCh adresine bakmakta, burada bir paralel port bulundu ise buna LPT1 ismini vermektedir. Daha sonra 378h yerleşimini araştırmakta ve burada paralel kart bulundu ise ona da sıradaki port adres numarasını vermektedir.

Yukarıda verilen adres numaralarına göre;

- DATA portu 0378,
- STATUS portu 0378+1, yani 0379,
- CONTROL portu ise 0378+2, yani 037A olmaktadır [58].

### **3.3. DATA Portu ile Programlama**

DATA portunun adresi bilindiğine göre adres pinlerinin yardımı ile veri akışı sağlanabilmektedir. Bilgisayarın paralelportu ve projemizde kullandığımız MATLAB yardımı ile adım motoru kontrol edebilecek programlamalar yapılabilmektedir. Paralelport bilgileri gönderilirken dikkat edilmesi gereken; sistem çıkışlarının 10'luk düzende yazılıp yazılım dilinde bunların 2'lik sayı düzeyinde bellekten okunacağıının önceden bilinmesidir. Bu yolla gönderilen bilgiler ışığında paralelport çıkışlarına bağlı sürücü devresi anahtarlama yaptırılarak adım motoru kontrol edilebilir[58].

#### 4. YAPILAN ÇALIŞMALAR VE İRDELEME

Günümüzde İnsanların güvenliğe olan gereksinimleri hızla her geçen gün artmaktadır. Son yıllarda ise şüphe çeken kişi veya nesnelerin doğrudan görüntü işleme teknikleri kullanılarak belirlenmesine yönelik çalışmalar yapılmaktadır. Bu tez kapsamında gerçekleştirilen sistemde, hareketli bir platform üzerine yerleştirilmiş hareketli kamera alınan görüntüleri gerçek zamanlı olarak değerlendirerek, hareketli bir nesneyi değişken ortamlarda takip etmektedir.

Sistemde ilk olarak renkli görüntü çekimi yapan bir kameradan görüntü alımı yapılmaktadır. Daha sonra renk derinliği indirgenen görüntü üzerinde hedef nesne aranmaktadır. Hedef nesne sistemin görüş açısına girene kadar sistem hedef nesneyi beklemekte ve aynı zamanda ortamı gözlemektedir. Hedef nesne sistemin görüş açısına girdiği zaman nesne bulma algoritması hedef nesnenin yerini belirlemekte ve bu yer bilgisini nesne takibi aşamasına aktarmaktadır. Nesne takibi aşaması bu yer bilgisi kullanarak sıradaki görüntüde hedef nesnenin yeni yerini bulmakta ve bu bilgiyi nesne takibi aşamasının bir sonraki adımına aktarmaktadır. Diğer adımlar da ardışık görüntüler üzerinde çalışarak hedef nesnenin bir önceki görüntüde bulunduğu bölgenin civarında hedef nesneyi aramakta ve nesnenin bulunduğu yeri bir sonraki adıma aktarmaktadır.

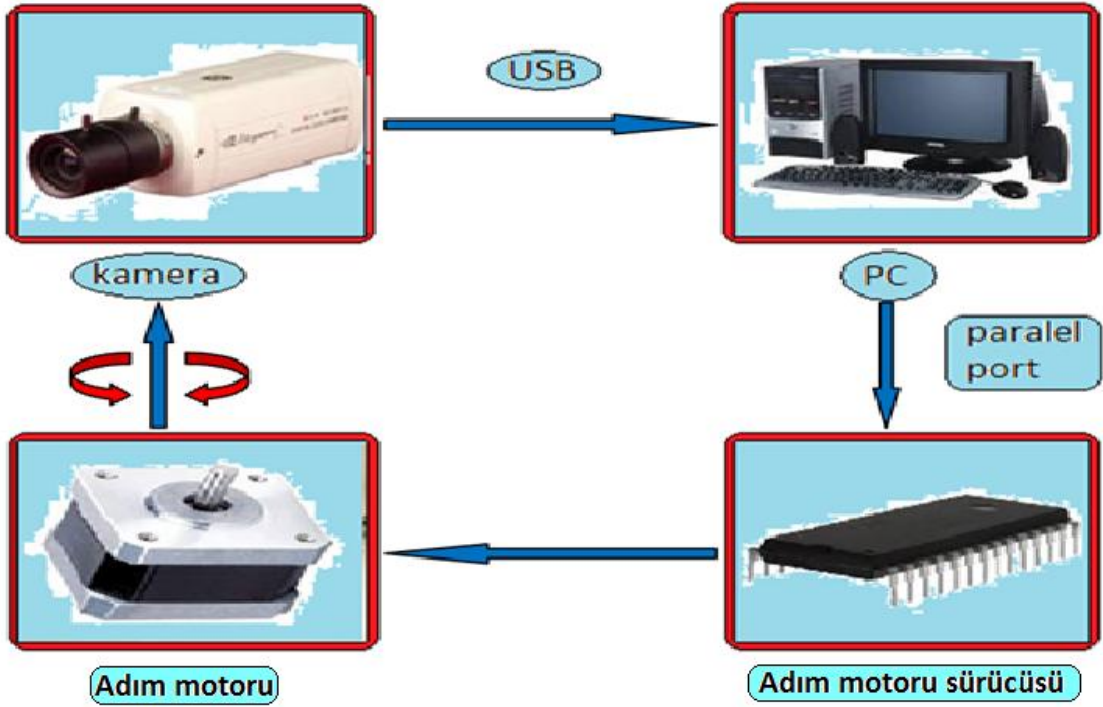
Bu amaçla kişisel bilgisayarın paralel portuna bağlı adım motoru düzeneğinin üzerinde bulunan kamera, sürekli olarak hareket eden cismi takip edecek şekilde yönlendirilmektedir. Uygulama yazılımı MABLAB 7.9 kullanılarak geliştirilmiştir.

Tüm kontrolleri bilgisayardan yapabilmek için MATLAB 7.9 yazılımı kullanılarak aşağıda Şekil 4.1'de görüldüğü gibi bir kontrol programı yazılmıştır. Program görüntü işleme ve kontrol amaçlı olarak kullanılmıştır. Görüntü ile ilgili olarak şu işlemler yapılabilmektedir; Görüntü yakalama, görüntüyü okuma, görüntüyü gri tona çevirme ve bir eşik değeri belirleme için image processing Toolbox özel kütüphanesi ve bu kütüphaneye ait komutlar kullanılmıştır. Kontrol amaçlı olarak program şu işlemleri gerçekleştirebilmektedir. Görüntü işlendikten sonra hareket yönünün kestirimi yapılarak adım motorunun konum kontrolü bilgisayarın paralelportundan gelen işaretler ile gerçekleştirilmiştir.



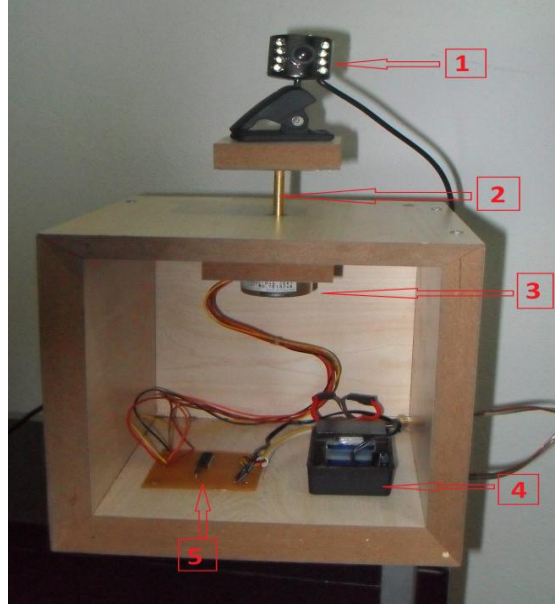
#### 4.1. Sistemin Yapısı

Bu çalışmada kamera tarafından alınan görüntü MATLAB programı ile işlenerek hedef çerçeve ile bir sonra ki hedef çerçeve arasında hareket olup olmadığını ve şayet hareket varsa X eksenini veya Y eksenini üzerinde mi olduğu kestirimi yapılmaktadır. Tespit edilen hareketin yönüne bağlı olarak adım motorumuzun sürücüsüne işaret gönderilerek gerçek zamanlı nesne takip uygulaması gerçekleştirilmiştir. Takip etme işlemi temel olarak iki ayrı blok kullanılarak gerçekleştirilmiştir. Birinci blok; kameradan alınan görüntünün işlenerek bir sonuca ulaşılmasını sağlamakta, ikinci blok ise bilgisayardan gelen veriye göre üzerine kamera yerleştirilen adım motoru adım motoru sürücüsü ile tahrik etmektedir. Geliştirilen sistemin yapısı aşağıda Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Hareketli nesne takip sistemi

Tasarlanan bu kontrol sistemlerinin özelliklerinden bahsetmeden önce sistemin genel görünümünü hakkında bilgi vermek yararlı olacaktır. Aşağıda Şekil 4.2'de sistemin üstten görünüşü verilmiştir:



Şekil 4.2. Hareketli kamera sistemi

Yukarıda görünüşü verilen sistemin çalışma şekli ve özelliklerinden bahsetmeden devre üzerinde bulunan ve numaralandırılmış ana parçaları şu şekilde tanımlayabiliriz:

### 1. Kamera

Tarantula A Type High Definition HD PC Kamera, 1280 x 720 px Video çekim çözünürlüğü ile High Definition(yüksek çözünürlük) (HD) video kayıt, driver (sürücü) yüklemeye ihtiyaç duymadan direkt kullanabilme , 1280 x 800 px Fotoğraf çekme çözünürlüğü, yazılım desteği ile 10.0 Megapiksel (3648 x 2736 px) resim çekebilme çözünürlüğe ulaşma, USB 2.0 ile daha hızlı görüntü akışı, 5 kat cam lens ile daha net görüntü, 8 adet beyaz LED aydınlatma ışıkları, Işık seviyesini kablo üzerindeki kontrolden el ile ayarlayabilme, dahili mikrofon, 10X Dijital zoom, notebook ile kullanabilme, 1280 x 720 px video çekim hızı 25-30 fps, efektli resim çekebilme özelliği, odaklanma mesafesi: 5.0cm – sonsuz, otomatik beyazlık kontrolü/ arka plan ışığı dengeleme, resim keskinlik kontrolü, windows™ 98 SE, ME, 2000, XP ve Vista uyumludur.

2. Aktarma düzeneği; Motor hareketini panele aktaran düzenektir.

### 3. Adım Motoru

Astrosyn tarafından üretilmiş, +12 V ile çalışan, 1,8°'lik adım derecesine sahip unipolar hibrit adım motorudur.

