

**ARDIŞIK GÖRÜNTÜLER İLE PİKSEL ALTI BİLGİ
ÇIKARIMI VE ÇÖZÜNÜRLÜK İYİLEŞTİRME**

**YÜKSEK LİSANS TEZİ
Müh. Evrim ÖZÇELİK
(504031535)**

**Tezin Enstitüye Verildiği Tarih : 21 Temmuz 2006
Tezin Savunulduğu Tarih : 26 Temmuz 2006**

**Tez Danışmanı : Yrd.Doç.Dr. Osman Kaan EROL
Diğer Jüri Üyeleri Prof.Dr. Muhittin GÖKMEN (İ.T.Ü.)
Doç.Dr. Işın ERER (İ.T.Ü.)**

TEMMUZ 2006

ÖNSÖZ

Öncelikle beni bu keyfili ve güncel tez konusuna yönlendiren, çalışmalarımın her anında bana zaman ayırıp her türlü konuda bana yol gösteren değerli hocam Yrd. Doç. Dr. Osman Kaan EROL'a çok teşekkür ederim.

Ayrıca tez çalışmamın deneysel bölümündeki robotik konusunda bana İTÜ Robotik Laboratuvarı'nı kullanmama izin veren Doç. Dr. Hakan TEMELTAŞ'a ve robotik çalışmamda benden bilgisini ve zamanını esirgemeyen, kendisinden çok şey öğrendiğim Araş. Gör. Murat YEŞİLOĞLU'na çok teşekkür ederim.

Beni yirmi beş sene önce dünyaya getiren, kişiliğimin oluşmasında ve bu günlere gelmemde en büyük emeği olan, şu ana kadar benden maddi ve manevi konuda desteklerini asla esirgemeyen ve bana olan inançlarını asla kaybetmeyen aileme ne kadar teşekkür etsem azdır.

En özel teşekkürü de bundan dört sene önce tanıştığım ve hayatımın her anını anlamlandıran, tez çalışmam sırasında en umutsuzluğa kapıldığım anlarda bile beni manevi olarak sürekli motive edip destek olan Araş. Gör. Şennur ÖKSÜZ'e iletmek istiyorum.

Temmuz 2006

Evrinm ÖZÇELİK

İÇİNDEKİLER

TABLO LİSTESİ	v
ŞEKİL LİSTESİ.....	vi
ÖZET.....	viii
SUMMARY	x
1. GİRİŞ	1
2. GENEL TANIMLAR	3
2.1 Çözünürlük Nedir?	3
2.2 Çözünürlük Modeli	4
2.3 Görüntüyü Yakınlaştırma.....	6
2.4 Çözünürlük İyileştirme Nedir?.....	7
2.5 Çözünürlük İyileştirme İşlemi	10
2.6 Mevcut Yöntemler	12
3. GELİŞTİRİLEN YÖNTEM.....	16
3.1 Amacı	16
3.2 Kamera Sistemlerindeki Donanımsal Kısıtlar.....	17
3.3 Bileşik Göz Yapılarının Çalışma Modeli.....	18
3.4 Yapılan Çalışma ile Bileşik Göz Yapılarının İlişkisi.....	21
4. YÖNTEMİN MATEMATİKSEL MODELİ.....	23
4.1 Tek Boyutta.....	23
4.2 İki Boyutta Dijital Kamera Kullanıldığında.....	25
5. SİMÜLASYON ÇALIŞMASI.....	29
5.1 Simülasyonun 1. Adımı.....	30
5.2 Simülasyonun 2. Adımı.....	32
5.2.1 Tek Yönde Tarama.....	33
5.2.2 Kare Şeklinde Tarama.....	33
6. DENEYSEL ÇALIŞMA	40
6.1 Çalışmanın Mimarisi.....	40
6.2 Çalışmanın Akış Diyagramı.....	42
6.3 Robot Kontrolü Çalışması.....	44
6.3.1 Çalışmanın Amacı.....	44
6.3.2 Robot Kolunun Bir Düzlem Üzerinde Hareket Ettirilmesi.....	44

6.3.3	Seri Manipölatör için Ters Kinematik Algoritması	45
6.3.4	Gerçeklenen Çalışma	48
6.3.4.1	Simülasyon Çalışması	48
6.3.4.2	Deneysel Çalışma.....	49
6.3.4.3	Kullanılan Kütüphaneler	49
6.3.4.4	Geliştirilen Kütüphaneler	50
6.3.4.5	Hassasiyet Denemeleri	55
6.3.5	PA10-7C Robot Kolunun Yapısı	57
6.4	Kamera Kontrolü Çalışması.....	58
6.4.1	Çalışmanın Amacı.....	58
6.4.2	Kullanılan Kameranın Yapısı.....	59
6.4.3	Kullanılan Resim Sayısallaştırıcının (Frame Grabber) Yapısı.....	59
6.4.4	MultiCam Sürücüsü	60
6.4.5	Gerçeklenen Çalışma	60
6.4.5.1	Deneysel Çalışma.....	60
6.4.5.2	Kullanılan Kütüphaneler	61
6.4.5.3	Geliştirilen Kütüphaneler	61
6.4.5.4	Hassasiyet Denemeleri	65
7.	DENEYSEL ÇALIŞMA SONUÇLARI	68
7.1	Karşılaşılan Problemler	68
7.2	Elde Edilen Sonuç	70
8.	SONUÇLAR VE TARTIŞMA	73
8.1	Simülasyon Sonuçları	74
8.2	Deneysel Sonuçlar.....	74
8.3	İleriye Yönelik Çalışma Önerileri.....	75
	KAYNAKLAR	77
	EK 1: PA10-7C Robot Kolunun Mekanik Yapısı	80
	EK 2: JAI CV-S3200 Marka Kameranın Teknik Özellikleri	81
	EK 3: Euresys Picolo Tetra Mimarisi	82
	EK 4: Geliştirilen Kütüphaneler	83
	ÖZGEÇMİŞ.....	101

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 5.1: Farklı görüntü modelleri için tek yönde tarama sonucunda elde edilen çözünürlük iyileştirmeleri.....	35
Tablo 5.2: Kare şeklinde 52 adımlık kaydırma sonucunda kamera çözünürlüğüne bağlı olarak elde edilen görüntüler	36
Tablo 6.1: Robot kontrol bilgisayarı donanım ve yazılım listesi.....	41
Tablo 6.2: Kamera kontrol bilgisayarı donanım ve yazılım listesi.....	42
Tablo 7.1: Farklı eşik değerleri için kamera görüntüsünün ve elde edilen sonucun karşılaştırılması.....	72

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1 : Hedeflenen çözünürlük iyileştirmesi; a) Düşük çözünürlüklü Satürn görüntüsü, b) Düşük çözünürlüklü Satürn görüntüsünün büyütülmüş hali, c) Önerilen yöntem ile hedeflenen çözünürlük iyileştirmesi.....	2
Şekil 2.1 : Düşük ve yüksek çözünürlüklü örnekleme	5
Şekil 2.2 : Çözünürlük modeli.....	6
Şekil 2.3 : Ara-değerleme ile üst örnekleme	7
Şekil 2.4 : Çözünürlük iyileştirme akışı	8
Şekil 2.5 : Görüntüde alt piksel ve tam piksel kaydırma.....	9
Şekil 2.6 : Görüntü yakalama modeli.....	11
Şekil 2.7 : Çok kare kullanarak çözünürlük iyileştirme	12
Şekil 2.8 : Görüntü yakalama modeli.....	13
Şekil 2.9 : Düşük ve yüksek çözünürlüklü görüntüler arasındaki örtüşme ilişkisi.	14
Şekil 3.1 : Çözülme istenen problem	16
Şekil 3.2 : Bileşik gözlerin yapısı	19
Şekil 3.3 : a) Böcek gözünün oluşturduğu görüntü, b) İnsan gözünün oluşturduğu görüntü.....	19
Şekil 3.4 : Ommatidium'un yapısı	20
Şekil 3.5 : Yapay gözün yapısı	21
Şekil 3.6 : Bileşik göz yapılarının çalışma prensibi	21
Şekil 3.7 : Çalışmada oluşturulan yapının çalışma prensibi.....	22
Şekil 4.1 : Tek boyutta sensor hareketi	23
Şekil 4.2 : Gerçek görüntünün kamera ile tek piksel olarak alınması	26
Şekil 4.3 : Sensor kaydırmalarına bağlı olarak incelenecek bölge üzerinde kapsanan alanlar	26
Şekil 4.4 : Gerçek görüntüden alınan tek piksel değerlerinin referans matrisi üzerine kaydedilmesi	27
Şekil 5.1 : Kameranin görüntüyü tarama biçimleri. a) Kare tarama; b) Tek yönde tarama	29
Şekil 5.2 : Kullanılan kamera pikselleri ve incelenen bölge ilişkileri	30
Şekil 5.3 : Sensorun görüntü üzerinde <i>deltax</i> ve <i>deltay</i> kaydırılması	31
Şekil 5.4 : 4 pikseli bir kameranin daire şeklinde bir bölgeyi kaplaması	31
Şekil 5.5 : 4 pikseli bir kameranin gerçek bir görüntüyü 2x2 adımda tarama simülasyonu	32
Şekil 5.6 : Kâğıt üzerindeki şeklin tek yönde tarama ile çözünürlüğünün iyileştirilmesi simülasyonu; a,b,c) Kamera ile yakalanan görüntüye farklı büyütme oranları uygulanması; d) Tarama işlemi sonrasında şeklin çözünürlüğünün iyileştirilmesi	34
Şekil 5.7 : Kare şeklinde tarama için kullanılacak 52x52 piksel boyutunda orijinal görüntü.....	36
Şekil 5.8 : Ekranın gerçek görüntüsü.....	37
Şekil 5.9 : Ekranın kamera ile yakalanmış görüntüsü	37
Şekil 5.10 : Ekranın yüksek çözünürlüklü görüntüsünün elde edilmesi	38

Şekil 5.11: Satürn gezegeninin gerçek görüntüsü	38
Şekil 5.12: Satürn gezegeninin teleskop ve kamera ile yakalanmış görüntüsü	39
Şekil 5.13: 32x32 adımda görüntünün taranması sonucunda elde edilen çözünürlük iyileştirmesi.....	39
Şekil 6.1 : Çalışmanın mimarisi	41
Şekil 6.2 : Çalışmanın akış diyagramı	43
Şekil 6.3 : Robot kolu eklem açıları ve uç nokta hız vektörü	44
Şekil 6.4 : k linkine ilişkin vektörler	45
Şekil 6.5 : N linkli seri manipülatör.....	46
Şekil 6.6 : Ters kinematik çalışmasının akış diyagramı	48
Şekil 6.7 : Robot kolunun simülasyondaki başlangıç konumu.....	49
Şekil 6.8 : Matris kütüphanesi sınıf diyagramı	51
Şekil 6.9 : Ters kinematik kütüphanesi sınıf diyagramı	52
Şekil 6.10: Robot kütüphanesi sınıf diyagramı	53
Şekil 6.11: Robotun uç noktasının tarama ızgarası üzerindeki dolaşımı	54
Şekil 6.12: PI kontrol devresi.....	54
Şekil 6.13: PI kontrolde Ki ve Kp'ye bağlı hareket modelleri	54
Şekil 6.14: Zamanlayıcı kütüphanesi sınıf diyagramı.....	55
Şekil 6.15: a) Robota yollanan komutlardan teorik hesaplanan konum değerleri; b) Eklemde alınan açı değerleri	56
Şekil 6.16: Teorik ve gerçek veriler üst üste bindirilmiştir. a) Teorik hesaplanan konum; b) Eklemde alınan açısal konum	56
Şekil 6.17: PA10-7C robotunun 3 boyutlu görüntüsü	57
Şekil 6.18: PA10 robotunun açık sistemi	57
Şekil 6.19: Kamera kontrolü protokolü.....	58
Şekil 6.20: Kullanılan kameranın şematiği.....	59
Şekil 6.21: Euresys marka Pico Tetra model resim sayısallaştırıcı	60
Şekil 6.22: Görüntü kütüphanesi sınıf diyagramı	62
Şekil 6.23: Resim sayısallaştırıcı kütüphanesi sınıf diyagramı.....	63
Şekil 6.24: Soket kütüphanesi sınıf diyagramı.....	64
Şekil 6.25: Kamera soket kütüphanesi sınıf diyagramı	64
Şekil 6.26: Kamera haberleşme diyagramı	65
Şekil 6.27: Flüoresan lamba ile gözlemlenen piksel değişimi	65
Şekil 6.28: Akkor lamba ile gözlemlenen piksel değişimi	66
Şekil 6.29: Halojen lamba ile gözlemlenen piksel değişimi	66
Şekil 6.30: Halojen lamba ile robot çalışırken gözlemlenen piksel değişimi.....	67
Şekil 7.1 : Bölge üzerine düşen ışık miktarının farklılık göstermesi, a) Işık miktarı daha fazla; b) Işık miktarı daha az.....	69
Şekil 7.2 : Kamera lensinden kaynaklanan görüntü bombeleşmesi	70
Şekil 7.3 : Deneysel çalışma sonuçları, a) Gerçek görüntü; b) Kamera görüntüsü; c) Oluşturulan yüksek çözünürlüklü görüntü; d) Yüksek çözünürlüklü görüntüye 220 eşik değeri uygulanması.....	71

ARDIŞIK GÖRÜNTÜLER İLE PİKSEL ALTI BİLGİ ÇIKARIMI VE ÇÖZÜNÜRLÜK İYİLEŞTİRME

ÖZET

Günümüzde birçok uygulama (plaka tanıma, yüz tanıma, tıbbi ve askeri uygulamalar, OCR, v.b.) görüntü işleme tekniklerini kullanmaktadır. Bu tür uygulamalarda başarımın yüksek olması görüntü kalitesiyle doğrudan ilişkilidir. Kaliteli görüntü almak için gereken kameralar ise uygulama maliyetini yükseltmektedir. Bu ihtiyacı karşılamak üzere düşük çözünürlüklü görüntüler bazı algoritmalarla yüksek çözünürlüklü görüntülere dönüştürülebilmektedir. Bu sayede basit kameralar kullanarak yüksek çözünürlüklü görüntüler elde edilebilmekte ve uygulamaların başarımları yükseltilebilmektedir.

Çözünürlük iyileştirme yöntemleri dijital görüntülemedeki donanımsal limitleri geçerek, var olan donanımdan daha yüksek çözünürlüklü görüntüler elde etmeyi hedeflemektedirler. Bu amaçla geliştirilmiş tüm yöntemlerde gerçek görüntü üzerinden alınan çok sayıda düşük çözünürlüklü görüntü karşılaştırılıp ara görüntü pikselleri hesaplanarak gerçek görüntünün yüksek çözünürlüklü görüntüsü elde edilmektedir.