Kurulan sistemde pozisyon kontrolünün hassas olması için adım motoru kullanılmıştır. Kullanılan motor 6 kablolu 1 fazlı adım motorudur. Her fazın endüktansı 3.5 mH ve direnci  $8.5\Omega$ ' dur. Çalışma akımı 2.7mA ve çalışma gerilimi +5V'dur. Siyah-turuncu-kahverengi-sarı kabloları sırası ile enerji verilirse motor saat ibresi yönünde döner. Sarı-kahverengi-turuncu-siyah kabloları sırası ile enerji verilirse motor saat ibresinin tersi yönünde döner. Kırmızı kablo ortak uçtur.

#### 4. Güç kaynağı

Köprü diyot, kondansatör ve 7812 regülâtöründen meydana gelen +12 V üreten güç kaynağıdır.

Trafo: 220/12 V transformatör.

#### 5. Sürücü Entegre

ULN2003A isimli tetikleme işlemini yapan entegre arabirimidir.

Paralel Port Arabirimi: Lojik bilgilerin devreye aktarıldığı ve programlama ile konfigüre edilen 25 pinli konektör birimidir.

Sisteme gelen işaretlerin değerlendirilmesi için bilgiler, bilgisayar ortamına aktarılmıştır. Adım motor sürücü devresi de bilgisayardan gönderilen işaretler ile sürülmektedir. Bilgisayar ile sistem arasındaki bilgi alış verişi paralelport vasıtasıyla yapılmaktadır. Paralelportun kontrol yazmacı yapısı gereği sadece çıkış vermektedir. Bu nedenle kontrol portu adım motorunu sürmek için kullanılmıştır. Kontrol portunun data0(D0), data1(D1),data2(D2) ve data3(D3) bacakları kullanılmıştır.

### 4.2. Tasarım Aşaması

Projenin tasarım aşaması 3 ana bölümden oluşmaktadır. Bunlar;

- **Mekanik düzeneğin oluşturulması:** Yatay hareketi sağlayan adım motoru, motor sürücü devresi, güç kaynağı ve aktarma elemanları sistemin mekanik ve elektriksel bölümünü oluşturmaktadır.
- **Kontrol sistemi:** Bu bölümde mekanik sistemin hareketinin kontrolü amaçlanmıştır. Webcamden alınan görüntüde ki hareketli cismin konum bilgilerinin değerlendirilmiştir. Bu değerlendirmeye göre adım motorun hareketi saat ibresi veya saat ibresinin tersi yönünde ki harekettir.
- **Algoritma ve yazılım:** Mekanik sistemin tüm hareketlerinin kontrolü için mikro

kontrolör / mikro denetleyici kullanılmıştır. Mikrokontrolör en kolay şekilde anlatılacak olursa bir bilgisayardır. Sistem hareketlerini bir algoritma doğrultusunda yapmaktadır. Tasarım sırasında bu algoritma esnek tutulmuş ve nesnenin en az hareketle bulunması amaçlanmıştır.

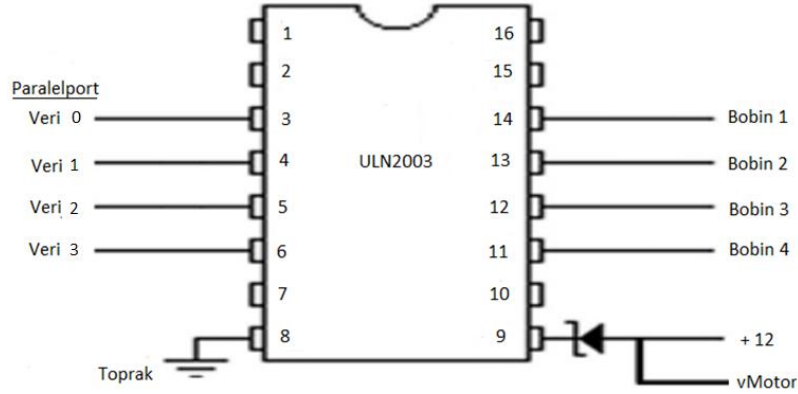
#### **4.2.1. Mekanik düzenek**

Mekanik sistem, sağa-sola hareketi sağlayacak adım motoru ve hareketi panele aktaracak olan milden oluşmaktadır. Hazırlanan yazılım sayesinde aşağıda anlatılan voltaj bilgilerine göre adım motor uçlarına darbe uygulanmıştır. Adım motorunu döndürmek için gerekli olan akımı verebilmek için sürücü devresi tasarlanmıştır.

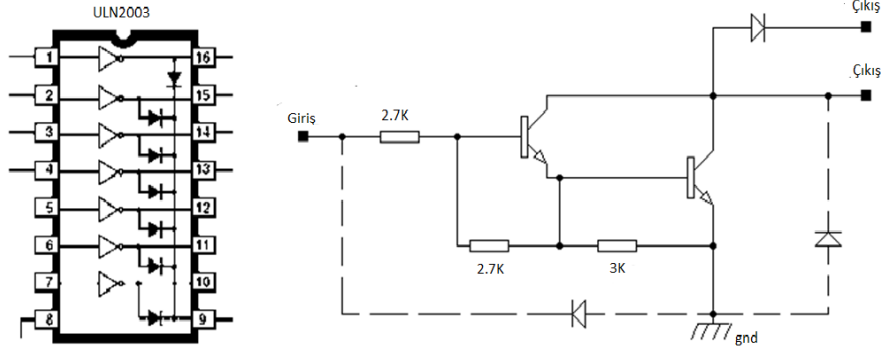
#### **4.2.2. Sürücü Devresi**

Sürücü devresi olarak kullanılan ULN2003 içerisinde Şekil 4.4'te görüldüğü gibi 7 adet NPN transistör ve dahili diyot barındırmaktadır. Kullanımı ise oldukça kolaydır. Devre şemasından da anlaşılacağı gibi 9 numaralı bacağına +12 Volt ve 8 numaralı bacağına da toprak uyguluyoruz. Daha sonra Şekil 4.3'te görüldüğü gibi 3 ve 6 numaralı bacaklara paralel portun DATA pinlerinden gelen +5 Voltluk değerleri uygulayacağız. Bu sayede örneğin 3 numaralı bacağına +5 Volt işaret uyguladığımızda 14 numaralı bacak toprak olacaktır. Aynı şekilde sırayla 4 için 13, 5 için 12, 6 için ise 11 numaralı bacaklar toprak olacaktır. Bu sargıların uçlarını nasıl tespit edildiği ayrıntılı olarak yukarıda anlatılmıştır. Yukarıda anlatıldığı gibi motora adım attırmak için yapmamız gereken, ortak uç kablosuna +12 Volt verirken, diğer sargılara bağlı kabloları belli bir sıra ile toprak göndermektir.

### 4.2.3. Devrenin Bağlantı Şeması

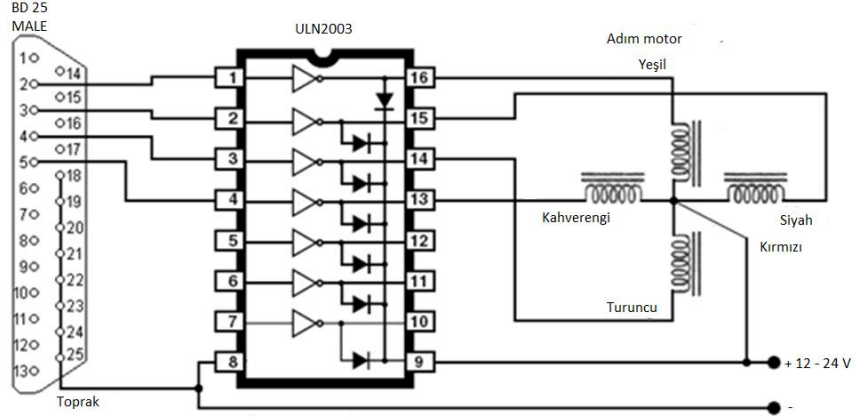


Şekil 4.3. ULN 2003 entegresi bağlantı şekli



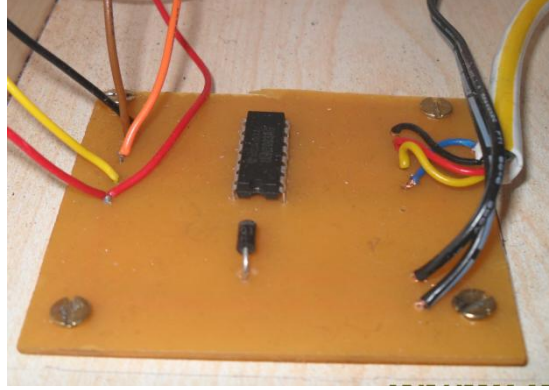
Şekil 4.4. ULN2003 entegresinin iç yapısında darlington bağlı transistörün yapısı

Şekil 4.4'te görüldüğü gibi ULN2003 entegresinin iç yapısında darlington bağlı transistörler bulunmaktadır. Transistörlerin ilettime girebilmesi için entegrenin giriş uçlarına +5V'luk gerilim uygulamak yeterli. Giriş uçlarına seri bağlı dâhili dirençler bulunduğu için giriş akımını sınırlamak üzere harici dirençler kullanmaya gerek yok. Bundan dolayı, Şekil 4.5'te görüldüğü gibi ULN2003 entegresi mikro denetleyici çıkışına doğrudan bağlanabiliyor. Her bir kanalın çıkış akımı 500mA seviyesindedir.



Şekil 4.5. ULN 2003'ün Adım motoru bağlantısı

### 4.3. Devrenin Kurulumu ve çalışması



Şekil 4.6. Adım motorun sürücü devrenin bağlantı şekli

Projede kullanılan sürücü devresi Şekil 4.6'da gösterilmiştir. Bu devre direk paralel port data pinleri ile beyzinden sürülen transistörlerin anahtarlama prensibine göre çalışmaktadır. Motor her adım da bir pin tarafından kontrol edilmektedir.

Projede paralel port adresi olarak 378 hex adresi kullanılmıştır. Paralel porttan x eksenini adım motorunu sürmek için bu porta ait data pin çıkış uçları kullanılmaktadır. Bu pinler ilk pin 1 nosu ile isimlendirilmek üzere 2 ile 9 nolu pinler dahil ve bu pinler arasında kalan pinlerdir. Bu pinlerden sürücü devresi vasıtasıyla ilgili eksene ait adım motorunun ilgili adımına ait adım motoru uçlarına gerilim uygulanmakta ve adım motoru bir adım atmaktadır. Adım motorlar yapıları gereği, çok sayıdaki uçlarına düzenli olarak gelen darbelerle hareket ederler. Her bir adım için belirli uçlara gerilim verilir. Adım motor bir adım hareket eder ve öylece kalakalır. Sonra yeni darbeler farklı bir sırayla tekrar verilir. Bir adım daha gider. Daha sonra yeni bir darbe dizisi gelir ve bir adım daha gider. Bu

darbe dizilerini belirleyen tek şey, motorun içerisindeki, sargıların şekli ve sıralamasıdır.

#### **4.4. MATLAB Programı**

##### **4.4.1. Matlab Yazılımı**

Projenin ana yazılım kısmı, MATLAB ortamında kod ile uygulama gerçekleştirme (yani, m dosyası denilen metin dosyaları oluşturularak) kullanılarak yapılmıştır. Bu programı oluşturan öğeler şöyledir:

- a) Görüntü yakalama fonksiyonları
- b) Görüntü işleme fonksiyonları
- c) Adım motoru kontrol fonksiyonları

Sistemi kontrol etmek için bilgisayar kullanılmıştır. Gerekli işlemleri bilgisayara yaptırmak için MATLAB programı kullanılmıştır. MATLAB programı programlamaya elverişli olması ve yaygın bir kullanıma sahip olduğu için tercih edilmiştir. Yazılan programın tamamı ekler bölümünde verilmiştir.

Birçok resim işleme yönteminin ortak sorunu resimlerdeki kontrast ve parlaklıktır. Resimdeki her bir ayrıntı bilgisayar için çok önemli ve bu ayrıntılardaki ufak değişiklikler bilgisayar tarafından hemen algılanmaktadır. Hareket algılama ile başlayalım. Akan bir görüntü üzerinde bir cismin hareket ediyor olarak kabul edilebilmesi için görüntü üzerinde belli bir bölgede piksel değerlerinin sürekli değişiyor olması gerekir. Yukarıda bahsedilen parlaklık sorunu da işte burada ortaya çıkmaktadır. Webcam ile çalışılacağı için özellikle otomatik olarak ışık dengesini ayarlayan webcam bu konuda sorun çıkaracaktır. Bu sorunu çözmek için piksellerde hareket belirtme ihtimali düşük olan değişiklikler göz ardı edilmiştir. Hareketli görüntüyü oluşturan resimler ikişer ikişer incelenmiştir. 1. resmin piksel değerleri ile 2. resmin piksel değerleri karşılaştırılıp, aralarında belirlenen eşik değerinin üstünde fark bulunan pikselleri görüntü üzerinde işaretlenmiştir. İkişer ikişer tüm resimlere aynı işlemi uygulandığında aslında hareket eden pikseller işaretlenmiş olacaktır.

Nesne takibi yukarıdaki yönteme dayanıyor. Hareketli görüntü elde edebilmek için bu işlemlerin arka arkaya yapılması gerekir. Ancak bu yöntemi, 2.1 Ghz işlemcili bilgisayar kullanılarak yapılmıştır. Bu bilgisayar üzerinde saniyede 2-3 resim işlenmiştir. Hızı arttırmak için tüm pikselleri taramak yerine, her pikselden sonra 4-5 piksel atlanarak işlem yapılmıştır. Her bir resim bittiğinde webcamden yeni bir resim isteyip onu işleyerek hareketli görüntü sağlamaya çalışılmıştır.

MATLAB'da webcam'den görüntü almak için ; vid = videoinput('winvideo');  
arkaplan = getsnapshot(vid);

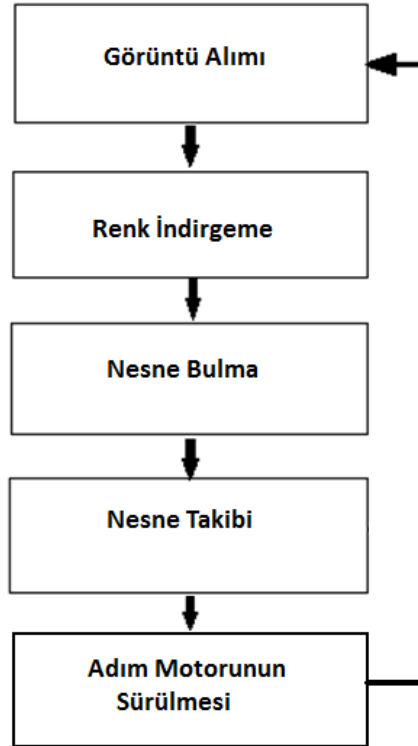
Bilgisayara bağlı bulunan kameradan görüntü alabilmek için yukarıdaki fonksiyon kullanılmıştır.

- Videoinput('winvideo') komutu ile webcam'den kesintisiz görüntü alınmış ve bu görüntüyü 'vid' değişkenine atanmıştır.
- Getsnapshot komutu ile webcamden aldığımız sürekli görüntünün herhangi bir T anında ki fotoğraf karesini yakalamıştır.

#### 4.5. Pozisyon Kontrol Sisteminin Tasarımı

Deneysel çalışmada adım motoru ve görüntü işleme sistemleri kullanılarak bir hedef takip sistemi kurulmuştur. Sistemde; görüntü işleme yazılımından gelen bilgiler bilgisayara aktarılarak değerlendirilmekte, elde edilen sonuca göre adım motoru saat ibresi veya saat ibresinin tersi yönünde dönmektedir. Hedef nereye giderse kamera o yöne döner.

Hareketli bir nesnenin hareketli kamera ile gerçek zamanlı takibini sağlayan sistemin ana adımlarını gösteren blok diyagramı Şekil 4.7'de görüldüğü gibidir.



Şekil 4.7. Sistem blok diyagramı



#### 4.5.1. Görüntü Alımı

Sistem, hareketli nesne takibini hareketli bir düzeneğe yerleştirilmiş kamerayla alınan görüntü akışı üzerinden yapmaktadır. Alınan renkli görüntüler 1280 x 800 'lik genişliktedir ve saniyede alınan görüntü sayısı 25-30'dur.

#### 4.5.2. Renk İndirgeme

Kameradan alınan renkli görüntülerin RGB bileşenlerinin her biri 256 renge sahiptir. Sistemin karmaşık yapısı, gerçek zamanlı çalışıyor olması açısından programın çalışma zamanını azaltmak için 256 renge sahip RGB uzayı Bit-Cutting [55] yöntemi ile 16 renge indirgenmiştir. Bunun için, her bir renk değerinin düşük anlamlı 4 bitine bakılmaksızın yüksek anlamlı 4 bitine bakılarak o rengin yeni değeri belirlenmiştir. Başka bir deyişle üzerinde çalışılan toplam renk sayısı 256x256x256'dan 16x16x16'ya indirilmiştir. Alınan görüntülerde her bir renk bileşeni değeri 8 bit ile ifade edilmektedir. 8 bitle 256 ( $2^8$ ) farklı sayı gösterilebilir. Her rengi 4 bitle ifade edersek renk derinliği de 16'ya ( $2^4$ ) düşmüş olur. 16x16x16 renge indirgenmiş görüntü üzerinde çalışılırken, öncelikle aşağıda görüldüğü gibi her bir rengi (kırmızı, yeşil, mavi) temsil eden 8 bitlik sayı dört kere sağa kaydırılır, yani yüksek anlamlı 4 bit düşük anlamlı 4 bite yerleştirilir ve 8 bitlik sayı 4 bitle ifade edilmiş olur.

1000 0000 → 0000 10001011 0000 → 0000 10111111 0000 → 0000 1111

Bundan sonra kırmızı, yeşil ve mavi renklerin 4 bitleri yan yana birleştirilerek 12 bitlik bir sayı elde edilir ve her bir pikselin renk değeri bu 12 bitlik sayı ile ifade edilir.

Kırmızı (R) → 0000 0100

Yeşil (G) → 0000 1101    Pikselin ifade edildiği renk değeri → 0100 1101 1111

Mavi (B) → 0000 1111

- $i = \text{im}2\text{int}16$ , 16-bit işaretli tamsayı görüntü dönüştürme fonksiyonudur.
- Web kamerasından arka arkaya yakalanan renkli imge çerçeveleri işlem yükü azaltmak için 256 renk derinliğinden 16 renk derinliğine indirgenmiştir.

#### 4.5.3. Nesne Bulma ve Takibi

Nesne takibi bilgisayarlı görüntü işleme uygulamalarındaki önemli konulardan biridir ve çoğu uygulamalarda gereklidir. Nesne takibi, video görüntülerinde edinilen bilgilerin zamana göre değişiminin takip edilmesini amaçlar.

Bu tez çalışmasında nesne takibi Mean-Shift yönteminin geliştirilmiş hali olan Cam-Shift yöntemi ile gerçekleştirilmiştir. Oluşturulan sistemin gerçek zamanlı çalışıyor olması sistemi oluşturan modüllerde kullanılacak yöntemlerin belirlenmesinde önemli kriterdir. Bu nedenle kullanılacak yöntemin başarısının yanında çalışma süresi de çok önemlidir. Hareketli kamera, hedef nesne görüntüde belirene kadar bulunduğu ortamı gözlemektedir. Bu süre içerisinde ortamda hedef nesneyi bulmaya çalışmaktadır. Hedef nesneyi bulmak için video akışı üzerinden alınan 16 renk derinliğine indirgenmiş görüntüler üzerinde çalışılır. Hedef nesneyi temsil etmek için öncelikle bir özellik uzayı seçilir. Bu tez çalışmasında RGB renk özellik uzayı kullanılmıştır.

Hedef model, görüntü üzerinde dikdörtgen bir bölge ve bu bölge içerisindeki renk olasılık yoğunluk fonksiyonu ( $q$ ) ile temsil edilir. Bu dikdörtgen bölgenin merkezi (0, 0) noktası olarak kabul edilir. Bölgedeki diğer piksellerin koordinatları da bu kabule göre normalize edilir. Bölgedeki piksel koordinatları,  $\{x_i\}_{i=1:n}$  ile ifade edilir. Dikdörtgen bölgedeki her bir piksele merkezden uzaklaştıkça azalan bir ağırlık değeri ( $k(x)$ ) atanır. Bu ağırlık ataması olasılık yoğunluk fonksiyonunun sonuçlarını güçlendirir. Çünkü merkezden uzak bölgelerin başka bir nesnenin arkasında kalıp kaybolma olasılığı yüksektir. Hedef modeldeki  $u$  özelliğinin bulunma olasılığı Eşitlik 4.1'de görülmektedir.

$$q_u = C \sum_{i=1}^n k(\|X_i\|^2) \delta[b(X_i) - u] \quad (4.1)$$

□  $\delta$  Kronecker delta fonksiyonudur (Eşitlik 4.2).

$$\delta_{[n]} = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (4.2)$$

$C_{sabit}$  =  $\sum_{i=1}^n q_u = 1$  eşitliğinden yararlanılarak çıkarılmıştır (Eşitlik 4.3).

$$C = \frac{1}{\sum_{i=1}^n k(\|X_i\|^2)} \quad (4.3)$$

Hedef nesnenin renk dağılımına göre görüntüde hedef nesne aranmaktadır. Arama alanı bütün görüntü olarak seçilmektedir. Görüntüde kullanılan renklerin olasılık yoğunluk fonksiyonu ( $p$ ) bulunur (Eşitlik 4.4).