Çözünürlük iyileştirme ile ilgili örnek bir yöntem olarak ara-değerleme (interpolation) tekniği gösterilebilir. Bu teknikte orijinal görüntüdeki komşu piksellerin arasına çeşitli algoritmalarla (bicubic, bilinear, v.b.) değerleri hesaplanmış yeni pikseller konulmakta, böylece toplam piksel sayısı artırılarak görüntü çözünürlüğü yükseltilmektedir. Ancak bu yöntem hesaplamaları bakımından tamamıyla yaklaşımlara dayanmaktadır.

Ara-değerleme dışındaki tekniklerde çözünürlük iyileştirmek için incelenecek bölge üzerinden kamera ile ardışık çekilmiş görüntüler kullanılmaktadır. Bu görüntüler işlenerek elde edilen piksel altı değerlerden yüksek çözünürlüklü görüntüler elde edilebilmektedir.

Çalışmada gerçekleştirilen çözünürlük iyileştirme yönteminin amacı görüntünün tümünün çözünürlüğünü arttırmak yerine standart bir kamerayla yakalanan düşük çözünürlüklü görüntüdeki çok küçük detayların (birkaç piksellik) uygun bir taramayla çok daha algılanabilir bir hale getirilmesini sağlamaktır. Tarama işleminde bir düzlem üzerine matris yapısında yerleştirilmiş ve belli bir alana yönlenmiş çok sayıda küçük görüntü sensörü modellenmekte ve alınan bu sensor değerleri işlenerek incelenen bölgenin yüksek çözünürlüklü görüntüsü elde edilmektedir.

Çalışmada geliştirilecek yöntemle benzerliğinden dolayı bileşik göz yapısına sahip böcek gözleri de incelenmiştir. Böcek gözlerinin görüntü yakalama konusunda mercekli gözlere göre birçok avantajı vardır; odaklama problemi yoktur, çok küçük ve hafiftir, aynı anda çok daha geniş görme alanına (örn. Panoramik görüntü) sahiptir, kırılğan değildir, ölçeklenebilir bir yapıya sahiptir ve görüntüdeki hareket

değişimlerini çok daha hassas bir şekilde algılayabilir. Bütünleşik gözlerin bu avantajlarını askeri ve sivil alanlara uygulamak üzere dünya çapında çeşitli çalışmalar yapılmaktadır.

Çalışmada simülasyon sonuçlarının yanı sıra deneysel bir sonuç da elde edilmiştir. Deneyler için Mitsubishi PA10-7C robot kolu ve Euresys Picolo resim sayısallaştırıcı (frame grabber) kullanılmıştır. Ters kinematik hesaplarıyla kontrol edilen robot kolu bir düzlem üzerinde tarama ızgarası adım aralıklarında hareket ettirilmekte ve her adıma ilişkin görüntü dijital bir kamera tarafından yakalanmaktadır. Elde edilen görüntüler geliştirilen yöntemle işlenerek, incelenen detay bölgenin yüksek çözünürlüklü görüntüsü elde edilmektedir.

SUB-PIXEL INFORMATION EXTRACTION WITH IMAGE SEQUENCES AND RESOLUTION ENHANCEMENT

SUMMARY

Nowadays, several applications (plate recognition, face recognition, medical and military applications, OCR, etc.) use image processing techniques. In such applications, the performance is directly related to the image quality. Cameras, required obtaining high-quality images, increase application costs. For this reason, low-quality images can be converted to high-resolution images with some algorithms. Thus, high-resolution images can be obtained by means of simple cameras and performances of the applications can be improved.

Resolution enhancement methods aim to get better high-resolution images than can be get from the existing hardware which degrades the image quality because of the hardware's limits. In all methods developed for this reason, several low-resolution images taken from the scene are compared to each other and sub-pixels are calculated to yield a high-resolution image of the scene.

Interpolation technique can be given as an example method of resolution enhancement. In this technique, new pixels, whose values are calculated with several algorithms (bicubic, bilinear, etc.) are placed among adjacent pixels of the original image, therefore the image resolution is increased by increasing the number of pixels. However, this method relies completely on approximations in terms of its calculations.

In order to obtain a high resolution image with a technique instead of interpolation, successive takings of image sequences are used over the scene to be examined. By processing these images, high-quality images can be achieved from the sub-pixel values obtained.

The aim of the resolution enhancement method performed in this study is making a detail area of the low resolution image captured by an ordinary camera more perceptible instead of increasing the whole image resolution. Many small image sensors, placed on a plane as a matrix framework and pointed to a definite area, capture many pixel values simultaneously and the high resolution image is obtained by processing these values.

Due to the similarity of the method to be developed in this study, feature of the insect eyes that can be considered as a compound eye are also examined. Insect eyes have several advantages over lens-type eyes in terms of image capturing; no focusing problems, tiny and light nature, providing a much larger visible area simultaneously (e.g. panoramic view), not brittle, scalable, and finally, much more sensitive to motion-based differences. Several studies are in progress all over the world to apply these advantages of compound eyes to military and civil areas.

In this study, experimental values as well as simulation results were obtained. Mitsubishi PA10-7C robot arm and Euresys Picolo frame grabber were used in the experiments. The robot arm, controlled by inverse kinematics calculations, is moved on a scanning grid over a plane, and the image corresponding to each step is captured by a digital camera. The images obtained are processed with the method developed, and the high resolution image of the area in question was obtained.

1. GİRİŞ

Dijital görüntüleme tekniklerinin çok farklı alanlarda kullanılmaya başlanmasıyla birlikte, uygulamalarda yüksek çözünürlüklü görüntü elde etme ihtiyacı da artmıştır. Artan bu ihtiyaç çözünürlük iyileştirme tekniklerinin gelişmesine ve donanımsal gelişmenin yerini almasını sağlamıştır. Bu tekniklerle elde edilen yüksek çözünürlüklü görüntüler sadece geniş ve iyi bir görünüm vermekle kalmamakta aynı zamanda görüntüye detay bilgilerin de katılmasını sağlamaktadırlar.

Günümüze kadar yüksek çözünürlüklü görüntü elde etmek için kullanılan en genel yaklaşım sensor üreticilerinin dijital kameraların piksel boyutlarını küçültürken birim alana daha çok piksel sığdırmalarına dayanmaktadır [1]. Bu yaklaşımda yüksek duyarlılıklı optik ve sensor kullanılmasından dolayı genel amaçlı ticari uygulamaların maliyetleri oldukça artmakta ve pikselde meydana gelen shot-noise etkisinden dolayı piksel boyları belli bir fiziksel limitten daha aşağıya çekilememektedir [2]. Bu donanımsal dezavantajları gidermek amacıyla tamamıyla hesaplamaya, matematiğe ve istatistiksel tekniklere dayalı çözünürlük iyileştirme (süper çözünürlük) yaklaşımları ortaya çıkmıştır.

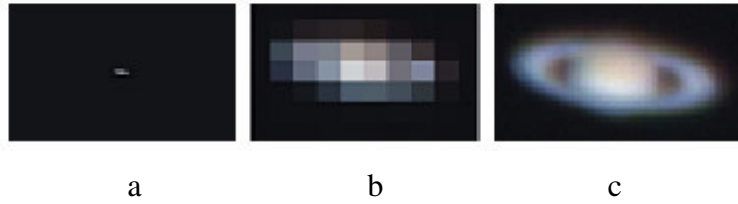
Klasik çözünürlük iyileştirme tekniklerinde [4-6] yüksek çözünürlüklü görüntü elde etmek için incelenecek bölge üzerinden dijital kamera ile art arda çekilmiş görüntüler kullanılmaktadır. Bu görüntüler arasında yapılan işlemler ile tek bir görüntüyle yakalanmayan alt piksel değerleri bulunmakta ve bunlar bir araya getirilerek yüksek çözünürlüklü bir görüntü elde edilmektedir.

Yapılan bu çalışmada dijital kameraların var olan donanımsal yapıları kullanılarak tek bir görüntü karesinde algılanamayan bölgelerde yüksek çözünürlüklü görüntüler elde etmek için yeni bir yöntem geliştirilmiştir.

Çalışmada geliştirilen yönteme benzerliğinden dolayı bütünleşik göz yapısına sahip böcek gözleri de incelenmiştir. Böcek gözlerinin görüntü yakalama konusunda mercekli gözlere göre birçok avantajı vardır; odaklama problemi yoktur, görünür

bölgenin dışındaki görüntüleri de yakalayabilir, çok küçük ve hafiftir, aynı anda çok daha geniş görme alanına (örn. Panoramik görüntü) sahiptir, kırılğan değildir, ölçeklenebilir bir yapıya sahiptir [7-8]. Bütünleşik gözlerin bu avantajlarını askeri ve sivil alanlara uygulamak üzere çeşitli çalışmalar yapılmaktadır [9-10].

Geliştirilmek istenen yöntem, dijital kamera ile yakalanan görüntülerdeki çok küçük bölgeleri çok küçük adımlarla tarayarak piksel altı bilgileri çıkartmakta ve böylece taranan bölgelerin çözünürlüklerini çok yukarılara çıkartıp o bölgeleri algılanabilir hale getirmeyi amaçlamaktadır. Örneğin uzay gözlemlerinde yakalanan yıldız ya da gezegen görüntüleri çok iyi bir teleskop kullanılmamışsa ancak birkaç piksel içine sığacak kadar büyüklükte olmaktadır. Bu durumda uygun bir tarama işlemi ile sadece o bölgeye odaklanarak gezegen ya da yıldız görüntüsü daha net hale getirilebilir. Şekil 1.1’de yöntemin temsili adımları görülebilmektedir.



Şekil 1.1: Hedeflenen çözünürlük iyileştirmesi; a) Düşük çözünürlüklü Satürn görüntüsü, b) Düşük çözünürlüklü Satürn görüntüsünün büyütülmüş hali, c) Önerilen yöntem ile hedeflenen çözünürlük iyileştirmesi

Bu tezin kalan kısmında yer alacak konular şunlardır: Bölüm 2’de çözünürlük modeli ve çözünürlük iyileştirme hakkında genel bilgiler verilecek ve daha önceden bu konuyla ilgili yapılmış çalışmalar tanıtılacaktır. Bölüm 3’de önerilen yöntem anlatılacak ve yapılan çalışmaya ilham kaynağı olan böcek gözlerinin çalışma prensibi ve avantajlarından bahsedilecektir. Bölüm 4’de yöntemin matematiksel modeli ve Bölüm 5’de simülasyon çalışmaları ve sonuçları anlatılacaktır. Bölüm 6’da deneysel çalışmanın genel mimarisi, altyapısı, deneysel çalışmanın robotik kontrol kısmı ve deneysel çalışmanın kamera kontrol kısmı anlatılacaktır. Bölüm 7’de deneysel çalışmada karşılaşılan problemler ve elde edilen sonuçlar verilecektir. Son bölüm olan Bölüm 8’de ise çalışmanın sonuçlarından ve ileriye yönelik geliştirme konularından bahsedilecektir.

2. GENEL TANIMLAR

2.1 Çözünürlük Nedir?

Çözünürlüğün farklı tanımları olmasından dolayı görüntü özelliklerini belirtirken oldukça karıştırılan bir terim olmaktadır. Görüntü çözünürlüğü basit bir ifadeyle görüntüdeki fark edilebilir veya ölçülebilir en küçük detay ölçüsüdür. Görüntü işlemede ve bilgisayarlı görüntülemelerde çözünürlük 3 farklı şekilde kullanılmaktadır [11].

— Uzamsal (Spatial) Çözünürlük

Bir görüntüdeki piksel aralığını ifade eder ve bir inch başına düşen piksel sayısı (ppi) ifade edilir. Uzamsal çözünürlük ne kadar büyükse görüntüdeki piksel sayısı da o oranda artar ve dolayısıyla piksel boyutları da küçülür. Bu da görüntünün daha fazla detay ve renk geçiş bilgisi taşıması anlamına gelmektedir. Uzamsal çözünürlük görüntüleme araçlarında inç başına düşen nokta (dpi) olarak ifade edilir ve cihazın çalışma sırasında oluşturduğu noktasal birimlerin boyutlarını belirler.

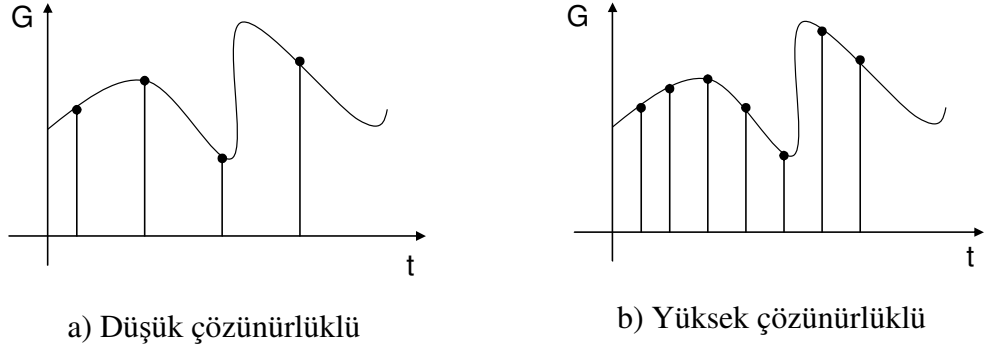
— Parlaklık Çözünürlüğü

Her piksel tarafından oluşturulabilen parlaklık seviyelerinin sayısını ifade eder. Bu da CCD (charge-coupled device) tarafından toplanan ışık enerjisinin sayısallaştırılmasıyla ilgilidir. Tek renkli görüntüler için parlaklık çözünürlüğü 256 seviyeden oluşmaktadır ve her seviye 8 bit ile ifade edilmektedir. Tam renkli görüntülerde ise parlaklık seviyesi en az 24 bit (örneğin kırmızı, yeşil, mavi renk düzlemleri için 8 bit) ile ifade edilmektedir.

Bir saniyede kamera tarafından yakalanan kare (frame) sayısıdır ve çoğunlukla kare hızı olarak bilinir. Geçici çözünürlük kareler arasındaki algılanabilir hareket miktarıyla ilgilidir. Yüksek kare hızları incelenen bölgedeki hareketlerden kaynaklanan lekelenmelerden daha az etkilenir. Geçici çözünürlüğün en düşük limiti çekilen iki kare arasında beklenen hareket miktarıyla doğrudan orantılıdır. Standart kameralar iyi bir sürekli görüntü sağlayabilmek için saniyede 25 kare civarında veya üzerinde yakalamaktadırlar.