$$P_u(y) = C_h \sum_{i=i}^{n_h} k\left(\left\|\frac{y-X_i}{h}\right\|^2\right) \delta[b(X_i) - u] \quad (4.4)$$

Hedef nesnenin ve görüntünün renk olasılık dağılımdan yararlanılarak ağırlıklar hesaplanır (Eşitlik 4.5).

$$W_i = \sum_{u=1}^m \delta[(X_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}} \quad (4.5)$$

Daha sonra Mean-Shift (Ortalama Kayması) vektöründen yararlanılarak arama

alanının merkezinin yeni koordinatları belirlenir (Eşitlik 4.6).

$$y_1 = \frac{\sum_{i=1}^{n_h} X_i W_i g\left(\left\|\frac{y_0 - X_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} W_i g\left(\left\|\frac{y_0 - X_i}{h}\right\|^2\right)} \quad (4.6)$$

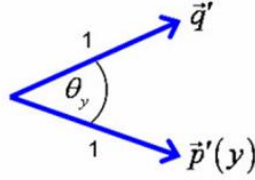
İteratif olarak bu işlemler tekrar edilerek hedef nesnenin olası merkezi bulunmuş olur. Daha sonra bu olası bölgenin olasılık yoğunluk fonksiyonu ile hedef nesnenin olasılık yoğunluk fonksiyonu Bhattacharyya uzaklığı (Eşitlik 4.7) kullanılarak kıyaslanır.

$$\rho[p(y_0), q] = \sum_{u=1}^m \sqrt{p_u(y_0)} q_0 \quad (4.7)$$

Bhattacharyya uzaklığı iki vektör arasındaki benzerliği vermektedir (Şekil 4.8).

$$\vec{q}' = (\sqrt{q_1}, \dots, \sqrt{q_m})$$

$$\vec{p}'(y) = (\sqrt{p_1(y)}, \dots, \sqrt{p_m(y)})$$

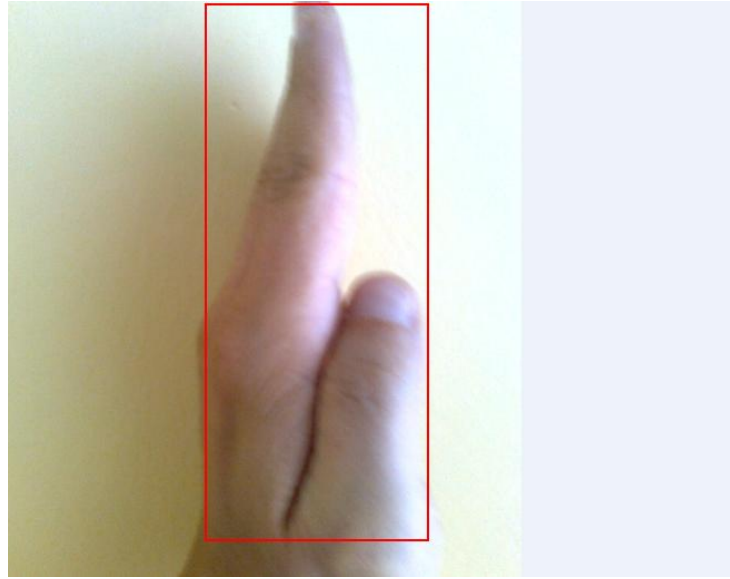


Şekil 4.8. Bhattacharyya uzaklığı

Bhattacharyya uzaklığı 0-1 arası değerler üretmektedir. İki olasılık yoğunluk fonksiyonunun kıyaslanması ile elde edilen sonuç 1'e ne kadar yakınsa benzerlik de o kadar fazladır. Pencere olasılık yoğunluk fonksiyonu ile hedef nesnenin olasılık yoğunluk fonksiyonu kıyaslandığında hedef nesneye en fazla benzeyen pencerenin benzerlik oranına bakılır. Eğer bu oran önceden belirlenmiş olan eşik seviyesi değerinin üzerindeyse hedef nesnenin bu bölgede olduğuna karar verilir. Bu bölgenin merkez koordinatları nesne takibi aşamasına aktarılarak nesne takibine başlanır. Eğer benzerlik derecesi eşik seviyesini geçmediyse hedef nesne bulunana kadar ortam gözlenmeye devam edilir. Mean-Shift ile tespit edilmiş bir hedef nesne Şekil 4.10'da görülmektedir. Hedef nesnenin nesne bulma aşamasına başlamadan önce alınmış olan görüntüsü de Şekil 4.9'da görülmektedir.



Şekil 4.9 Hedef nesnenin görüntüsü



Şekil 4.10 Elin Mean-Shift yöntemi ile bulunması

#### 4.5.3.1. CamShift Yöntemi ile Nesne Bulma

Bu tez çalışmasında hedef nesneyi takibe başlamadan önce, nesneyi bulma amacıyla CamShift yöntemi kullanılmıştır. Nesne bulma aşamasına girdi olarak takip edilecek hedef nesnenin renk dağılımı verilmektedir. Nesne bulma adımında ilk olarak

renk dağılımı verilen hedef nesnenin histogramı çıkarılmaktadır (Eşitlik 4. 13). Daha sonra çıkarılan histogram değerleri [0-255] arasına normalize edilmektedir (Eşitlik 4. 14). Elde edilen bu normalize edilmiş histogram moment hesaplamalarında kullanılmak üzere saklanmaktadır. Daha sonra alınan görüntü üzerinde sıfıncı ve birinci momentler hesaplanmaktadır (Eşitlik 4.8, 4.9, 4.10).

$$M_{00} = \sum_x \sum_y P(x, y) \quad (4.8)$$

$$M_{10} = \sum_x \sum_y xP(x, y) \quad P(x, y) \rightarrow \text{Olasılık Yoğunluk Görüntüsü} \quad (4.9)$$

$$M_{01} = \sum_x \sum_y yP(x, y) \quad (4.10)$$

Moment hesabından sonra bu momentlerden yararlanılarak hedef nesnenin görüntüde yer aldığı olası merkez bulunur (Eşitlik 4.11, 4.12).

$$X_c = M_{10}/M_{00} \quad (4.11)$$

$$Y_c = M_{01}/M_{00} \quad (4.12)$$

Belirlenen iterasyon sayısı kadar döndürülür; Bu iterasyonlar boyunca ilgili görüntüde momentler ve yeni merkez hesapları yapılır. Elde edilen merkez koordinatlarından ve hedef nesnenin olası boyutlarından yararlanılarak hedef nesnenin yer aldığı aday bölge belirlenir. Daha sonra bu bölgedeki renk olasılık yoğunluk dağılımı bulunur. Elde edilen bu dağılımla hedef nesnenin renk olasılık yoğunluk dağılımının benzerliği Bhattacharyya uzaklığı kullanılarak hesaplanır. Hesaplanan benzerlik değeri, daha önceden belirlenmiş olan bir eşik seviyesi değerinin üzerindeyse aday bölgenin hedef bölge olduğuna karar verilir. Hedef nesnenin bulunduğu yer bilgisi nesne takibi aşamasına aktarılır ve bu noktada nesne takibi aşaması başlar. Aday bölge ile hedef nesne benzerliğinin eşik seviyesinin altında olması durumunda ise nesne bulma aşaması bir sonraki görüntüyü alarak çalışmaya devam edilir.

#### 4.5.3.2. CamShift Yöntemi ile Nesne Takibi

Sistemin uygulama aşamasında değişik problemlerle karşılaşılmış, değişik senaryolar üretilmiş ve bu durumlara karşı çeşitli çözümler üretilmeye çalışılmıştır. Bu tez kapsamında öncelikle Mean-Shift yöntemi ile nesne takibi gerçekleştirilmiştir. Hareketli nesnenin kameraya olan uzaklığının değiştiği, nesnenin bir kısmının veya tamamının kaybolduğu durumlarda Mean-Shift yönteminin yetersiz kaldığı görülmüş ve bu durumlara karşı başarılı çözümler üreten CamShift yöntemi seçilmiştir.

CamShift algoritması, yalnızca statik dağılımlara çözüm getirebilen Mean-Shift algoritması temel alınarak üretilmiş efektif bir takip algoritmasıdır. Nesnenin olasılık

dağılımının yeri ve büyüklüğü; nesnenin hareketine, ışık değişimine, görüş açısının değişimine, gölgeye göre değişiklik gösterir. CamShift algoritması dinamik olarak değişen dağılımlar için tasarlanmıştır [57].

CamShift yöntemi kullanılarak tasarlanan nesne takibi sisteminde ardışık her bir görüntü nesnenin yerinin bulunması için kullanılan olasılık yoğunluk fonksiyonu renk bilgisinden yararlanılarak oluşturulur. Algoritma, ortalama yoğunluğun maksimum artış gösterdiği yönde iteratif olarak çalışarak dağılımın ortalamasını bulur. Olasılık yoğunluğu ardışık görüntülerin her birinde tekrar hesaplanmaktadır. Olasılık görüntüsündeki her bir piksel, o anki görüntüdeki ilgili pikselin renk değerinin hedef nesneye ait bölgedeki olasılığını temsil eder. CamShift yönteminde, Mean-Shift algoritmasından farklı olarak iterasyonlar boyunca dağılımın merkezine doğru moment hesabı yapılır [58]. Hedef bölgedeki renk dağılımını temsil etmek için literatürde parametrik ve parametrik olmayan çeşitli metotlar kullanılmaktadır. Histogram en sık kullanılan parametrik olmayan olasılık tahmin metodudur. Histogram, ilgili bölgedeki piksellerin renk dağılımı bulunarak elde edilir. Yapılan renk kuantalama işlemi sayesinde benzer renkler aynı gruba dahil edilir ve aynı olasılık değerine sahip olur. Bu sayede işlem yükü ve hafıza kullanımı azaltılmış olur [58] Ağırlıksız histogram, aşağıdaki gibi hesaplanır (Eşitlik 4.13):

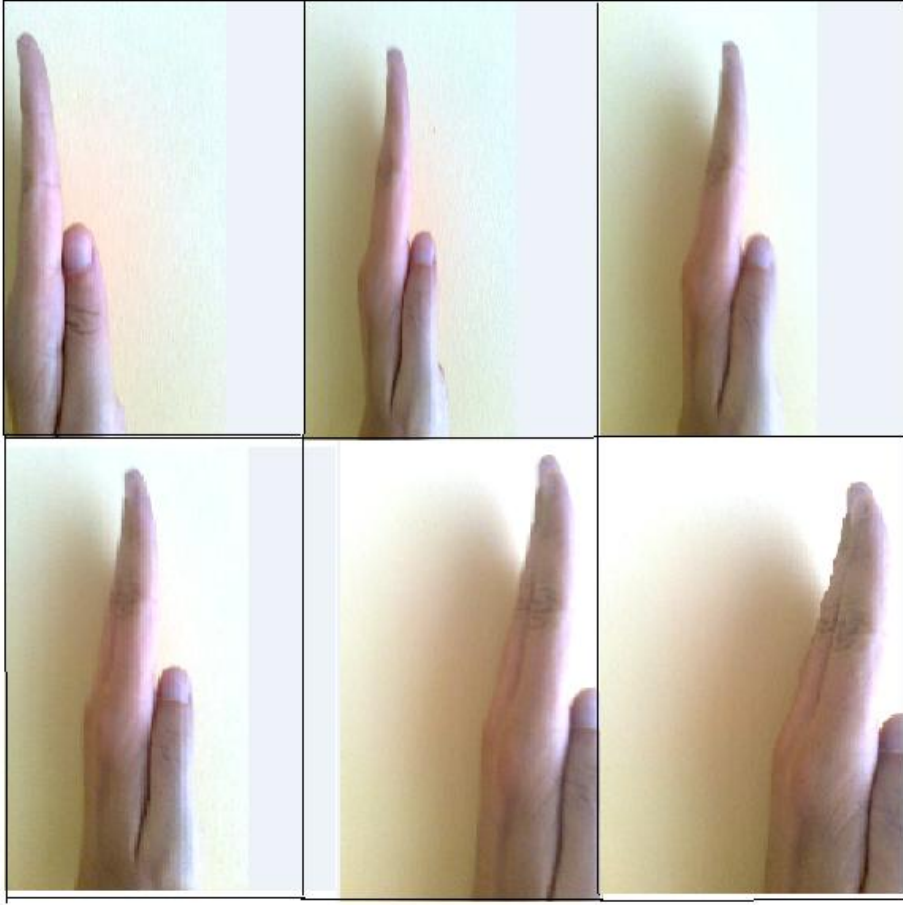
$$P_u = \sum_{i=1}^n \delta[c(x_i - u)] \quad (4.13)$$

$c(x_i)$  fonksiyonu  $c_1$  koordinatındaki pikselin değerini,  $n$  bölgedeki toplam piksel sayısını,  $\delta$  ise Kronecker delta fonksiyonunu simgelemektedir. Özellik olarak kullanılan nesnenin renk dağılımı, nesnenin kameraya olan uzaklığının artıp azalmasıyla değişiklik göstermez. Kenar tabanlı uygulamaların efektif olmadığı, nesnenin bir kısmının kaybolduğu durumlarda da renk dağılımı özelliği başarılı sonuçlar vermektedir. Olasılık yoğunluk görüntüsü  $P(x, y)$ , histogram çıkarımından yararlanılarak elde edilir. Bu işlem giriş görüntüsündeki piksel değerlerini o pikselin renk değerine ait histogram değeri ile değiştirir. Bu sayede olasılık görüntüsündeki her bir pikselin değeri o pikselin hedef modelde yer alma olasılığı ile temsil edilmiş olur. Olasılık değerlerinin 0 ile 255 arası değerler alabilmesi için histogram değerleri lineer olarak aşağıdaki formüldeki gibi normalize edilir (Eşitlik 4.14):

$$P_u = \min\left(\frac{255}{q_{max}} q_u, 255\right), u = 1, \dots, m \quad q_{max} = \{\max(q_u)\}_{u=1}^m \quad (4.14)$$

#### 4.6. CamShift Yöntemi ile Nesne Takibi Adımları

1. Arama alanının merkezi seçilir ( $x_0, y_0$ ).
2. Hedef nesnenin konumunu ( $x_1, y_1$ ) bulmak için Mean-Shift iterasyonları gerçekleştirilir, sıfıncı moment ( $M_{00}$ ) saklanır.
3. Arama alanı büyüklüğü sıfıncı momenti kullanan bir fonksiyon ile hesaplanır.
4. 2. ve 3. adımlar merkez koordinatların yer değişimi belirli bir eşik seviyesinin altında olduğu sürece tekrarlanır. Takip edilecek hedef nesnenin renk olasılık dağılımı girdi olarak alınıp nesne bulma işlemi ile nesnenin merkez koordinatları bulunduktan sonra CamShift ile nesne takibi aşamasına geçilir. Takip aşamasında gerçek zamanlı olarak alınan görüntülerde hedef nesneyi bulma amacı ile özellik olarak seçilen renk olasılık dağılımı aranmaktadır. Takip aşamasının ilk adımında, bir önceki adım olan nesne bulmadan gelen hedef nesnenin merkezinin görüntüde yer aldığı koordinatlar arama alanının merkezi olarak belirlenir. Bu alanda hedef nesnenin renk olasılık dağılımı CamShift yöntemi ile aranmaktadır. İlk olarak arama alanının sıfıncı momenti (Eşitlik 4.15),  $x$  ve  $y$  için birinci momentleri (Eşitlik 4.16, 4.17) bulunmaktadır. Aşağıdaki eşitliklerde görülen  $P(x, y)$ , arama alanındaki her bir pikselin olasılık dağılımını ifade etmektedir.



Şekil 4.11. Elin CamShift yöntemi ile takibi

Nesne takibi aşaması, nesne bulma aşamasından gelen hedef nesnenin bir önceki görüntüdeki yer bilgisini ve hedef nesnenin renk olasılık dağılım bilgisini kullanarak nesne takibine başlamasıdır. Nesne takibi aşamasında gerçek zamanlı olarak alınan ardışık görüntüler üzerinde anlık sonuçlar üretilmektedir. Hedef nesne, gerçek zamanlı olarak alınan her bir ardışık görüntü üzerinde CamShift yöntemi ile aranmaktadır. O anki görüntü üzerinde hedef nesnenin aranacağı alanın merkezi bir önceki görüntü üzerinde nesnenin bulunmuş olduğu yerin merkezi olacak şekilde belirlenir. Arama alanının büyüklüğü ise piksellerin renk olasılığı ve piksellerin yer bilgisi kullanılarak hesaplanan momentler üzerinden bulunmaktadır. Hedef nesne, büyüklüğü ve yeri belirlenen arama alanı içinde aranmaktadır. Arama işlemi yine CamShift yönteminde yer alan moment hesabı kullanılarak yapılmaktadır. Bu momentlerden yararlanılarak arama alanında hedef nesnenin renk olasılık dağılımının en yoğun olduğu bölge bulunmaktadır ve hedef nesnenin yeri bu bölgenin merkezi olarak belirlenmektedir. Yine momentlerden yararlanılarak hedef nesnenin görüntü üzerinde kapladığı alan bulunmaktadır. Hedef



nesnenin görüntü üzerindeki yer bilgisi, nesnenin görüntüde kapladığı alan ve arama alanının boyutu bir sonraki nesne takibi aşamasına aktarılır. Görüntü alımı devam ettiği sürece nesne takibi adımları da bu şekilde devam etmektedir.

$$M_{00} = \sum_x \sum_y P(x, y) \quad (4.15)$$

$$M_{10} = \sum_x \sum_y x P(x, y) \quad (4.16)$$

$$M_{01} = \sum_x \sum_y y P(x, y) \quad (4.17)$$

Daha sonra yukarıdaki eşitliklerden bulunan momentlerden yararlanılarak hedef nesnenin merkezi bulunur (Eşitlik 4.18, 4.19).

$$x_c = \frac{M_{10}}{M_{00}} \quad (4.18)$$

$$y_c = \frac{M_{01}}{M_{00}} \quad (4.19)$$

Belirlenen iterasyon sayısı kadar ilerlenip her iterasyonda sıfıncı ve birinci momentler hesaplanır ve bu momentler kullanılarak arama alanının merkezi güncellenir. İterasyonlar bittikten sonra nesneyi temsil eden pencerenin boyutları (geniřliđi ve yüksekliđi) Eşitlik 4.26, 4.27'den, arama alanı ise Eşitlik 4.28'den yararlanılarak yeniden hesaplanır. Bu hesaplamalar için öncelikle ikinci momentler bulunur (Eşitlik 4.20, 4.21, 4.22).

$$M_{11} = \sum_x \sum_y x y P(x, y) \quad (4.20)$$

$$M_{20} = \sum_x \sum_y x^2 P(x, y) \quad (4.21)$$

$$M_{02} = \sum_x \sum_y y^2 P(x, y) \quad (4.22)$$

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \quad (4.23)$$

$$b = 2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right) \quad (4.24)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \quad (4.25)$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (4.26)$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (4.27)$$

$$s = 2 \sqrt{\frac{M_{00}}{256}} \quad (4.28)$$

Hedef nesnenin geniř olan kenarını temsil etmek için  $l$  uzunluđu, kısa olan kenarını

temsil etmek içinse  $w$  uzunluğu kullanılır. Eşit kenarlı bir nesnenin bir kenarını temsil etmek içinse  $l$  uzunluğu kullanılır. Arama alanının boyutu ise  $s$  ile temsil edilir. Bu hesaplamalar yapıldıktan sonra nesnenin o anki görüntüdeki yeri, nesneyi temsil eden pencerenin boyutları ve arama alanının boyutları bulunmuş olur. Bu aşamada sıradaki görüntü işleme alınır. Yeni görüntüde arama alanının merkezi bir önceki adımda iterasyonlar sonucu bulunmuş olan merkez olarak güncellenir ve aynı adımlar Şekil 4.21'de görüldüğü gibi tekrar edilir.