2.2 Çözünürlük Modeli

Günümüzde modern görüntü sensorları olarak ışık kaynaklarına cevap verebilen CCD'ler kullanılmaktadır [12]. Yüksek yoğunluklu foto detektörlerden oluşan bir sensor yüksek uzamsal çözünürlükte görüntü yakalayabilmektedir. Fakat az sayıda foto detektörden oluşmuş bir sensor Şekil 2.1'deki örnekleme teorisine bağlı olarak ancak düşük çözünürlüklü görüntüler yakalayabilir ve ayrı pikseller arasındaki geçişler çıplak gözle dahi görülebilir. Bu durum örnekleme teorisinde olduğu gibi uzamsal çözünürlüğün uzamsal örnekleme oranı (örneğin belli bir yönde birim uzunluktaki foto detektör sayısı) tarafından sınırlandığını göstermektedir. Çözünürlüğü etkileyen bir diğer faktör de foto detektörlerin boyutudur. Basit bir yaklaşımla foto detektörlerin boyutunu küçülterek birim alana daha çok detektör sığdırılması ve böylece çözünürlüğün artırılması düşünülebilir. Fakat piksel boyutu küçüldükçe görüntü kalitesi oluşan gürültüleri (örn. shot noise) gidermek için yapılan iyileştirme işlemleri sonucu düşmektedir. Foto detektör boyutunun tahmin edilen en küçük değeri $50\mu\text{m}^2$ [6] olabilmektedir. Bu limite günümüz CCD teknolojileri erişilebilmektedir ancak bu boyutta bir sensor kullanmak nokta dağılım fonksiyonunun (PSF) nokta kaynağında bulanıklaşmaya neden olmaktadır. Bunun yanı sıra eğer örnekleme oranı çok düşük olursa örtüşme (aliasing) etkisinden dolayı görüntüde bozulmalar meydana gelmektedir.



Şekil 2.1: Düşük ve yüksek çözünürlüklü örnekleme

Çözünürlük modelini daha iyi anlayabilmek için şöyle bir örnek verilebilir. Bir düzlem üzerinde açılan küçük bir delik kamera gibi davranarak a boyundaki ve u uzaklığındaki bir cismi delikten f uzaklığında bulunan başka bir düzleme odaklamaktadır. Bu modelde bir kenarı x mm olan ve toplam N^2 adet pikseli olan kare bir sensor kullandığımızı düşünelim. Eğer bakma açısı Şekil 2.2’de olduğu gibi $\propto$ ile belirtilirse;

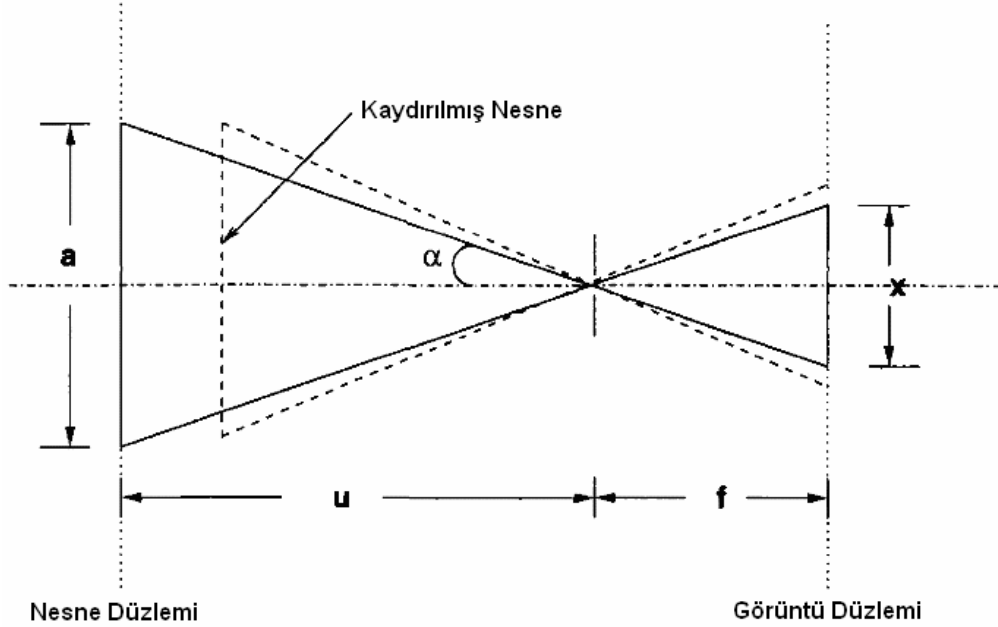
Bu modele göre $x=10$ mm ve $N=512$ olursa yaklaşık olarak 51 piksel/mm bir çözünürlük elde edilebilecek ve $u = fa/10$ mesafesindeki cisimler odaklanabilmektedir. Buna karşılık eğer cisim noktalı çizgi ile belirtilen yere kadar kameraya yaklaştırılırsa görüntü düzlemine daha önceki cismin bir kısmı düşmektedir. Böylece görüntünün düzlem üzerine düşen bölümünün daha yüksek çözünürlüklü değeri elde edilmiş olmaktadır [11].

Optik kuralları doğrultusunda çözünürlüğün en fazla ne olabileceği de belirlenebilmektedir. Optik sisteme giren toplam ışık enerjisi optik sistemin bir bölgesinde bulunan açıklık tarafından sınırlanmaktadır. Denklem 2.1’de görüldüğü gibi açıklık bir $a(x,y)$ fonksiyonu olarak tanımlanırsa optik transfer fonksiyonu olan $H(u,v)$ açıklık fonksiyonunun özilintisi (auto-correlation) olmaktadır [13].

$$H(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} a(x, y) a(x + u, y + v) dx dy \quad (2.1)$$

Eğer optik sistemdeki ışık geçişi mükemmel olarak düşünülürse açıklık içindeki bölgede $a(x,y) = 1$ olmakta ve açıklık dışındaki bölgede ise $a(x,y) = 0$ olmaktadır ve hiçbir ışık dalgası dağılmamaktadır. Dolayısıyla OTF, açıklık fonksiyonun özilintisi tarafından tanımlanan bölge dışında sıfıra gitmektedir ve açıklık bölgesinin dışındaki

tüm uzamsal frekans bilgileri kaybolmaktadır. Sonuç olarak açıklık içinden geçebilecek uzamsal frekansın maksimum limiti $f_c = \lambda F_1 / D$ denkleminde göre hesaplanabilmektedir. Denkleminde λ ışığın dalga boyu, F_1 optik sistemin odak uzaklığı, D dairesel açıklığın çapı ve f_c ise uzamsal kesim (cutoff) frekansıdır.



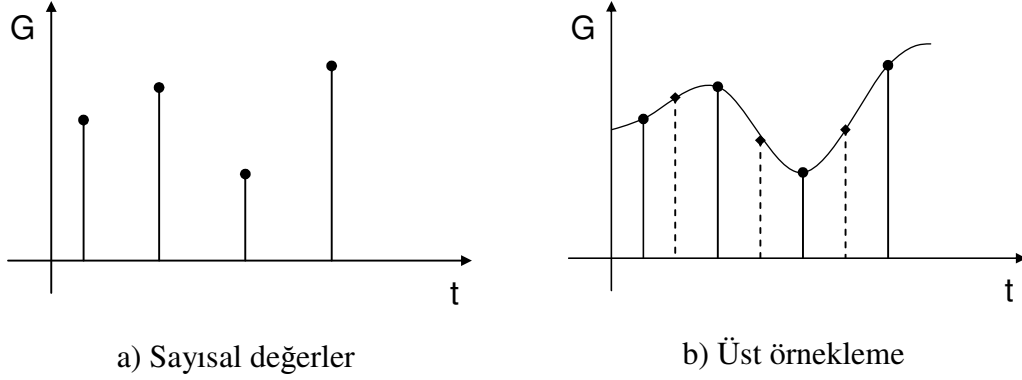
Şekil 2.2: Çözünürlük modeli

2.3 Görüntüyü Yakınlaştırma

Görüntü verisinin sadece boyutlarıyla oynamak çözünürlük artışı sağlamaz. Aslında boyut değişikliği orijinal görüntüden daha yüksek frekans değerlerinin hesaplanmasını içermelidir. Bu şekilde çözünürlük arttırmak amacıyla yapılan boyut değişikliği işlemine üst-örnekleme veya görüntü yakınlaştırma denilmektedir. Şekil 2.3’de görüldüğü üzere üst-örnekleme elde etmek için çoğunlukla ara-değerleme (interpolation) fonksiyonları kullanılmaktadır. Ara-değerleme orijinal veriyi sürekli bir fonksiyona uydurur ve daha küçük adımlarla tekrar örnekler [14].

Ara-değerleme yöntemleri ile görüntü yakınlaştırma işlemleri için ancak yeterince uygun bir model kullanılabilirse düşük çözünürlüklü örnekleme işlemi sırasında görüntüdeki yüksek frekanslı bileşenler kaybolmaz. Ara-değerleme yöntemleriyle görüntü yakınlaştırma işlemleri yaklaşımlara dayandığı için tek başına çözünürlük

iyileştirme teknikleri olarak değerlendirilmemektedirler. Bu konuda ileri düzey gelişmeler elde edebilmek için kullanılacak yöntemler giriş parametresi olarak birden çok görüntü alması gerekmektedir. Bu sayede bölgenin farklı görüntüleri incelenerek çözünürlük iyileştirmede kullanacağımız ek bilgiler hesaplanabilmektedir.



Şekil 2.3: Ara-değerleme ile üst örnekleme

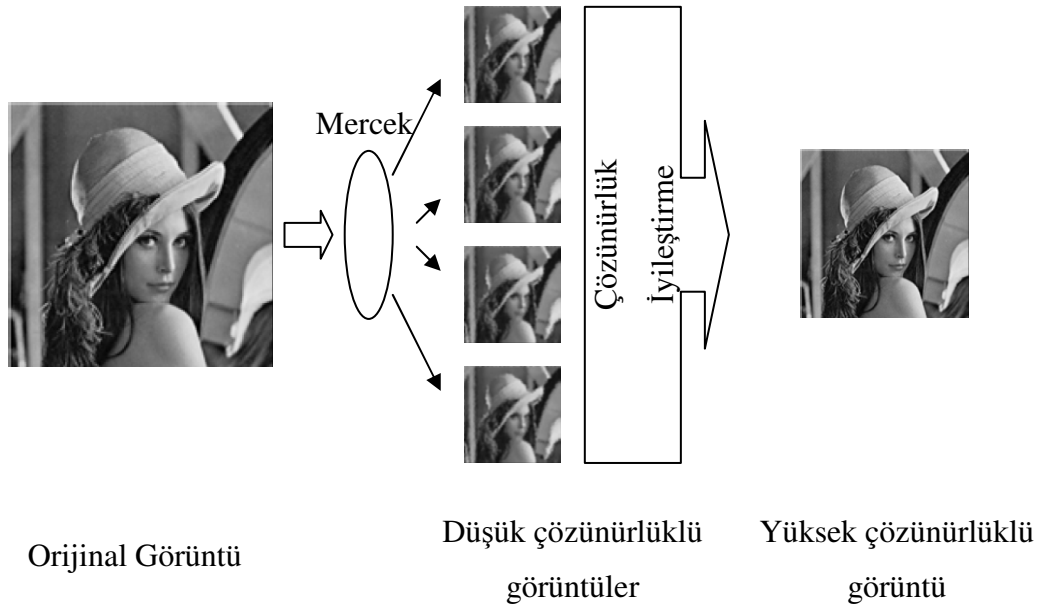
2.4 Çözünürlük İyileştirme Nedir?

Çoğu görüntüleme sistemlerinde, dijital kameralarda kullanılan sensorların sağladığı çözünürlük bazı uygulamalar için yeterince yüksek değildir. Ayrıca görüntü yakalama işlemi sırasında çevreden kaynaklanan ek gürültüler, kamera lensinin mükemmel olmamasından kaynaklanan yansıma ve kırılma sorunları (nokta dağılım fonksiyonu ile ifade edilir) ve foto-detektörlerin elektronik yapısından kaynaklanan hassasiyet problemleri sonucu orijinal görüntü üzerinde birçok tahribat meydana gelir. Bunun dışında kullanılan birçok dijital kamera sensorların fiziksel limitlerinden dolayı beklenenin çok altında bir çözünürlükte kayıt yapabilmektedir. Günümüzde araştırmalar ilerledikçe sensorların fiziksel limitleri çok daha yukarılara çıkarılmaktadır, fakat geliştirilen ürünler hem çok pahalıdır hem de çoğu mobil uygulamada kullanılmaya yeterince uygun değildir [15].

Çözünürlük iyileştirme çalışmaları kameranın kaydedebildiği görüntüden çok daha yüksek çözünürlüklü görüntüler elde etmeyi amaçlamaktadır. Çünkü gerçek görüntüler çoğunlukla keskin kıyı noktalara sahiptirler ve sınırlı bantlı (band-limited) değildirler. Sonuç olarak dijital görüntüler alt örneklemeden dolayı örtüşmek değildirler, düşük çözünürlüklü sensor nokta dağılım fonksiyonundan (PSF) dolayı yüksek frekanslı bileşenlere sahip değildirler ve optik kusurlardan, odak

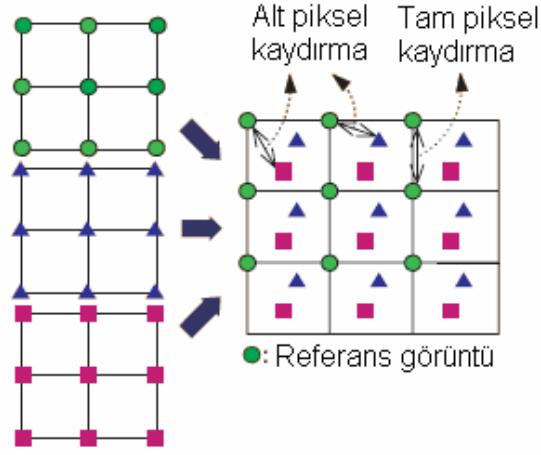
problemlerinden, hareket etkisinden ve benzeri sebeplerden dolayı bulanıktırlar [16-17].

Çözünürlük iyileştirme yöntemleriyle görüntü iyileştirmenin amacı elde edilen ardışık düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntüyü oluşturmaktır. Şekil 2.4’de sol taraf görüntünün düşük çözünürlüklü olarak yakalanması işlemini, sağ taraf ise düşük çözünürlüklü görüntülerin kullanılarak çözünürlük iyileştirme işlemini göstermektedir.