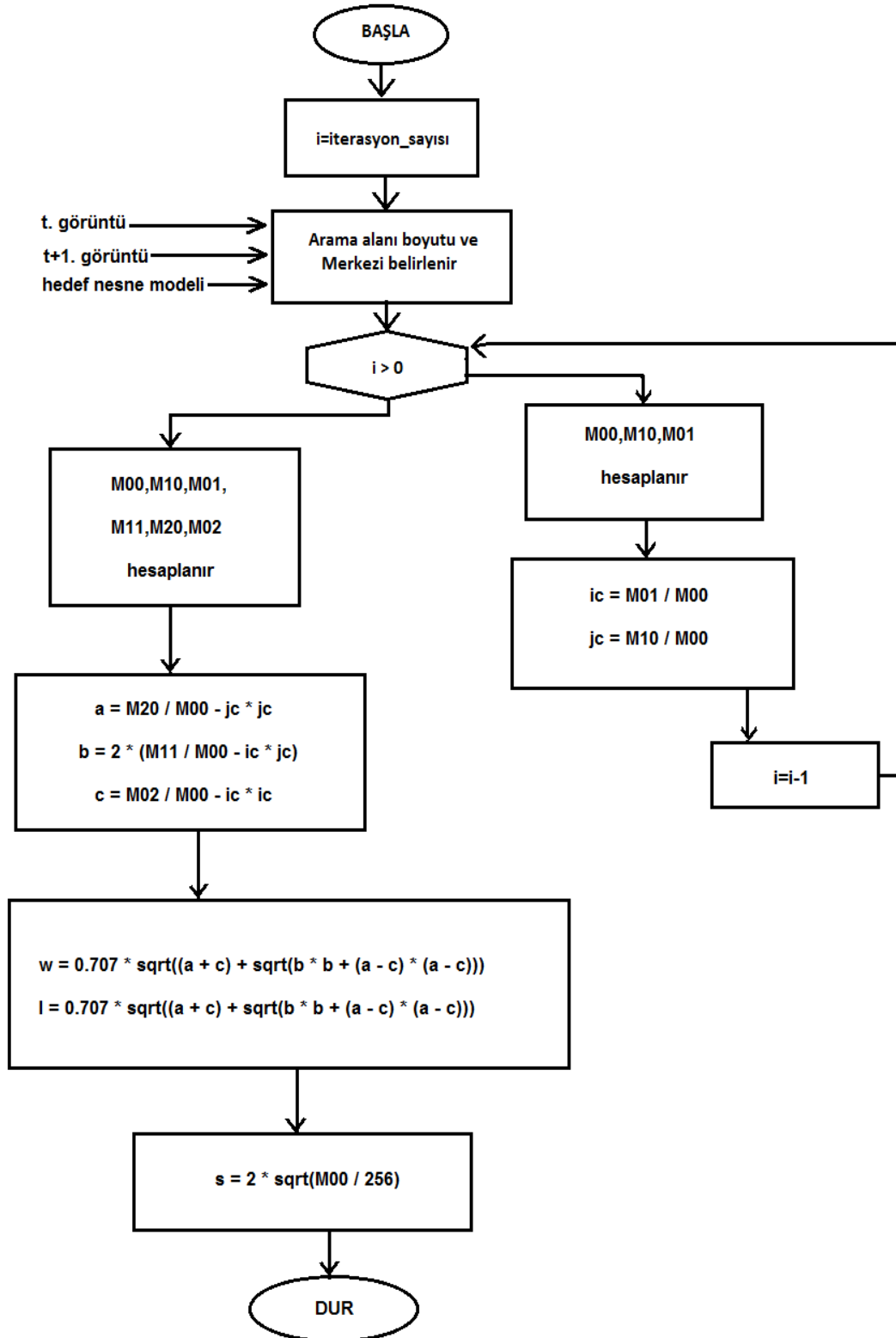
```

1      % Hedef nesnenin renk olasılık yoğunluk fonksiyonu hesaplanır (q)
2      % q [0-255] arasına normalize edilir (p)
3      while(1)
4      % Sıradaki görüntü kameradan alınır
5      i = 0;
6      % n. görüntü
7      end
8      for i=1:iterationNum:n
9      %(ic,jc) merkezli s genişlikli arama alanının
10     m00, m10, m01 momentleri hesaplanır
11     jc = m10 / m00;
12     ic = m01 / m00;
13     end
14     %(ic,jc) merkezli s genişlikli arama alanının
15     % m00, m10, m01, m11, m20, m02 momentleri hesaplanır
16     a = m20 / m00 - jc * jc;
17     b = 2 * (m11 / m00 - ic * jc);
18     c = m02 / m00 - ic * ic;
19     % hedef nesnenin genişliği ve yüksekliği hesaplanır
20     w = 0.707 * sqrt((a + c) - sqrt(b * b + (a - c) * (a - c)));
21     l = 0.707 * sqrt((a + c) + sqrt(b * b + (a - c) * (a - c)));
22     % arama alanının boyutu hesaplanır
23     s = 2 * sqrt(m00 / 256);

```

Şekil 4.12. CamShift yöntemi ile nesne takibi yarı kodlama [58].

### CamShift Yöntemi ile Nesne Takibi Akış Diyagramı:



Şekil 4.13. CamShift yöntemi ile nesne takibine ait akış diyagramı [58].

Sistemde açık çevrim kontrol tekniđi kullanılmaktadır. Bilgisayardan adım motoruna bilgiler gitmekte, fakat motorun hareketi ile ilgili bir bilgi bilgisayara gelmemektedir. Sistemde farklı kontrol tekniđi kullanılabilir fakat basit ve en ucuza mal olan kontrol tekniđi açık çevrim olduđu için çalışmada bu teknik kullanılmıştır. Sistemde kapalı çevrim kontrol tekniđi de iyi bir şekilde kullanılabilir. Fakat adım motoru kullanıldığı için kapalı çevrime ihtiyaç duyulmamaktadır. Çünkü adım motoru hassas işlemler için kullanılan doğruluđu yüksek bir motordur. Yalnızca adım hatasından kaynaklanan bir hataya sahiptir. Bu hata pek çok uygulamada yok sayılabilir.

## 5. BULGULAR

Sistem denetimi sayısal olarak bilgisayar ile gerçekleştirilmiş, MATLAB dilinde yazılan denetim algoritması kullanılarak deęişik alıřma durumları test edilmiřtir. Yazılan programda adım motorunun sargıları enerjilendirilirken motorun zaman sabitinin dikkate alınması gerekmektedir. Verilen gecikmeler motor zaman sabitinden daha kısa olursa motor adım atlayarak alıřmakta, sistem kendisinden beklenen iři yapmamakta ve titremeler meydana gelmektedir. Bu nedenle deneysel alıřmada en uygun süre tespit edilmiř olup 10-15 ms aralıklarla sargılara enerji verilmiřtir.

Yapılan alıřmada kullanılan g kaynaklarının sadece istenen gerilim deęil, aynı zamanda sistemin ihtiya duyduęu akımı saęlayacak řekilde tasarlanmaları gerektięi ortaya ıkmıřtır. Adım motorunun ihtiya duyduęu akım deęerini g kaynakları saęlamaktadır. Sistemdeki adım motoru maksimum 850 mA akım ekmektedir. Dzenlenen sistemde src devresinden adım motoruna maksimum 500 mA akım saęlanmaktadır. Motor milinin zorlayıcı bir yk bulunmadıęı iin 500 mA akım fazlasıyla yeterli olmaktadır.

Hedefte birden fazla nesne bulunması halinde bunların hangisinin izlenmesi gerektięi konusunda ortaya ciddi problemler ıkabilir. Bu alıřmada tek bir hareketli nesne kullanılmıřtır. Cismin řekli, boyutu, kameraya olan uzaklıęı ve ortamın parlaklıęı gibi bir ok faktr sistemden saęlıklı neticenin alınmasını engellemektedir.

## 6. TARTIŞMALAR

Bu çalışmada hareketli nesne izleme ve konum denetimi sistemi tasarlanmıştır. Sistemde açık çevrim kontrol tekniği kullanılmıştır. Sistem bilgisayar ile kontrol edilmektedir. Hareketli nesne görüntü işleme teknikleri ve adım motoru kullanılarak takip edilmektedir.

Sistemin doğruluğu oldukça iyidir. Bu nedenle açık çevrim kontrol tekniği bu sistemi kontrol etmek için yeterlidir. Bunun yanında daha etkin kontrol tekniği de bu sisteme başarı ile uygulanabilir. Sistemde ender olarak meydana gelen adım hataları, kapalı çevrim kontrol tekniği kullanılarak daha iyi bir şekilde kontrol edilebilir. Ancak bu durumda fazladan donanım ve maliyet ortaya çıkmaktadır.

Kullanılan adım motoru sayesinde hareketli nesne hassas bir şekilde takip edilebilmektedir. Diğer elektrik motorları ile aynı hassaslıkta işlem yapmak oldukça zordur.

Kurulan sistem gerçek zamanlı bir uygulamadır. Bu nedenle işlemleri hızlı bir şekilde yapması gerektiği için, kullandığımız Camshift algoritma her zaman istenilen sonucu vermeyebilir.

Çalışmadan elde edilen sonuçlara göre sistemde kısa sürede hedef tespit edilerek , takip edilmekte ve sistem hızlı çalıştığı için oldukça güvenilir sonuçlar vermektedir. Ayrıca sistemin bilgisayar destekli olması kısa sürede etkin bir şekilde sisteme müdahaleyi mümkün kılmaktadır.

## 7. SONUÇLAR ve ÖNERİLER

Bu çalışmada, standart bir web kamerası ve kişisel bilgisayar kullanılarak kamera görüş alanına giren nesnelere takibi sağlanmıştır. Çalışmamızda çoklu hareket ihtiva eden imge dizilerindeki hareket bilgilerini ortaya koymaya yönelik bir algoritma üzerinde durulmuştur. Nesne hareketi için öbek eşlemeye dayalı bir kestirim gerçekleştirilmiş ve objenin hareketi izlenmiştir. Algoritma içerisinde iki çerçeve arasında elde edilen kestirim değerlerinin gerçek değerlerden çok az sapma ile nesneyi izledikleri gözlenmiştir. Geliştirilen algoritma değişik imge dizileri üzerinde denenmiş ve hata seviyesine göre sapmanın yaklaşık aynı kaldığı tespit edilmiştir.

Bu tez kapsamında hareketli kameradan alınan görüntüler üzerinden gerçek zamanlı hareketli nesne takibi gerçekleştirilmiştir. Kamera hareketli bir düzencek üzerine yerleştirilmiştir. Sistem aynı anda hareketli kamera ve hareketli hedef nesne üzerinde çalışabilmekte ve hedef nesneyi bu koşullarda da başarıyla takip edebilmektedir. Sistem iki temel adımdan oluşmaktadır: Hareketli hedef nesnenin bulunması ve adım motor ile takibi. Sistemin çalışması nesne bulma aşamasıyla başlamaktadır. Hedef nesne sahneye giriş yapana kadar kameradan alınan görüntüler gerçek zamanlı işlenmekte, hedef nesne bu görüntüler üzerinde aranmaktadır. Hedef nesneyi görüntü üzerinde tespit etmek amacıyla CamShift algoritma yöntemi kullanılmıştır. Hedef nesne, görüntü bilgisi sisteme çalışma zamanından önceden aktarılmış olan herhangi bir hareketli nesne olabilir. Hedef nesneyi bulma amacıyla kullanılan özellik nesnenin renk olasılık dağılımıdır. Hareketli kameradan alınan ardışık görüntülerin her birinin renk olasılık dağılımı hedef nesnenin renk olasılık dağılımından yararlanılarak çıkartılmakta ve hedef nesnenin yeri görüntü üzerinde bulunmaktadır. Bu amaç doğrultusunda çalışılan görüntü üzerindeki bütün piksellerin momentleri renk olasılık ve yer bilgisi kullanılarak hesaplanmaktadır. Hesaplanan bu momentler ışığında, hedef nesnenin görüntü üzerinde bulunduğu aday bölge belirlenir. Bu aday bölge ile hedef nesnenin renk olasılık dağılım fonksiyonları, Bhattacharyya uzaklığı ile karşılaştırılır. Elde edilen benzerlik sonucu, önceden belirlenmiş sabit eşik seviyesinin üzerindeyse hedef nesnenin bulunmuş olduğuna karar verilir. Nesnenin elde edilen yer bilgisi nesne takibi aşamasına aktarılır ve nesne takibine başlanır. Eğer elde edilen benzerlik sonucu, eşik seviyesi değerinin altındaysa bu sonuç eşik seviyesi değerinin üzerine çıkana kadar nesne bulma aşamasına devam edilir.