Şekil 2.4: Çözünürlük iyileştirme akışı

Çözünürlük iyileştirme çalışması çekilen ardışık görüntüler arasında ancak alt piksel hareketleri mevcutsa gerçekleştirilebilir. Dolayısıyla çekilen her görüntü incelenecek bölge üzerinde tekil bir görünüm sunmaktadır. Görüntülerde alt piksel hareketleri elde edebilmek için kameranın çok küçük adım hareketleri yapması gerekmektedir. Şekil 2.5’de referans bir görüntüye bağılı olarak alt piksel ve tam piksel kaydırma işlemleri görülebilmektedir.



Şekil 2.5: Görüntüde alt piksel ve tam piksel kaydırma

Görüntüler üzerinde yapılan çözünürlük iyileştirme çalışmaları, görüntü işleme alanındaki diğer birçok çalışmayla ilişkilidir. Bu sayede bir alanda oluşacak bir ilerleme doğrudan bu konudaki çalışmaları da etkilemektedir. Örneğin çözünürlük iyileştirme çalışmalarında yukarıda da kısaca bahsedildiği üzere kameranın alt piksel düzeyindeki kaydırmalarına ihtiyaç duyulmaktadır ve her kaydırmaya ilişkin incelenen alan üzerinden ayrı bir görüntü alınmaktadır. Alınan ayrı görüntülerde kayıklık dışında ışık ve gürültü miktarında da farklılıklar görülmektedir. Bu sebeple çekilen görüntülerin iyileştirilmesi ve birleştirildikten sonra restore edilmesi gibi görüntü işlemenin diğer alanlarındaki çalışmalara ihtiyaç duyulmaktadır. Bunun dışında çekilen görüntüler arasında herhangi bir kayıklık oluşturulmazsa bu durumda problem görüntü restorasyon problemine dönüşmektedir. Dolayısıyla çözünürlük iyileştirme tek kare veya çok kare görüntü restorasyon işlemlerinde de kullanılabilir [18].

Çözünürlük iyileştirme problemi ayrıca görüntüler arasında kayıklık olmadığında ve delta PSF fonksiyonu olduğunda gürültü filtreleme problemi olarak da düşünülebilir. Gerçek uygulamalarda yakalanan düşük çözünürlüklü görüntü dizilerinde gürültü ve değer sapmaları (outliers) oldukça yaygın olarak görülmektedir. Bu sebeple çözünürlük iyileştirme yöntemi çalıştırılmadan önce alınan veriler üzerinde ön bir çalışma yapılarak gürültülerden arındırılması algoritmanın sağlıklı sonuç vermesi açısından oldukça büyük önem taşımaktadır.

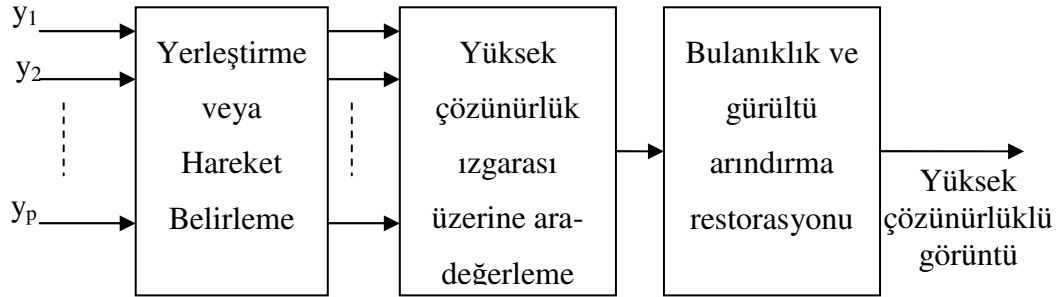
Çözünürlük iyileştirme teknikleri saptama, tanımlama veya sınıflandırma gibi başka görüntü işleme teknikleri için de oldukça büyük yararlar sağlamaktadır.

2.5 Çözünürlük İyileştirme İşlemi

Örnekleme oranının az olmasından kaynaklanan örtüşme etkisi özellikle kenar noktalar olmak üzere görüntü detaylarında bozulmaya neden olmaktadır. Buna ek olarak düşük çözünürlüklü nokta dağılım fonksiyonundan (PSF) dolayı görüntüdeki yüksek çözünürlüklü bileşenlerde kayıplar olmaktadır ve cisim hareketleri ve odaklama hatalarından kaynaklanan optik bulanıklaşmalar görülmektedir. Çözünürlük iyileştirme aynı zamanda giriş örnekleme oranını yukarı çekmeyi ve örtüşme ve bulanıklığı gidermeyi kapsamaktadır. Örnekleme oranını arttırmanın bir yolu sensordaki foto detektör sayısını arttırmak ve boyutlarını küçültmektir, böylece sensordaki detektör yoğunluğu arttırılmış olmaktadır. Ancak bu yöntem belli limitlere kadar geçerlidir, limitler aşıldığında görüntü üzerinde shot-noise etkisinden kaynaklanan büyük bozulmalar oluşmaktadır. Ayrıca yüksek çözünürlüğe uyumlu sensorlar günümüzde oldukça masraflıdır. Bu sebeple sensor üzerinde fiziksel modifikasyonlar yapmak her zaman en geçerli çözüm olamamaktadır. Dolayısıyla görüntü üzerinde çözünürlük iyileştirmesi yapabilmek için birtakım görüntü işleme tekniklerine başvurmak gerekmektedir. Gözlenen tek bir görüntüden yüksek çözünürlüklü görüntü elde etmeye çalışmak problemin yanlış ele alınmasını sağlar; çünkü eldeki görüntüden sonsuz sayıda orijinal görüntü ile ilişkili olabilecek görüntü elde edilebilir. Örnekleme oranını arttırabilmek için incelenecek bölge üzerinden birden çok görüntü alınır. Bu yöntemdeki en temel metot kamerayı çok küçük miktarlarda hareket ettirip çok sayıda görüntü yakalayarak tek görüntüde yakalanamayan alt piksel değerlerini hesaplamaktır [19].

Resim sayısallaştırıcılarının (frame grabber) kullanılmasıyla birlikte bir bölgenin birden çok görüntüsü kolaylıkla alınabilir hale gelmiştir ve çözünürlük iyileştirme bu noktadan sonra daha bilinen bir teknik olmuştur. Düşük çözünürlüklü kamera sisteminde ardışık yakalanan kareler arasındaki hareket miktarı çözünürlük limitlerini belirlemektedir. Resim sayısallaştırıcılar ile yakalanan komşu kareler yüksek çözünürlüklü görüntü için ek bilgiler taşımaktadırlar [21]. Çoğu çözünürlük iyileştirme metotları üç temel adımdan oluşmaktadır: (i) hareket dengeleme (ii) ara-

diyagramı görülmektedir. Blok diyagramında görünen adımlar günümüzde birçok çözünürlük iyileştirme yönteminde kullanılmaktadır.



Şekil 2.7: Çok kare kullanarak çözünürlük iyileştirme

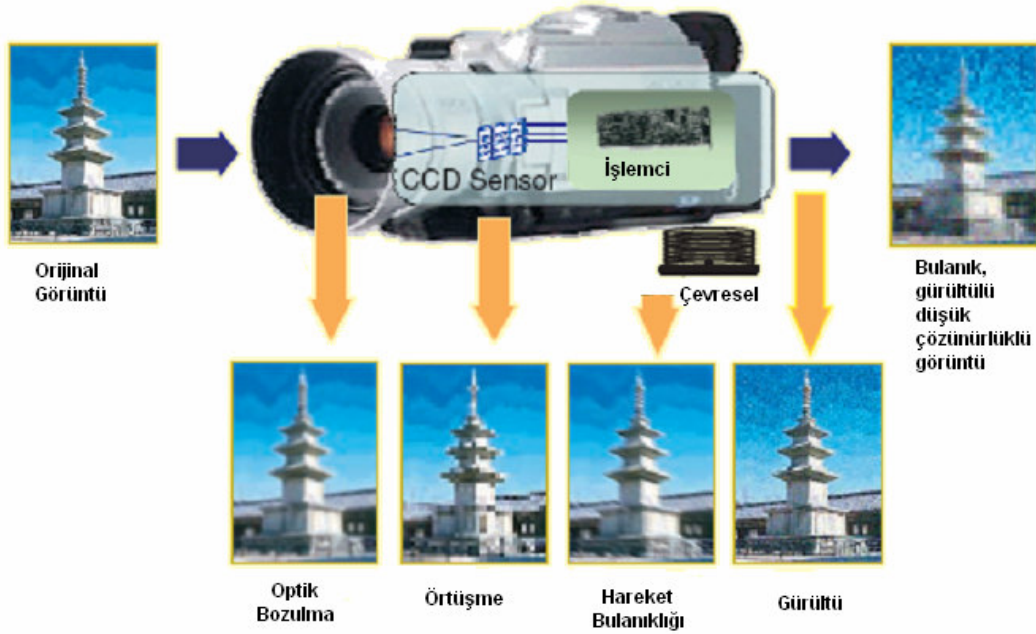
Burada ilk adımda $y_1, y_2, \dots, y_p$ ile belirtilen düşük çözünürlüklü görüntüler yerleştirme veya hareket belirleme modülüne yollanmaktadır. Sonraki adım hareket vektörlerine bakarak düşük çözünürlüklü görüntüleri hizalamakta ve bir yüksek çözünürlük ızgarasına yerleştirmektedir. Sonraki aşamada ara-değerlenmiş görüntüler bulanıklıklarından ve gürültülerinden arındırılmakta ve yüksek çözünürlüklü görüntü elde edilmektedir.

2.6 Mevcut Yöntemler

Bu bölümde mevcut çözünürlük iyileştirme yöntemlerinden bahsedilecektir ve anlatılan yöntemlerin tümü çalışmada geliştirilen yöntemin aksine görüntünün tamamının çözünürlüğünü arttırmaya çalışmaktadır.

Çözünürlük iyileştirmenin amacı çok sayıda düşük çözünürlüklü görüntüden bir tane yüksek çözünürlüklü görüntü oluşturmaktır. Çalışmalarda düşük çözünürlüklü görüntü kaynağı olarak arka arkaya çekilmiş durağan resimler veya video görüntüleri kullanılmaktadır. Hangi yöntemle elde edilirse edilsin, Şekil 2.8'den görüleceği üzere düşük çözünürlüklü görüntülerin her biri yakalanma sırasında fiziksel birçok bozulmaya (gürültüler, nokta dağılım fonksiyonu bulanıklığı, dönme hareketi, v.b.) maruz kalmaktadır. Kamera ile yakalanan görüntüler incelenen bölge üzerinde birtakım bilgiler bulundurlar. Eğer yakalanan görüntüler arasında farklı alt piksel kayma hareketleri oluşmuşsa, eldeki her görüntü orijinal görüntü hakkında farklı

detay bilgiler taşır. Eğer detay bilgiler yeterince elde edilebilirse düşük çözünürlüklü görüntülerden yüksek çözünürlüklü bir görüntü elde edilebilir olmaktadır.

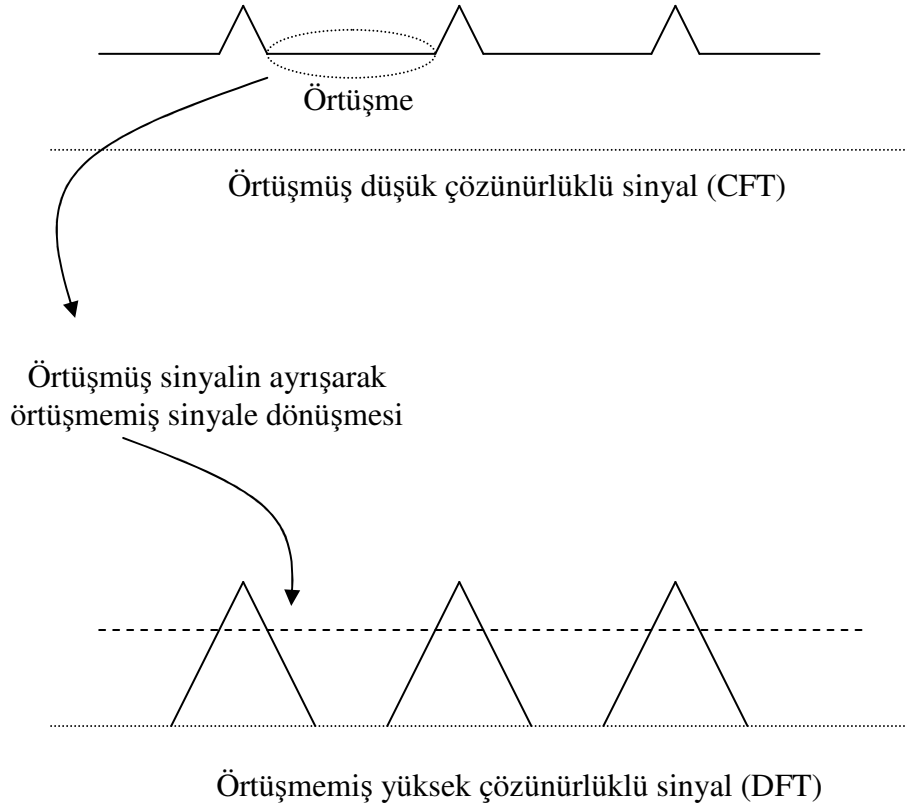


Şekil 2.8: Görüntü yakalama modeli

Düşük çözünürlüklü görüntü dizisinden yüksek çözünürlüklü görüntü elde etme fikri ilk olarak Tsai ve Huang [22] tarafından ortaya atılmıştır. Çalışmalarında sadece düzlemsel hareketi esas almışlardır ve yüksek çözünürlüklü görüntü oluşturma sırasında sadece hareket miktarı tayin etme ve restorasyon adımlarını gerçekleştirmişlerdir. Yaptıkları çalışmada görüntü bozulmaları ve gürültüler hesaba katılmamaktadır ve incelenen bölgenin sınırlı bantlı olduğunu kabul etmişlerdir. Metotlarını geliştirirken Fourier Dönüşümünün kayma özelliğinden ve Sürekli Fourier Dönüşümü (CFT) ile Süreksiz Fourier Dönüşümü (DFT) arasındaki Şekil 2.9'da görülen örtüşme ilişkisinden yararlanmışlardır. Kurdukları sistemde elde edilmek istenen yüksek çözünürlüklü görüntünün CFT'sinin uygun bir şekilde örneklenmesiyle elde edilecek DFT katsayılarının bulunması ve ters işlem ile yüksek çözünürlüklü görüntüye ulaşılması esasına dayanmaktadır.

Ortaya atılan bu yöntem Tekalp et. al. tarafından geliştirilerek kamera tarafından alınan görüntülerde Lineer Sabit Ötelemeli (LSI) bulanıklık Nokta Dağılım Fonksiyonu (PSF) etkisini de dahil ederek en küçük kareler yöntemiyle bir çözüm

önermiştir [23]. Daha sonra Kim et. al. da yönteme gürültü bileşenlerini de katarak ağırlıklı en küçük kareler yöntemiyle bir çözüm önermiştir. Kim ve Su elde edilen yöntem üzerinde farklı bulanıklık modelleri uygulamışlar ve bozulmalara karşı çözümler üretmişlerdir [24]. Şu ana kadar bahsedilen yöntemlerin ortak özelliği frekans kaynaklı olmalarıdır; bu sayede oldukça hızlı çalışabilmektedirler. Ayrıca şu ana kadar bahsedilen tüm metotlar görüntüde global bir hareketin olacağını varsaymışlardır.



Şekil 2.9: Düşük ve yüksek çözünürlüklü görüntüler arasındaki örtüşme ilişkisi

Karşılaşılan bir diğer yöntem ise kayma miktarları eş olmayan örnekler kullanarak yüksek çözünürlüklü görüntü oluşturmayı amaçlar. Naval Araştırma Laboratuvarı (NRL) tarafından TARID adı verilen bu yöntemde çekilen düşük çözünürlüklü kareler arasındaki optik akış miktarı hesaplanmakta ve ara-değerleme yöntemi ile daha büyük hale getirilmektedir, böylece yüksek çözünürlüklü görüntü oluşturulurken kullanılacak detay veriler ortaya çıkarılmaktadır. Son olarak görüntü restorasyon işlemleriyle görüntüleme cihazından kaynaklanan bulanıklıklar ortadan kaldırılmaktadır.

Bulanıklık giderme işlemi olarak benimsenen diğer bir çalışma da çok kanallı örnekleme teoremi kullanarak düzlem üzerinde kaydırılan çok sayıda görüntünün iyileştirilmesi Ur ve Gross tarafından Papoulis [25] ve Brown'ın [26] yöntemlerini geliştirerek ortaya çıkartılmıştır.

Irani ve Peleg'in [27] Döngülü Geri İz Düşüm adını verdikleri yöntemde, tomografide kullanılan geri iz düşüm yaklaşımına benzer bir yöntemi çözünürlük iyileştirmeye uyarlamışlardır. Bu yaklaşımda yüksek çözünürlüklü görüntü, hesaplanan düşük çözünürlüklü görüntüler ile gözlenen düşük çözünürlüklü görüntüler arasındaki hatanın geri iz düşümüyle tahmin edilmektedir. Bu işlem hata enerjisini minimize edene kadar döngülü bir şekilde devam etmektedir. Bu yöntemde bozulma matrisi olan h^{BP} sezgisel yöntemlerle elde edildiği için işlem sonrasında elde edilen sonucun en iyi sonuç olma garantisi yoktur.

Stark ve Osloui tarafından önerilen çalışma ise Dışbükey Kümeler Üzerine İz Düşüm (POCS) yöntemidir [28]. Bu yöntemde çözünürlük iyileştirme için en makul çözüm uzayını tanımlayacak kısıt kümeleri kullanılmaktadır. Kümeler incelenerek veriye uygunluk, kesinlik, kapsanan enerji, düzlük gibi bilgileri tutmaktadır. Çözünürlük iyileştirme içi kullanılacak çözüm uzayı ise kısıt kümelerinin kesişimleridir. Sonuca yaklaşım olarak bakıldığında döngülü (iteratif) bir yöntem olduğu görülmektedir.

Çözünürlük iyileştirme konusundaki bir diğer yöntem ise stokastik tekniklere dayanan ML (Maximum Likelihood) ve MAP (Maksimum a Posteriori) yaklaşımlarıdır [29-30]. Bu problemde çok kanallı yapılar içinde düşük çözünürlüklü görüntülerdeki kayma miktarı, bulanıklık ve gürültü gibi bilgileri tanımlayan denklemler vardır. ML yöntemi bu denklemleri kullanarak alt piksel kayıklıklarını çözer, gürültüleri temizler ve yüksek çözünürlüklü görüntüyü elde eder.

3. GELİŞTİRİLEN YÖNTEM

3.1 Amacı

Yapılan çalışmada kamera ile yakalanan görüntüdeki oldukça küçük bir bölgenin (örneğin 5x5 piksellik bir alanın) uygun bir tarama ile daha yüksek çözünürlüklü ve algılanabilir bir görüntüsü elde edilmeye çalışılmaktadır. Buradaki birincil amaç tüm görüntünün çözünürlüğünü arttırmak değil, belli bir bölgenin çözünürlüğünü çok yukarılara çıkarmaktır. Şekil 3.1’den de görüleceği üzere, kamera ile yakalanan düşük çözünürlüklü görüntü üzerindeki çok küçük boyutlarda algılanamayan bir bölgenin belli bir tarama işlemiyle arka arkaya çekilen diğer düşük çözünürlüklü görüntüleri işlenerek algılanabilir bir hale getirilmesi amaçlanmaktadır.



Şekil 3.1: Çözölmek istenen problem

Çözünürlük iyileştirme için önceki bölümlerde de anlatıldığı üzere ya kullanılan donanımın geliştirilmesi ya da aynı donanım ile birden çok sayıda alınan düşük çözünürlüklü görüntülerin işlenmesi gerekmektedir. Donanımın geliştirilmesi ek maliyetler doğuracağı gibi sensor ve mercek konusunda da belli limitler ortaya çıkartmaktadır. Önerilen yöntem ile kamerada donanımsal bir iyileştirme yapmadan, uygun bir tarama ile toplanan verilerden piksel altı değerlerin hesaplanması ve bu değerler ile daha iyi çözünürlüklü bir görüntünün yapılandırılması sağlanmaktadır. Yöntemdeki tarama işlemi için bileşik göz modelleri incelenmiştir.

3.2 Kamera Sistemlerindeki Donanımsal Kısıtlar

Günümüzde dijital görüntüleme teknolojilerinin ve görüntü işleme tekniklerinin gelişmesi ile birlikte birçok sektörel uygulama görsel verileri etkin olarak kullanır hale gelmiştir. Uygulamalarla birlikte en çok kullanılan görüntüleme cihazları ise dijital kameralardır. Bu kameralar sensorlarının önünde bulunan yüzeyleri yassı cam merceklerden kırılan ışığı algılayarak, bundan görüntü verisi oluşturmaktadır. Işığı kıran bu mercekler kalitelerine bağlı olarak oldukça iyi görüntüler yakalamayı sağlamaktadırlar, ancak kamera yapısında oldukça büyük bir yer kaplamaktadırlar ve ağırdırlar. Bunun dışında özel birçok uygulama bu cihazlardan geniş bir alandan düşük çözünürlüklü görüntü yakalamanın yanı sıra görüntüdeki detayları yakalayabilmek için daha dar alanlardan yüksek çözünürlüklü görüntüler de yakalamak istemektedirler.

Optik olarak kırılma esasına dayanan mercekli cihazlar istenilen bir bölgeye odaklanmak için mekanik bir yapıya sahip olmak zorundadırlar. Bu da cihazı ağırlaştırmakta ve daha fazla yer kaplamasına neden olmaktadır. Ayrıca uzaktaki bir noktaya odaklandıkları zaman, sadece odaklanan bölge net, odak dışındaki bölgeler ise bulanık olmaktadır. Bunun dışında mercekli cihazların bir kerede yakalayabilecekleri görüntü alanı da mercek boyutu ile kısıtlıdır.

Dijital kameralarda uzaktaki cisimleri de net olarak görüntüleyebilmek için cihaza sonradan eklenebilen ve birkaç mercek sisteminden oluşan dürbün aparatları da kullanılır. Dürbünlerin büyütme gücü çok basit bir formülle hesaplanır. Büyütme gücü, dürbünün yani objektifin odak uzaklığının göz merceğinin odak uzaklığına

bölünmesiyle bulunur. Bu basit formülünden anlaşılabileceği gibi, göz merceğini değiştirerek dürbünün büyültme gücünün değiştirilmesi mümkündür. Örneğin, 50 mm odak uzaklığına sahip bir dürbüne 5 mm odak uzaklığına sahip bir göz merceği takılırsa, 10 defa (10x) büyültme elde edilebilir.

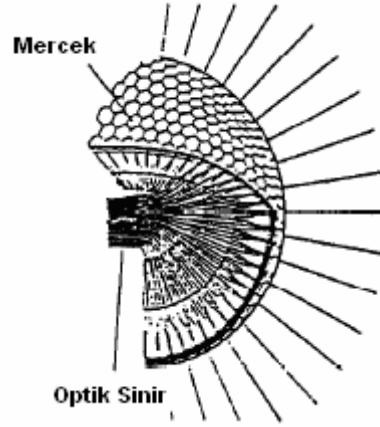
Objektifin ve göz merceğinin odak uzaklıkları istenildiği gibi ayarlanabileceğinden, kuramsal olarak büyültmenin bir sınırının olmadığı söylenebilir. Ancak pratikte bir takım sorunlarla karşılaşılır. Belirli çaptaki bir dürbünden, yeterli kalitede görüntü elde edebilmek için, büyültmenin de belirli bir sınırı aşmaması gerekir. Büyültme arttıkça, görüntünün parlaklığı ve ayrıntısı kaybolur. Hangi çaptaki dürbünle ne kadar büyültme yapılabileceğinin kesin bir formülü yoktur. Bununla birlikte kabul edilen bir oran vardır. Buna göre, yapılabilecek en fazla büyültme objektif çapının santimetresi başına 20x' dır.

Merceklerde üretim maddesi olarak cam kullanılması yapılarını oldukça kırılğan ve ağır yapmaktadır. Bunun dışında günümüzdeki üretim teknolojilerinin artmasına rağmen üretilebilecek düzgün bir cam mercek büyüklüğünün de bir limiti vardır. Mercek büyüklüğü arttıkça iyi bir yüzey oluşturmak zorlaşır ve maliyet artar.

Kamera sistemlerinde mercek dışında, elde edilecek çözünürlüğü doğrudan etkileyen diğer bir faktör de kullanılan sensordur. Sensorun birim alanında ne kadar çok foto detektör kullanılırsa çözünürlüğün de o oranda artacağı düşünülebilir. Ancak elektronik elemanlardan oluşan sensorlar belli bir boyutun altında üretildikleri zaman görüntüde shot-noise gürültüleri oluşturmaya başlamaktadır. Bu da çözünürlüğün artması yerine görüntünün bozulmasına neden olmaktadır.

3.3 Bileşik Göz Yapılarının Çalışma Modeli

Doğada mercekli gözlere sahip canlıların yanı sıra daha çok böceklerde görülen bileşik göz yapısına sahip canlılar da vardır. Kendi ölçülerinde oldukça üstün yeteneklere sahip bu tip gözler Şekil 3.2'de görüldüğü üzere birçok ayrık görsel birimden oluşmuşlardır. Her görsel birim merkezdeki sinir ucunda birleşmektedir. Böcek gözlerinin yapıları gereği hareket ve odaklama yetenekleri yoktur.



Şekil 3.2: Bileşik gözlerin yapısı

Bileşik gözlerdeki her bir görsel birim bakılan doğrultudaki ayrı bir bölgeyi algılar ve bu duyargalar tarafından alınan farklı veriler beyinde birleştirilerek tek bir görüntü elde edilir. Elde edilen görüntü Şekil 3.3’de görüldüğü üzere mercekli gözlerdeki gibi düzgün değildir, parça parçadır.

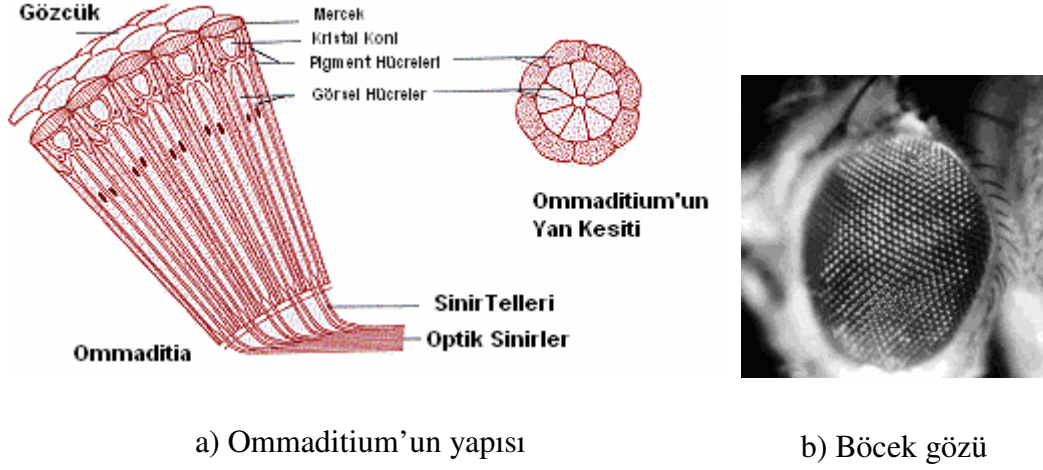


Şekil 3.3: a) Böcek gözünün oluşturduğu görüntü, b) İnsan gözünün oluşturduğu görüntü

Bileşik göz yapılarındaki her bir görsel birime ommatidia denilmektedir [31]. Her ommatidia şu bileşenlerden oluşmaktadır:

- Küçük bir mercek (ön kısımdaki yüzeyde fark edilen her bir gözcük)
- Geçirgen kristal koni

- Işığa duyarlı görüntü hücreleri (dizilimleri aynı portakalın iç bölümleri gibidir)
- Her bir ommatidium'u komşularından ayıran pigment hücreleri



Şekil 3.4: Ommatidium'un yapısı

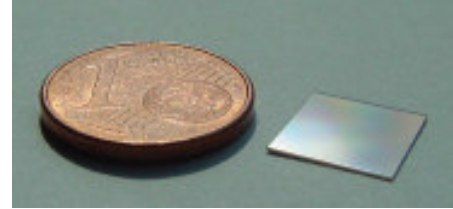
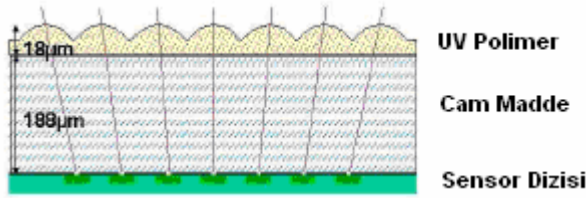
Ommatidium yapısındaki pigment hücreleri içeriği giren ışığın uzun eksenini boyunca doğrudan görsel hücrelere ulaşmasını sağlar ve gelen ışık dalgaları sinir hücrelerini tetikler. Her bir ommatidium uzayda küçük bir alan hakkında bilgi taşımaktadır. Bileşik göz yapılarında yüzeyde gözcükleri görülen binlerce ommatidia bulunmaktadır ve göz küresinin tamamını kaplamaktadırlar.