Nesne takibi aşaması, nesne bulma aşamasından gelen hedef nesnenin bir önceki görüntüdeki yer bilgisini ve hedef nesnenin renk olasılık dağılım bilgisini kullanarak nesne takibine başlamaktadır. Nesne takibi aşamasında gerçek zamanlı olarak alınan ardışık görüntüler üzerinde anlık sonuçlar üretilmektedir. Hedef nesne, gerçek zamanlı olarak alınan her bir ardışık görüntü üzerinde CamShift yöntemi ile aranmaktadır. O anki görüntü üzerinde hedef nesnenin aranacağı alanın merkezi bir önceki görüntü üzerinde nesnenin bulunmuş olduğu yerin merkezi olacak şekilde belirlenir. Arama alanının büyüklüğü ise piksellerin renk olasılığı ve piksellerin yer bilgisi kullanılarak hesaplanan momentler üzerinden bulunmaktadır. Hedef nesne, büyüklüğü ve yeri belirlenen arama alanı içinde aranmaktadır. Arama işlemi yine CamShift yönteminde yer alan moment hesabı kullanılarak yapılmaktadır. Bu momentlerden yararlanılarak arama alanında hedef nesnenin renk olasılık dağılımının en yoğun olduğu bölge bulunmaktadır ve hedef nesnenin yeri bu bölgenin merkezi olarak belirlenmektedir. Yine momentlerden yararlanılarak hedef nesnenin görüntü üzerinde kapladığı alan bulunmaktadır. Hedef nesnenin görüntü üzerindeki yer bilgisi, nesnenin görüntüde kapladığı alan ve arama alanının boyutu bir sonraki nesne takibi aşamasına aktarılır. Görüntü alımı devam ettiği sürece nesne takibi adımları da bu şekilde devam etmektedir.

Hareketli nesnenin hareketli kamera ile gerçek zamanlı takibinde sistemin başarısını etkileyen bazı dış faktörler vardır:

1. Hedef nesnenin parlak bir cisim olması ve ortamdaki ışığın fazla olması durumunda hedef nesne üzerine gelen ışığı yansıtmakta ve bilinen renk olasılık dağılımı ile tamamen benzeşmemektedir. Bu durumda hedef nesnenin tamamı yerine parlama yapmayan diğer kısımları hedef olarak işaretlenmelidir. Bu duruma çözüm olarak hedef nesnenin değişik ışık seviyeli ortamlarda birden fazla görüntüsü nesne takibi aşamasına girdi olarak verilebilir.

2. Kamera kalitesi, ortamdaki çok fazla ışık değişimi veya ortamdaki ışığın yetersizliğinden dolayı bazı çekimlerde hedef nesnenin takibi sırasında görüntünün takipçi tarafından gözden kaçırılmasına neden olmuştur..

3. Hedefin top, kitap gibi bütünsel bir nesne olmadığı durumlarda, yani insan, robot gibi karmaşık bir yapıya sahip nesne olduğunda nesne takibine girdi olarak verilen hedef nesnenin görüntüsü gereksiz bilgi içermekte ve bu gereksiz bilgi de nesne takibi başarısını düşürebilmektedir. Bu problemi çözmek için, girdi olarak verilen hedef nesnenin



görüntüsünden arka plan kısmı elenebilir.

4. Takip edilen hareketli nesnenin hareketinin çok hızlı olması durumunda nesne gözden kaçabilmektedir. Bu durumda arama alanını geniş tutmak takip başarısını arttırmakla beraber sistemin çalışma hızını düşürecektir.

Sistem, bilgisayar yerine mikrokontrolör kullanılarak çok daha ucuza mal edilebilir. Bu durumda bilgisayara ihtiyaç yoktur. Fakat bu durumda sistem monitörden izlenemez.

### **7.1. Geleceğe Yönelik Çalışmalar**

Gelişen teknoloji ile beraber günümüzde kullanımı ve gereksinimi artan video işleme teknolojileri ile ilgili uygulamaların önümüzdeki dönemde hızla artması beklenmektedir. Sınır güvenliği, otomatik gözetleme sistemleri, trafik gözetleme ve kontrol sistemleri gibi güvenliğe yönelik uygulamalardaki gereksinimlerin video işleme tekniklerinin önemini artırması kaçınılmazdır. Çalışmada kullanılan algoritma ve genel yapıda gerçekleştirilecek değişiklikler ile performans artışı sağlanabilir bunun yanı sıra yüksek oranda performans artışı gerçekleştirebilmek için algoritma ve genel yapıdaki değişikliklere ek olarak güçlü fiziksel donanım yardımı gerekmektedir. Örnek verilecek olursa geliştirilen algoritmaya ait testler standart kişisel bilgisayarlar yardımı ile yapılmıştır. Algoritmanın yüksek işlemci gücüne sahip sistemler üzerinde çalıştırılması ile beraber büyük oranda performans artışı sağlanması beklenmektedir.

Bu sistem, uygulama yazılımında yapılacak küçük değişiklikler ile birden fazla kamera ile kontrol edilmesi gereken yerlerin görüntüsünün alınmasında kullanılabilir. Ayrıca, sisteme insan algılayabilen algoritmalar eklenebilir.

## 8. KAYNAKLAR

1. Acarnley, P. P., "Stepping Motors: a guide to modern theory and practice Revised 3<sup>rd</sup> ed.", Antsaklis, P. J., Atherton, D. P., Warwick, K., *Peter Peregrinus Ltd.*, England, 5, (1982), 19-22, 42, 50 .
2. Çeşitli, B., Adım Motorların Bilgisayarla Kontrolü, Y.Tezi, Pamukkale Üniversitesi, Denizli, 1998.
3. Condit R., Jones D. W., "Stepping Motor Fundamentals", *Microchip Technology Inc.*, United States, Literature Number: AN907, 2004.
4. Freitas, M. A. A., Andrade, D. A., Borges, T. T., Azevedo, H. R., "Driving the step motor with controlled phase currents", *Power Electronic Drives and Energy Systems for Industrial Growth, 1998, International Conference on*, 2 , 493-498.
5. Kilian, C. T., "Stepper Motors", *Modern Control Technology - Components and Systems 2<sup>nd</sup>ed.*, Thomson Delmar Learning, (2004), 367-368.
6. Mizutami, K., Hayashi, S., Matsui, N., "Modeling and control of hybrid stepping motors", *Industry Applications Society Annual Meeting, 1993, Conference Record of the IEEE*, 1, (1993), 289-294.
7. Toliyat, H. A., Kliman, G. B., "Motor Control", *Handbook of Electric Motors Revised and Expanded 2<sup>nd</sup> ed.*, Thurston, M. O., *Marcel Dekker, Inc.*, United States, 2004, 501-511.
8. S.Çoşar, O. Urhan, M.K. Güllü, "'CCD Kamera Kullanarak Sayısal Görüntü İşleme Yoluyla Gerçek Zamanlı Güvenlik Uygulaması", Otomasyon Dergisi, (2004), 82-84.
9. Lim J.S., "Two dimensional signal and image processing", *PrenticeHall*, 1990.
10. ÜSTÜN, M., "Adım Motorları Sürücü Tasarımı", Yıldız Teknik Üniversitesi F.B.E. Yüksek Lisans Tezi, İstanbul, Türkiye, 2000.
11. BAL, G.: "Özel Elektrik Makineleri", Ankara , Türkiye, 1997.
12. Yılmaz, A., O. Javed, M. Shah , "Object tracking: a survey", ACM Surveys, 38, (2006),13.
13. J. F. Canny, "A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*", 1986, pages 679-698.
14. Kalman, R. E., A New Approach to Linear Filtering and Prediction Problems. Trans. ASME, 82, (1960), 35
15. Isard, M. and Blake, A., *International journal of Computer Vision* , 29, (1998), 5.
16. MacCormick, J. and Blake, A. , A Probabilistic Exclusion Pirinciple for Tracking

- Multiple Objects. In proc. IEEE Computer Society Conference on CVPR,1999.
17. Rasmussen, C. and Blake, A. , a Probabilistic Exclusion Principle for Tracking Multi-Part Object, In Proc. IEEE Computer Society Conference on CVPR,1998.
  18. Intille, S., Davis, J. and Bobick, A. , Real-time Closed-world Tracking, In Proc. IEEE Computer Society Conference on CVPR,1997.
  19. Koller, D., Weber, J.W. and Malik, J., Robust Multiple Car Tracking with Occlusion Reasoning, In Proc. European Conf. Computer Vision, 1994.
  20. Blackman, S. S., Tucker, G. K. , Dempster, R. J. and Roszkowski, S. H. , Application of Multiple Hypothesis Tracking to Shipboard First Tracking, SPIE, 1996.
  21. Koch, W., IEEE Trans. Aerospace and Electronic Systems ,36, (2000), 1.
  22. Leonard, J. J., Moran, B. A. , Cox I.J. and Miller, M.L. , Underwater Sonar Data Using an Efficient Multiple Hypothesis Algorithm, In IEEE Int. Conf. Robotics and Automation, 1995.
  23. Polat, E., Tracking Multiple Object Using Multiple Hypothesis Tracking Framework PhD Thesis, Pennsylvania State University, Pennsylvania, 2002.
  24. Nickels, K. and Hutchinson, S., IEEE Trans. Robotics and automation, 17, (2001), 28.
  25. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME - Journal of Basic Engineering, 2006, 35-45.
  26. Yilmaz, A., Javed, O., Shah, M., "Object tracking: a survey", ACM Computing Surveys, 38, 2006, 13.
  27. Isard, M., Blake, A., CONDENSATION - Conditional Density Propagation for Visual Tracking. Int. J. Computer Vision, vol. 29, (1998), 15-28.
  28. Barrera, P., Jose, M. , Matellan, V., "Visual Object Tracking in 3D with Color Based Particle Filter", International Journal of Information Technology Volume 2 Number 1, 2005.
  29. Gennery, D. B., International Journal of Computer vision,8, (1992),113.
  30. Barron, J.L., Fleet D.J. ve Beauchemin, S.S., "Performance of Optical Flow Techniques", Int. Journal of Computer Vision, 12, 1994, 43-77.
  31. Lowe, D., International Journal of Computer vision, 8, (1992), 113.
  32. Bregler, C. and Malik, J., Tracking People with twist and Exponential Maps. CVPR.1998.
  33. Hager, G., Real-time Feature Tracking and Projective Invariances as a Basis for Hand-Eye Coordination, CVPR, 1994.