Bileşik gözler genellikle yakın mesafeleri iyi bir şekilde görebilmektedirler. Bu mesafe sineklerde ve örümceklerde birkaç milimetreye kadar düşmektedir. Bunun yanı sıra ancak yakını net görebilen bu canlılar insanın ancak mikroskopla görebileceği detayları görebilmektedir [32].

Böcek gözleri yakın detayları görmeyi yanı sıra uzak mesafedeki hareket değişimlerini ve karanlık-aydınlık geçişlerini çok hassas biçimde algılayabilmektedir. Bu sayede düşmanlarını çok kısa sürede algılayarak kendilerini korumayı başarabilmektedirler. Ancak bu hassasiyet böcekler için temiz bir görüşün kazandırır anlamına gelmemektedir.

Bunun dışında böcek gözlerinin odaklama problemi yoktur ve aynı anda daha geniş bir alandan (panoramik) görüntü yakalayabilirler. Ayrıca kırılma olmayışları, hafif

olmaları, Şekil 3.5b de görüldüğü gibi oldukça küçük boyutlarda üretilibilmeleri, mekanik bir aksama sahip olmalarına gerek olmayışı ve odak problemlerinin olmayışı bu konuda özellikle askeri alandaki takip sistemlerinde olmak üzere birçok farklı alandaki çalışmayı cazip hale getirmiştir.

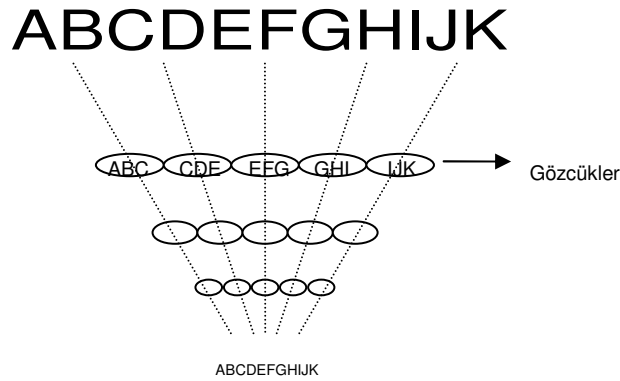


a) Yapay bir böcek gözünün yapısı

b) Yapay gözün üretimden sonraki hali

Şekil 3.5: Yapay gözün yapısı

Bileşik göz yapılarındaki görme sistemi Şekil 3.6'deki gibi modellenenbilmektedir.



Şekil 3.6: Bileşik göz yapılarının çalışma prensibi

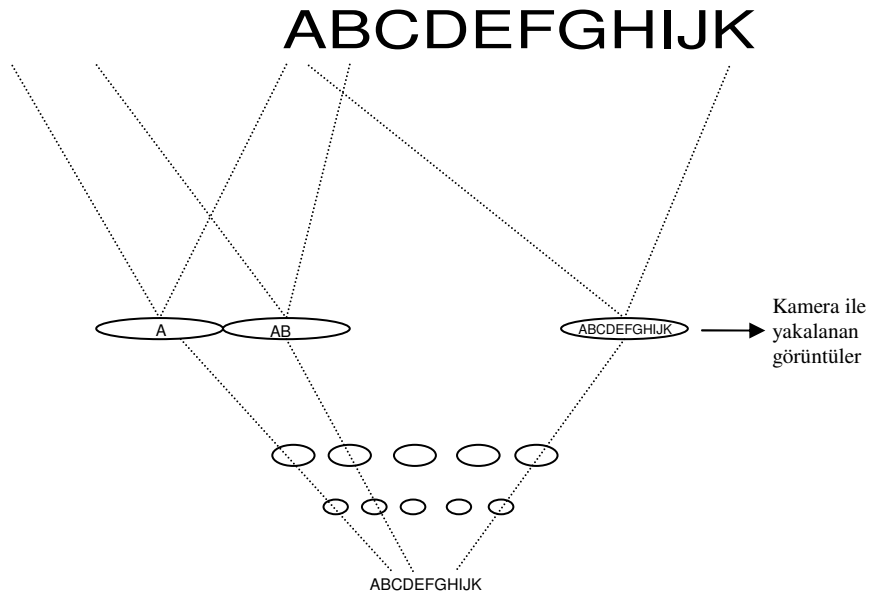
3.4 Yapılan Çalışma ile Bileşik Göz Yapılarının İlişkisi

Bileşik göz yapısına sahip böcek gözleri binlerce küçük gözcükten oluşmuştur. Her gözcük bakılan alanın belli bir bölgesini algılamaktadır. Farklı gözcükler tarafından algılanan veriler böcek beyninde kesleştirilerek 1 tane görüntü oluşturmaktadır. Dolayısıyla bu çalışma prensibi basit görsel yapılarla (her gözcüğün kendisine ait küçük bir merceği vardır) büyük bir mercek kullanmadan da görüntü oluşturulabileceğini göstermektedir.

Böcek gözlerini doğrudan mercekli gözlerle karşılaştırmak yanlış olabilir. Bileşik gözleri çalışma yapılarıyla dikkat çekilmek istenen nokta oluşturdıkları görüntü kalitesi yerine düşük seviyede görüntü algılayabilen gözcüklerden binlercesinin verisinin birleştirilmesiyle tek bir gözcüğün çözünürlük düzeyinin çok yukarılara çıkarılmasıdır. Önerilen yöntem var olan görüntüleme sistemlerinin yerine doğrudan böcek gözleri yapısında başka bir sistem getirmeyi amaçlamamaktadır. Burada çözülmek istenen problem var olan görüntüleme sistemlerindeki düşük çözünürlüklü çok küçük detayların uygun bir tarama işlemi ile daha algılanabilir hale getirilmesidir. Bileşik gözlerden esinlenilen nokta tarama işlemi sırasında alınan verilerin üst üste çakıştırılması özelliğidir.

Geliştirilen yöntem mercekleri ortadan kaldırmayı amaçlamamaktadır. Daha basit mercek sistemlerine sahip görüntüleme cihazları kullanarak daha büyük ve pahalı mercekli görüntüleme sistemlerinin yapabileceği işleri yapabilmek hedeflenmiştir.

Geliştirilen yöntemdeki tarama sistemi Şekil 3.7’deki gibi modellenenbilmektedir



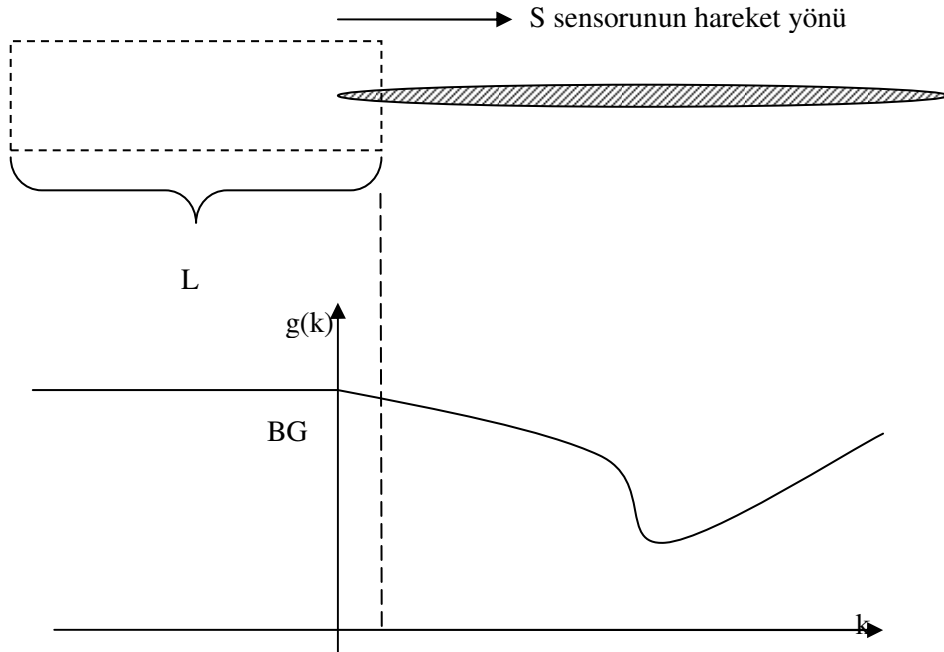
Şekil 3.7: Çalışmada oluşturulan yapının çalışma prensibi

4. YÖNTEMİN MATEMATİKSEL MODELİ

4.1 Tek Boyutta

Geliştirilen yöntemde yüksek çözünürlüklü görüntü oluştururken piksellerin ortalama değerlerinin hesaplanması yerine tam değerlerin hesaplanması hedeflenmiştir. Modeli daha iyi anlatabilmek için tek boyuttaki bir işaretin sensor kaydırma hareketlerine bağlı olarak taranmasını inceleyelim.

Şekil 4.1’de altta gösterilen eğri taranan bölgeye ilişkin grilik geçişini göstermektedir. S görüntü sensörü belirtilen yönde bölge üzerinde belli adım aralıklarında hareket ettirilmektedir. L sensorun ilerlenen adım birimiyle boyunu belirtmektedir. BG ise bölgenin sabit olarak kabul edilen arka fon grilik değeri olmaktadır.



Şekil 4.1: Tek boyutta sensor hareketi

Buna göre S sensorunun x ilerlemesinde sensordan ölçülecek değer Denklem 4.1'den hesaplanabilmektedir

$$S_x = \frac{\int_{k=x-L}^x g(k)}{L} \quad (4.1)$$

P_x , x kaydırma noktasındaki yüksek çözünürlüklü piksel değeridir. Formülden de görüleceği üzere eğer arka fon rengi sabit olarak kabul edilirse, P_0 S_0 'a eşit olmaktadır ve bölgenin Δx uzaklığındaki yüksek çözünürlüklü piksel değeri $P_{\Delta x}$ Denklem 4.2'den hesaplanabilir:

$$S_0 = P_0 = \frac{\int_{k=L}^0 g(k)}{L} = BG$$

$$S_{\Delta x} = \frac{\int_{k=\Delta x-L}^{\Delta x} g(k)}{L} = \frac{\int_{k=L}^0 g(k)}{L} - \frac{\left(\int_{k=0}^{\Delta x} BG - g(k) \right)}{L} = \frac{\int_{k=L}^0 g(k)}{L} + \frac{\left(\int_{k=0}^{\Delta x} g(k) - BG \right)}{L} \quad (4.2)$$

Bu işlemde Δx 'in çok küçük olduğu düşünülürse

$$S_{\Delta x} = \frac{\int_{k=L}^0 g(k)}{L} + \frac{(P_{\Delta x} - BG)}{L} = S_0 + \frac{(P_{\Delta x} - BG)}{L} \quad (4.3)$$

Denklem 4.3'den de görüleceği üzere kaydırma işlemi sırasında sensor bölgeyi taradıkça döngülü bir işlem dizisi ortaya çıkmaktadır. Bu özellik kullanılarak belli bir kaydırma adımında eğer önceki kaydırmalara ilişkin sensor değerleri bilinirse o adımdaki grilik seviyesinin daha hassas bir değeri hesaplanabilmektedir. Buna göre

$$S_0 = BG$$

$$S_1 = BG + \frac{P_1 - BG}{L}$$

$$S_2 = BG + \frac{P_1 - BG}{L} + \frac{P_2 - BG}{L}$$

$$S_3 = BG + \frac{P_1 - BG}{L} + \frac{P_2 - BG}{L} + \frac{P_3 - BG}{L}$$

$$S_x = BG + \frac{P_1 - BG}{L} + \frac{P_2 - BG}{L} + \frac{P_3 - BG}{L} + \dots + \frac{P_x - BG}{L} \quad (4.4)$$

Sensorun tamamıyla bölgeyi kapladığı durumda ise

$$S_L = BG + \frac{P_1 - BG}{L} + \frac{P_2 - BG}{L} + \frac{P_3 - BG}{L} + \dots + \frac{P_L - BG}{L}$$

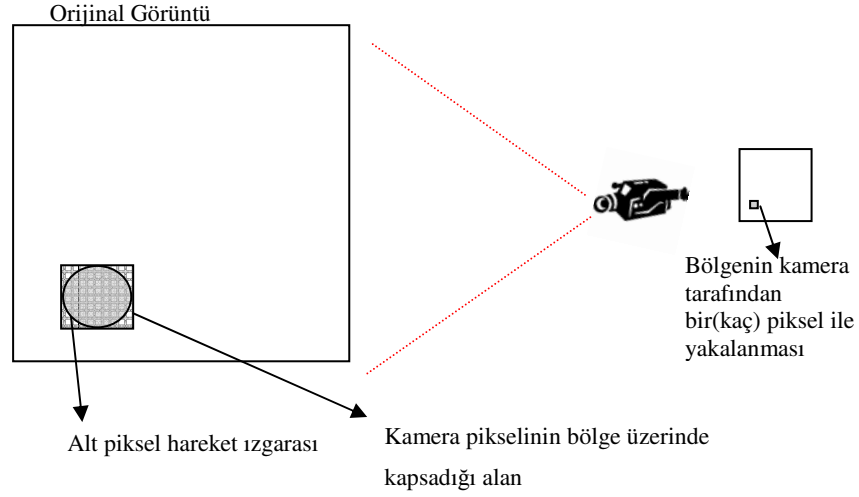
$$= \frac{P_1 + P_2 + P_3 + \dots + P_L}{L} \quad (4.5)$$

4.2 İki Boyutta Dijital Kamera Kullanıldığında

Tek boyutta olduğu gibi iki boyutta da tüm görüntü kameradaki piksel değerlerinin değişimlerine göre oluşturulacaktır. Eğer çözünürlüğü arttırılmak istenen bölge kamerada birkaç piksel tarafından algılanmışsa yöntemin çalışması için piksel değeri giriş parametresi olarak bölgenin kapsandığı piksel ortalamalarının verilmesi yeterli olacaktır.

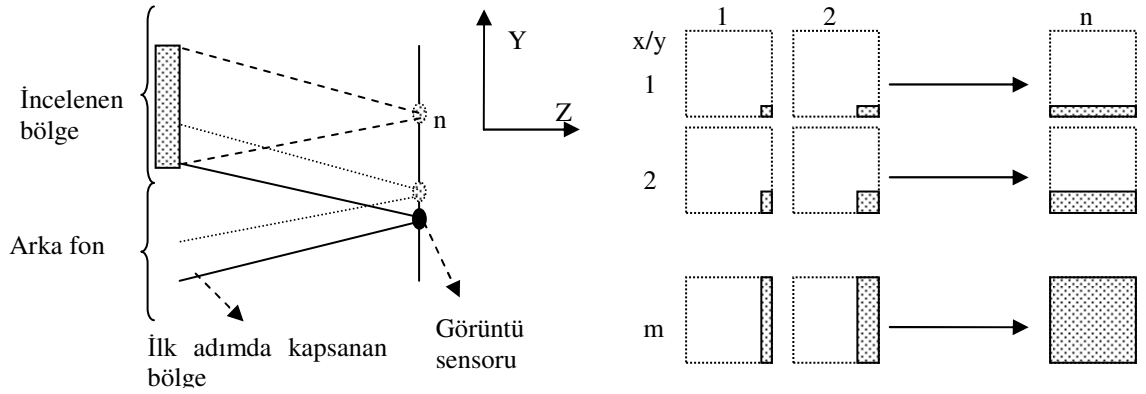
Modelde kullanılan yöntem gerçek görüntüdeki çok küçük bir bölgeyi uygun bir şekilde tarar ve yüksek çözünürlüklü görüntüsünü elde eder. Yöntemde çözünürlüğü arttırılmak istenen bölge kameranın seçilen piksel(ler)i tarafından tamamen kapsanabilmelidir. Dolayısıyla kamerada oldukça belirsiz olarak görünen cisimler bu model ile algılanabilir hale getirilebilmektedir.

Şekil 4.2’de görüldüğü üzere gerçek görüntü üzerindeki bir bölge kamera tarafından bir(kaç) piksel ile yakalanmaktadır. Dolayısıyla bu bölgenin tek bir görüntü karesine bakılarak algılanabilmesi mümkün olamamaktadır. Kamera bir başlangıç noktasından başlayarak incelenecek bölgeyi alt piksel adımlarıyla tarar. Elde edilen düşük çözünürlüklü görüntüler işlenerek piksel altı bilgiler ortaya çıkartılır ve bölgenin çözünürlüğü artırılmış olur.

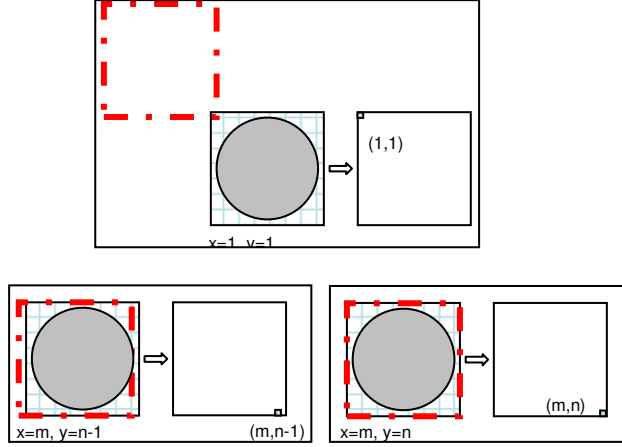


Şekil 4.2: Gerçek görüntünün kamera ile tek piksel olarak alınması

Şekil 4.3 tarama işleminin nasıl gerçekleştirildiğini göstermektedir. Görüntü sensörü S , elde edilmek istenen çözünürlük oranına göre oluşturulan alt piksel ızgarasının aralıkları ölçüsünde önce $\vec{x}$ ve sonra da $\vec{y}$ yönünde adım adım kaydırılmaktadır. Elde edilen piksel değerleri Şekil 4.4’de görüldüğü üzere sonradan işlenmek için bir referans matrisi içinde saklanır. Izgara üzerinde tarama işlemi tamamlandığı zaman bölge piksel tarafından tamamen kapsamaktadır. Burada $\vec{x}$ yatay eksenin normali ve $\vec{z}$ de bölgenin ters yönündeki doğrultu olmak üzere $\vec{y} = \vec{z} \times \vec{x}$ ’dir.



Şekil 4.3: Sensor kaydırmalarına bağılı olarak incelenen bölge üzerinde kapsanan alanlar



Şekil 4.4: Gerçek görüntüden alınan tek piksel değerlerinin referans matrisi üzerine kaydedilmesi

Şekil 4.4’de gösterilen gerçek görüntünün elde etmek istediğimiz yüksek çözünürlüklü görüntüsünün boyu (alt piksel hareket ızgarasının boyutları) $m \times n$ piksel olsun ve kamera pikselinin gerçek görüntü üzerindeki kayma miktarları x ve y ile belirtilsin. Buna göre:

BG : Arka fon rengi

S_{xy} : x - y kaydırmasında elde edilen sensor değeri (referans matrisinin x,y noktasındaki değeri)

P_{xy} : İşlemler sonrası elde edilen yüksek çözünürlüklü görüntünün x - y noktasındaki piksel değeri

$N = m \times n$ olmak üzere

$$S_{11} = BG + \frac{P_{11} - BG}{N}$$

$$S_{12} = BG + \frac{P_{11} - BG}{N} + \frac{P_{12} - BG}{N}$$

$$S_{1n} = BG + \frac{P_{11} - BG}{N} + \frac{P_{12} - BG}{N} + \dots + \frac{P_{1n} - BG}{N}$$

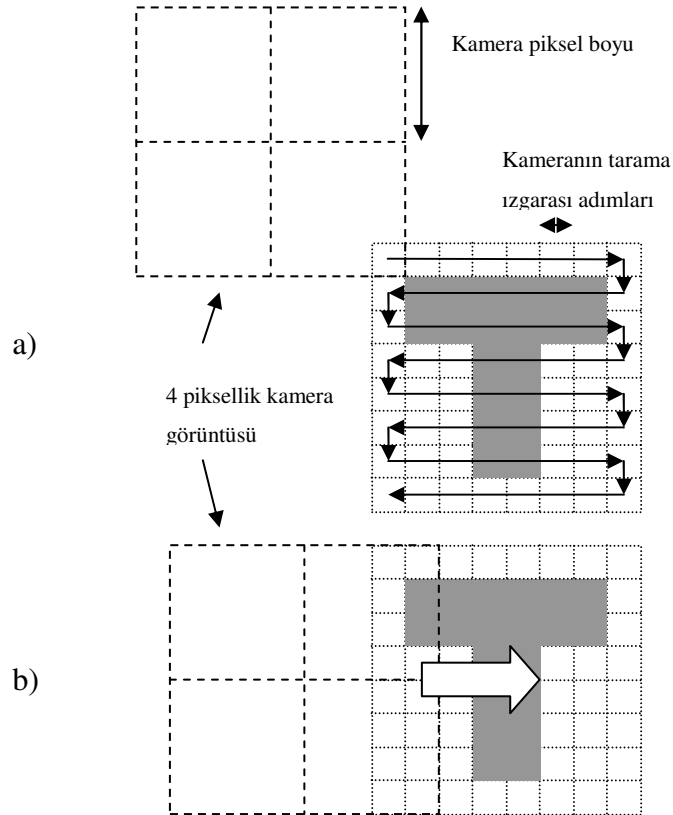
$$S_{mn} = BG + \frac{P_{11} - BG}{N} + \frac{P_{12} - BG}{N} + \dots + \frac{P_{1n} - BG}{N} + \dots + \frac{P_{mn} - BG}{N}$$

$$S_{xy} = BG + \sum_{i=1}^x \sum_{j=1}^y \frac{(P_{ij} - BG)}{N} \quad (4.6)$$

Eğer kaydıma miktarlarındaki tüm sensor değerleri bilinirse, alt piksel ızgarasındaki tüm piksel değerleri Denklem 4.6'daki döngüden hesaplanabilir.

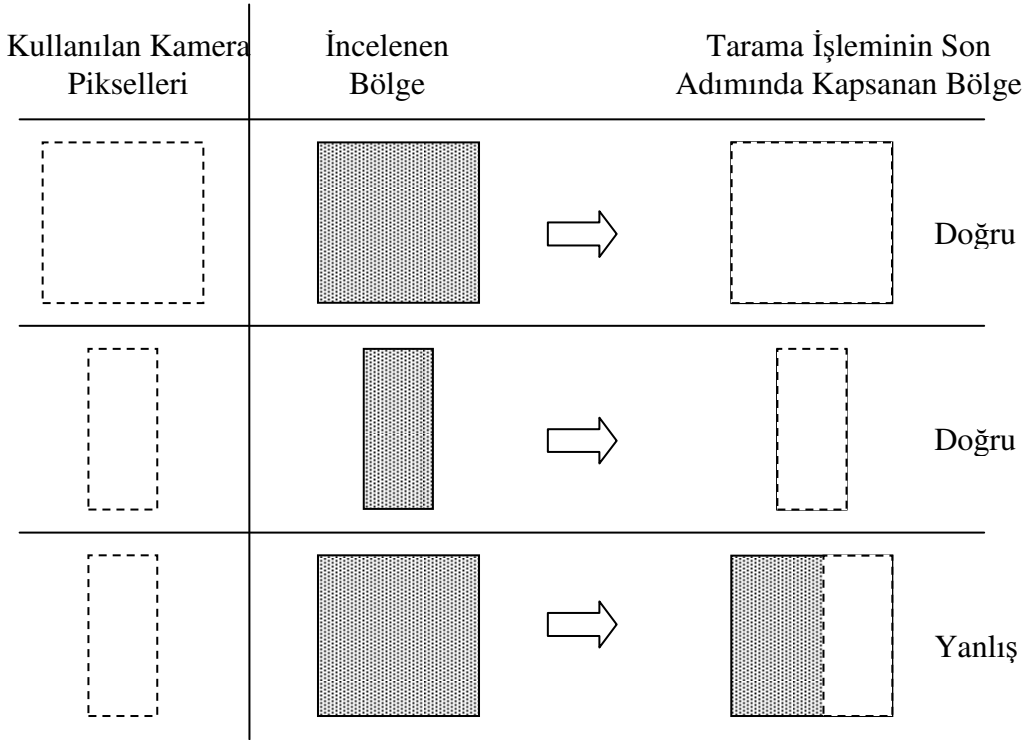
5. SİMÜLASYON ÇALIŞMASI

Simülasyon çalışmasında sabit bir görüntü üzerinde algoritmaya uygun olarak gerçekleştirilen kamera tarama hareketleri modellenmiştir ve elde edilen verilerden yüksek çözünürlüklü görüntü oluşturulmuştur. Simülasyon iki bölümden oluşmaktadır. İlk bölümde, seçilen görüntü ve belirlenen tarama ızgarasına göre adım hareketleri belirlenmektedir ve kameranın bu adımlara ilişkin nasıl görüntüler yakalayacağı hesaplanmaktadır. Seçilen yöntemle göre (Şekil 5.1’de görülen kare şeklinde tarama ve tek yönde tarama) her adımda alınan görüntü sabit diske kaydedilmekte ve simülasyonun ikinci adımı başlatılmaktadır. İkinci adımda ilk adımda kullanılan bilgiler eşliğinde diskteki görüntüler işlenmekte ve bir tane yüksek çözünürlüklü görüntü elde edilmektedir.



Şekil 5.1: Kameranın görüntüyü tarama biçimleri. a) Kare tarama; b) Tek yönde tarama

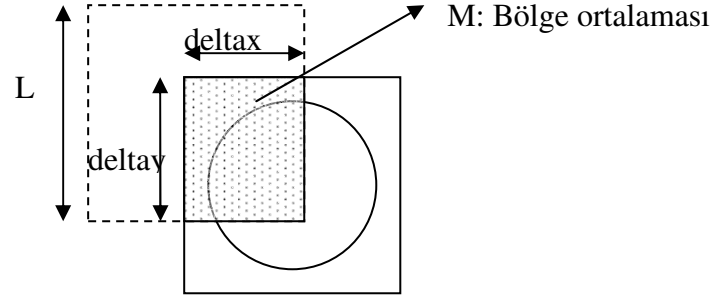
Simülasyonlarda, tarama için seçilen kamera pikselleri incelenen bölgeyle uygun olmalıdır. Yöntemde tarama işleminin son adımında, kamera pikselleri incelenen bölgeyi tamamiyla kaplaması gerekmektedir. Şekil 5.2’de farklı durumlar için kullanılan kamera pikselleri ve incelenen bölge ilişkileri görülebilmektedir. Ayrıca yöntemin çalışabilmesi için incelenen bölgenin çevresinin sabit bir arka fon rengine sahip olması gerekmektedir.