34. Weiss, I. and Ray, M., IEEE Trans. On PAMI, 23, (2001), 116.
35. Marchand, E., Bouthemy, P., Chaumette, F. and Moreu, V., Robust Real-time VisualTracking Using a2D-3D Model Based Approach, IEEE Int. Conf. On Computer Vision, 1999.
36. Jurie, F., Model-Based Object Tracking in Cluttered scenes with Occlusions Int. Conf. On Intelligent Robots and Systems, 1997.
37. Aggarwal, J. K. and Cai, Q., CVIU, 73, (1999), 428.
38. Gavrilla, D. M. and Davis, L. S., 3D Model-Based Tracking of Humans in Action: A Multi-View Approach, CVIU, 1996.
39. Moeslund , T. B. and Granum, E., CVIU, 81, (2001), 231.
40. Wren, C., Azarbayejani, A. , Darrell, T. and Pentland, A., PAMI, 19, (1997), 7.
41. Haritaoglu, I., Harwood, D. and Davis, L., Hydra: Multiple people Detection and tracking Using Silhouettes, CVPR, 1999.
42. Cheng, Y., IEEE Trans, PAMI,17,790(1995).
43. Fugunaga, K., Hostetler, L. D., IEEE Trans. Info. Theory, 21, (1975), 32.
44. Yilmaz, A., Shafique, K., Lobo, N., Olson, X. Li, T. and Shah, M., Target tracking inFLIR Imagery Using Mean Shift and Global Motion Compensation, IEEE Workshop Competer Vision Beyond Visible Spectrum, 2001.
45. Porikli, F. M., Human Body Tracking by Adaptive Background Models and Mean-shift Analysis, IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2003.
46. Zieren, J., Unger, N. and Akyol, S., Hands Tracking From Frontal View for Vision-Based Gesture Recognition, DAGM-symposium, 2002.
47. Brandsk, G. R., Computer Vision Face Tracking for Use in a Perceptual User Interface, IEEE Workshop on Applications of Computer Vision, Princeton, NJ,1998.
48. Collins, R. T., Mean- sift Blob Tracking Through scale Space, IEEE Conference on CVPR, 2003.
49. Lindeberg, T., International Journal of Computer Vision ,30, (1998),7.
50. Gonzalez, R.C. and Woods, R.E., Digital Image Processing SE, Addison-Wesley Publishing Company, USA, 1993.
51. Rudolf, K., Bit-Cutting, 1998.
52. Bellanger, M., Digital Processing of Signal: Theory and Practice, John Wiley and Sons, USA, 2000.

53. Cavallaro, A., Steiger, O., and Ebrahimi, T., Tracking video objects in cluttered background. *Circuits and Systems for Video Technology*, IEEE Transactions on. 1998.
54. Gonzalez, R.C. and Woods, R.E., *Digital Image Processing SE*, Addison-Wesley Publishing Company, USA, 1993.
55. Rudolf, K., *Bit-Cutting*, 1998, 27-30.
56. Bellanger, M., *Digital Processing of Signal: Theory and Practice*, USA, 2000.
57. Bradski, G. R., Microcomputer Research Lab, Santa Clara, CA, Intel Corporation, *Computer Vision Face Tracking For Use in a Perceptual User Interface*”, *Intel Technology Journal* Q2 '98, 2005.
58. Bogdan, K., “CamShift-Based Tracking in Joint Color-Spatial Spaces”, *CAIP 2005*, LNCS 3691, 2005, 693-700.

## 9. EKLER

### Ek 1.

```
clc;
disp('Running...');
close all;
clear;
vid=videoinput('winvideo',1);
triggerconfig(vid,'manual');
set(vid,'FramesPerTrigger',1);
set(vid,'TriggerRepeat', Inf);
start(vid);
for i=1:3
trigger(vid);
frames= getdata(vid,1);
figure; imshow(frames);
end
% -----
% Sabitler
WHITE = 255;
% yakınsama Eşiği
T = 1;
% Piksel sayısı arama penceresi ile genişletmek için
P = 5;
% -----
% Ortalama kayma
% İlk arama pencere boyutu
%W = [10 10];
W = [80 94];
% Arama penceresi başlangıç konumu
%L = [95 193];
L = [71 141];
% Hareket noktaları
Xpoints=[];
Ypoints=[];
disp('Frame: Coordinates');
for frame = 1:50,
filename = sprintf('%3.3i.jpg', frames);

R = frames;
% RGB uzaydan HSV uzayına dönüştürme
I = rgb2hsv(R);
% Renk bilgisini çıkarma
I = I(:, :, 1);
I = roicolor(I, 0.83, 1.0);
% Başlatma
oldCamL = [0 0];
MeanConverging = 1;
% -----

% Görüntü üzerinde arama penceresi kutusu oluşturma
```

```

for i = L(1) : L(1)+W(1),
    x = i;
    y = L(2);
    if x > size(I,1) || y > size(I,2) || x < 1 || y < 1
        continue;
    else
        R(x, y, :) = 0;
    end
end
for i = L(1) : L(1)+W(1),
    x = i;
    y = L(2) + W(2);
    if x > size(I,1) || y > size(I,2) || x < 1 || y < 1
        continue;
    else
        R(x, y, :) = 0;
    end
end
for i = L(2) : L(2)+W(2),
    x = L(1);
    y = i;
    if x > size(I,1) || y > size(I,2) || x < 1 || y < 1
        continue;
    else
        R(x, y, :) = 0;
    end
end

end

% -----
while MeanConverging,
    % Arama penceresinde kitle merkezini hesaplamak
    M00 = 0.0;
    for i = L(1)-P : (L(1)+W(1)+P),
        for j = L(2)-P : (L(2)+W(2)+P),
            if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
                continue;
            end
            M00 = M00 + double(I(i,j));
        end
    end
    M10 = 0.0;
    for i = L(1)-P : (L(1)+W(1)+P),
        for j = L(2)-P : (L(2)+W(2)+P),
            if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
                continue;
            end
            M10 = M10 + i * double(I(i,j));
        end
    end
    M01 = 0.0;
    for i = L(1)-P : (L(1)+W(1)+P),

```

```

for j = L(2)-P : (L(2)+W(2)+P),
    if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
        continue;
    end
    M01 = M01 + j * double(I(i,j));
end
end
xc = round(M10 / M00);
yc = round(M01 / M00);
oldL = L;
L = [floor(xc - (W(1)/2)) floor(yc - (W(2)/2))];
% Eşik değerini kontrol edilir.
if abs(oldL(1)-L(1)) < T || abs(oldL(2)-L(2)) < T
    MeanConverging = 0;
end
end
% MOO ve kitle merkezini bulabiliriz.
% Bu bilgi araştırma penceresinin boyutlarını değiştirmek için
% kullanılmıştır.
% Pencere boyutları ayarlanır.
s = round(1.1 * sqrt(M00));
W = [ s      floor(1.2*s) ];
L = [floor(xc - (W(1)/2)) floor(yc - (W(2)/2))];
% Kitle merkezinin koordinatları bilgisayar ekranında gösterilir.
disp(sprintf('%3i:   %3i, %3i', frame, xc, yc));
Xpoints = [Xpoints xc];
Ypoints = [Ypoints yc];
% Cismin üzerine artı işareti konur.
plus_sign_mask = [0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  1 1 1 1 1 1 1 1 1 1 1 1 1;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0;
                  0 0 0 0 0 0 1 0 0 0 0 0 0];
sizeM = size(plus_sign_mask);
for i = -floor(sizeM(1) / 2):floor(sizeM(1) / 2),
    for j = -floor(sizeM(2) / 2):floor(sizeM(2) / 2),
        if plus_sign_mask(i+1+floor(sizeM(1) / 2), j+1+floor...
(sizeM(2) / 2)) == 1
            R(i+xc, j+yc, :) = WHITE;
        end
    end
end
end
% -----
% Fotoğraf karesi üzerindeki olasılıkları gösterme
I = rgb2hsv(R);
S = [];
S(:, :, 1) = I(:, :, 1);
S(:, :, 2) = I(:, :, 1);
S(:, :, 3) = I(:, :, 1);
% Renk bilgisini elde etmek.
end

```



```

disp('AVI move parameters:');
% -----
plot(Ypoints,Xpoints, 'go' , Ypoints, Xpoints);
axis([0 320 0 240]);
% Kamerayı sağa döndürmek için
if 150-Xpoints < 0
    parport=digitalio('parallel','LPT1');
    line=addline(parport,0:3,'out');
    output1=[0 0 0 1;0 0 1 0; 0 1 0 0;1 0 0 0];
    for n=1:5
        for x=1:4
            pval1=output1(x,:)
            putvalue(parport,pval1);
            pause(0.3);
        end
    end
% Kamerayı sola döndürmek için
elseif 150-Xpoints>0
    parport=digitalio('parallel','LPT1');
    line=addline(parport,0:3,'out');
    output1=[1 0 0 0;0 1 0 0; 0 0 1 0;0 0 0 1];
    for n=1:5
        for x=1:4
            pval1=output1(x,:)
            putvalue(parport,pval1);
            pause(0.3);
        end
    end
% Kameranın sabit kalması için
else
    parport=digitalio('parallel','LPT1');
    line=addline(parport,0:3,'out');
    output1=[0 0 0 0;0 0 0 0; 0 0 0 0;0 0 0 0];
    end
% -----
cd ..
disp('Done. ');
% -----

```

**Ek 2.**

```
Command Window
Running...
Frame: Coordinates
1: 44, 96
2: 40, 97
3: 36, 97
4: 32, 97
5: 30, 97
6: 30, 97
7: 29, 97
8: 29, 97
9: 29, 97
10: 29, 97
11: 29, 97
12: 29, 97
13: 29, 97
14: 29, 97
15: 29, 97
16: 29, 97
17: 29, 97
18: 29, 97
19: 29, 97
20: 29, 97
```

## ÖZGEÇMİŞ

18.01.1984 tarihinde Trabzon'un Köprübaşı ilçesi Çifteköprü köyünde doğdum. 1996'da Çifteköprü İlkokulunu, 1999'da köprübaşı Orta Okulunu, 2002'de Sürmene Endüstri Meslek Lisesini aynı yıl kazandığım Fırat Üniversitesi Elektrik Öğretmenliği Bölümünü 2004 yılında bitirdim. Hâlen Karadeniz Teknik Üniversitesi Elektrik-Elektronik Mühendisliği bölümünde yüksek lisans yapmaktayım.

Sonay KADIOĞLU ile evliyim. Kitap okumaktan ve futbol oynamaktan hoşlanırım. Orta derecede İngilizce biliyorum. MATLAB programıyla tasarımlar yapacak derecede bilgisayar kullanabiliyorum. Bu konuda iki yıl araştırma yaptıktan sonra böyle bir projeyi hazırlamaya karar verdim.