Şekil 5.2: Kullanılan kamera pikselleri ve incelenen bölge ilişkileri

5.1 Simülasyonun 1. Adımı

Bu adımda kamera tarama hareketlerine göre yakalanacak görüntüler modellenmiştir. Görüntünün yakalanması işlemi gerçek görüntünün belli bir adım miktarında kaydırılması ve düşük çözünürlüklü hale getirilmesini kapsamaktadır. Şekil 5.3’de arka fon rengi BG olmak üzere görüntü üzerinde $deltax$ ve $deltay$ kaydırma miktarlarına ilişkin kamera sensorundan yakalanan görüntü şöyle hesaplanmaktadır

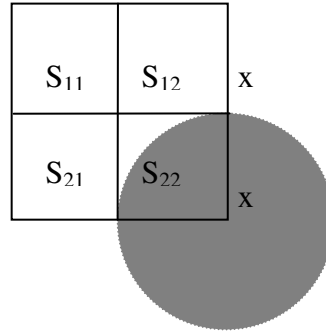


Şekil 5.3: Sensorun görüntü üzerinde *deltax* ve *deltay* kaydırılması

S sensor değeri olmak üzere;

$$S = BG + \frac{(M - BG) * deltax * deltay}{L^2} \quad (5.1)$$

Örneğin arka fon grilik seviyesi 255 olan bir bölgede 4 pikseli bir kamera tarama sırasında içi 100 grilik değerinde olan bir dairenin içine aşağıdaki oranda girerse, kamera sensorlarının alacağı değerler şöyle olmaktadır;



Şekil 5.4: 4 pikseli bir kameranın daire şeklinde bir bölgeyi kaplaması

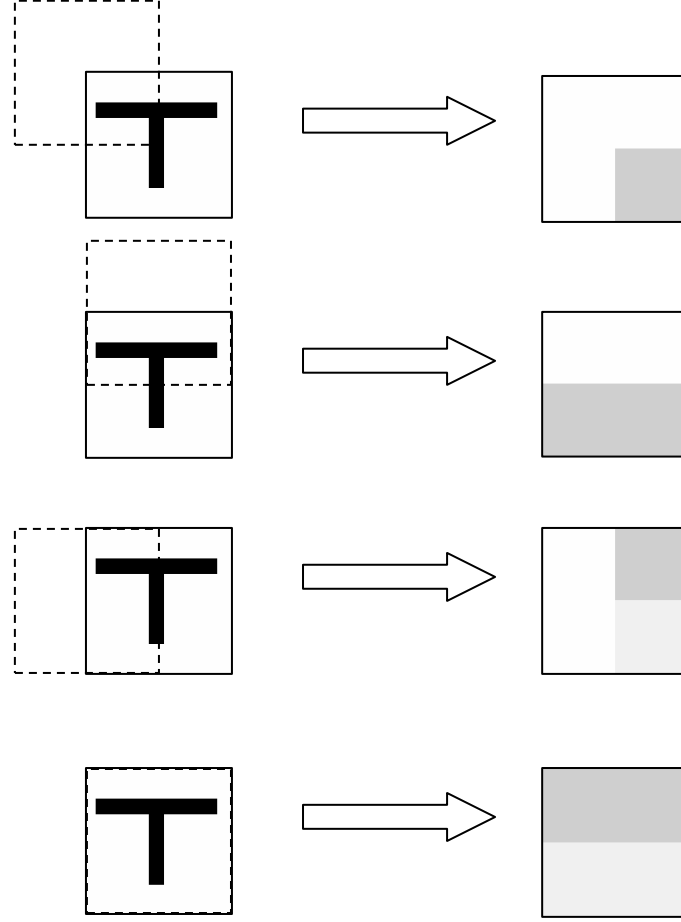
$$S_{11} = 255 + \frac{(0 - 255) * 0 * 0}{x^2} = 255$$

$$S_{12} = 255 + \frac{(0 - 255) * 0 * 0}{x^2} = 255$$

$$S_{21} = 255 + \frac{(0 - 255) * 0 * 0}{x^2} = 255$$

$$S_{22} = 255 + \frac{\left(255 - \frac{155\pi}{4}\right) - 255}{x^2} * x * x = 133.2$$

Anlatılan örnek gerçek bir görüntüyü uyarlandığında alınan sonuçlar Şekil 5.5’de görülmektedir.



Şekil 5.5: 4 pikseli bir kameranın gerçek bir görüntüyü 2x2 adımda tarama simülasyonu

5.2 Simülasyonun 2. Adımı

İlk adımda kamera hareketleri modellenerek yakalanan ve sabit diske kaydedilen düşük çözünürlüklü görüntüler bu adımda işlenerek bölgenin yüksek çözünürlüklü görüntüsü elde edilmeye çalışılmaktadır.

5.2.1 Tek Yönde Tarama

Eğer bölge tek yönde tarandıysa sonuçta çıkacak yüksek çözünürlüklü görüntüde de tek yönde iyileştirme beklenmektedir. Arka fon rengi BG , tek yöndeki adım sayısı L , ilerlenen yöndeki kayma miktarına ilişkin düşük çözünürlüklü sensordan elde edilen değer S_x olmak üzere ilerlenen yöndeki kayma miktarına ilişkin yüksek çözünürlüklü piksel değeri P_x Denklem 5.2'deki döngüden elde edilebilmektedir. Bu işlem döngünün her satırı için ayrı ayrı uygulanmalıdır.

$$S_x = BG + \sum_{i=1}^x \frac{(P_i - BG)}{L} \quad (5.2)$$

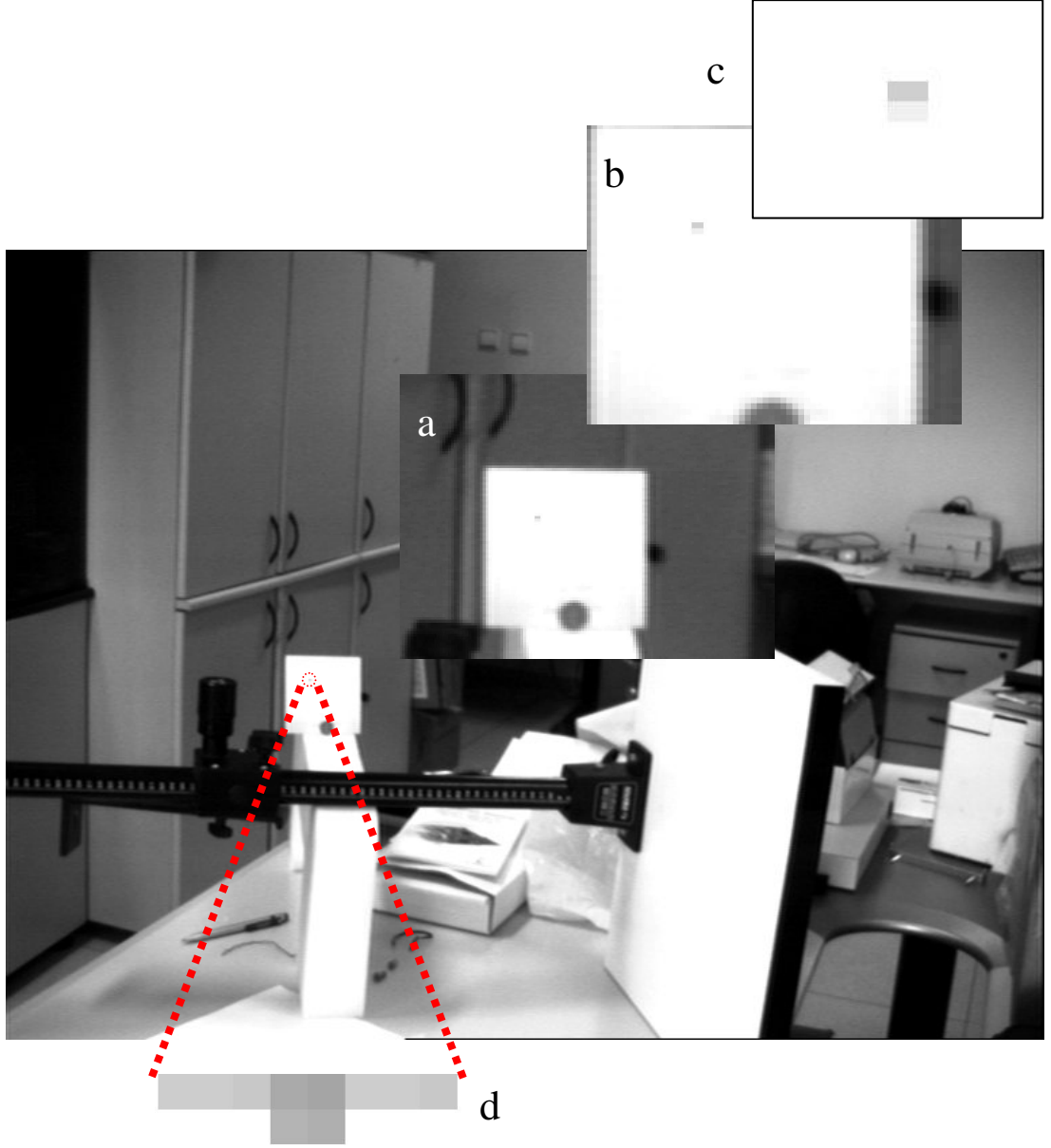
Tek yönde tarama çok karmaşık olmayan görüntülerde (örneğin siyah-beyaz yazı görüntülerinde harflerin daha iyi algılanmasını sağlayarak OCR uygulamalarının başarımını arttırmak için) kısa sürede tek boyutta çözünürlük artışı sağlamak için kullanılır. Örneğin Şekil 5.6'daki görüntüde kâğıt üzerindeki 2x2 boyutundaki görüntü yatay ekseninde 10 adım kaydırılarak çözünürlük iyileştirmesi sağlanmıştır.

Tek yönde tarama simülasyonu için yapılan deneme sonuçları Tablo 5.1'de görülmektedir. Burada dikkat edilmesi gereken nokta karmaşık olmayan şekiller için kare taramasındaki kadar çok görüntü yakalamadan da çözünürlük iyileştirmesinin yapılabileceğinin gözlemlenmesidir.

5.2.2 Kare Şeklinde Tarama

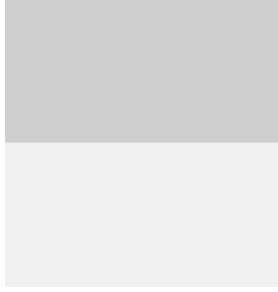
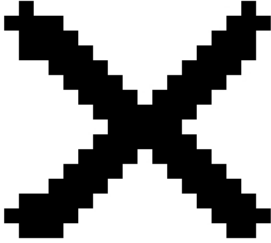
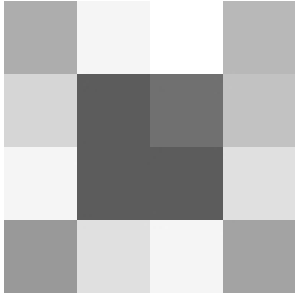
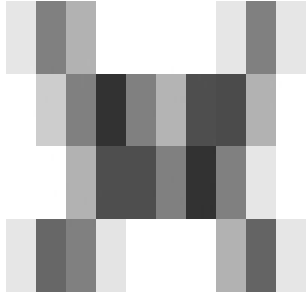
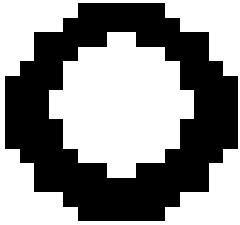
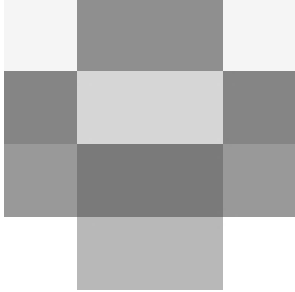
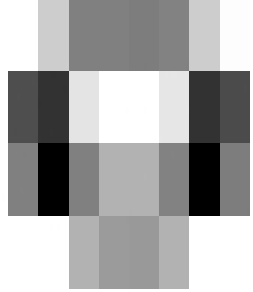
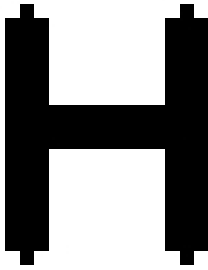
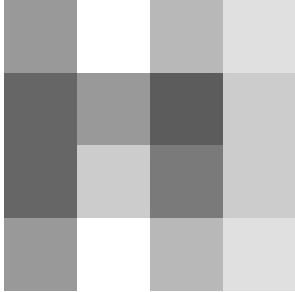
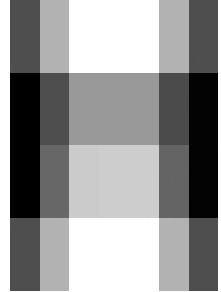
Eğer bölge kare şeklinde tarandıysa; arka fon rengi BG , oluşturulan yüksek çözünürlüklü görüntünün piksel sayısı N ve x-y kayma miktarına ilişkin düşük çözünürlüklü sensordan elde edilen değer S_{xy} olmak üzere x-y kayma miktarına ilişkin yüksek çözünürlüklü piksel değeri P_{xy} Denklem 5.3'deki döngüden elde edilebilmektedir.

$$S_{xy} = BG + \sum_{i=1}^x \sum_{j=1}^y \frac{(P_{ij} - BG)}{N} \quad (5.3)$$



Şekil 5.6: Kâğıt üzerindeki şeklin tek yönde tarama ile çözünürlüğünün iyileştirilmesi simülasyonu; a,b,c) Kamera ile yakalanan görüntüye farklı büyütme oranları uygulanması; d) Tarama işlemi sonrasında şeklin çözünürlüğünün iyileştirilmesi

Tablo 5.1: Farklı görüntü modelleri için tek yönde tarama sonucunda elde edilen çözünürlük iyileştirmeleri

Gerçek Görüntü	Kamera Görüntüsü	Çözünürlük İyileştirme Sonucu
 20x20	 2x2	 10x2
 20x20	 4x4	 10x4
 20x20	 4x4	 10x4
 20x20	 4x4	 10x4

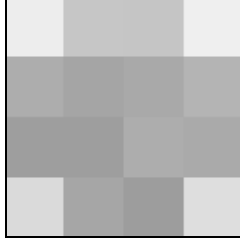
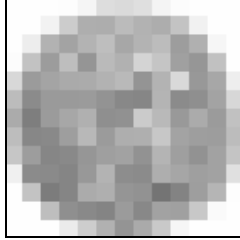
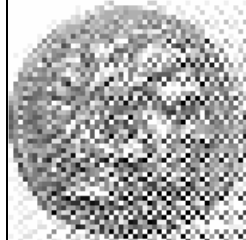
Örnek 1:

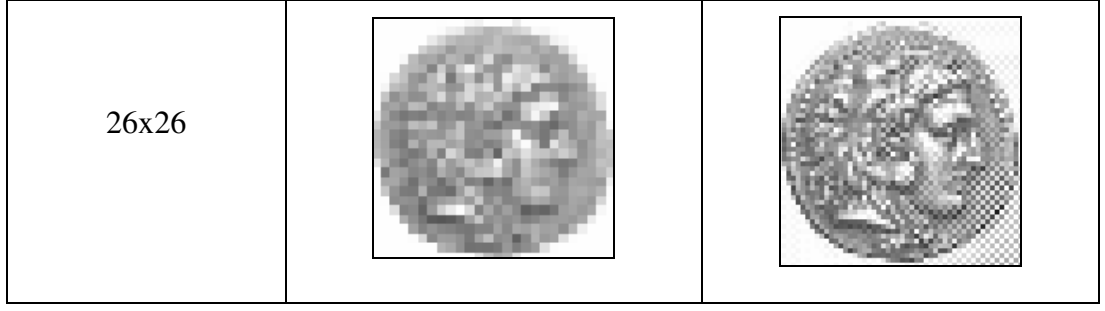
Kare şeklinde tarama simülasyonu için ilk örnek olarak Şekil 5.7’de görünen 52x52 piksellik resim kullanıldı. İlk denemede 4x4 çözünürlüklü bir kamera, ikinci denemede 13x13 piksellik bir kamera ve üçüncü denemede ise 26x26 çözünürlüklü bir kamera kullanarak orijinal görüntü 52 adımda tarandı. Her tüm denemeler sonucunda da elde edilen 52x52 piksellik yüksek çözünürlüklü görüntüler Tablo 5.2’de görülmektedir.



Şekil 5.7: Kare şeklinde tarama için kullanılacak 52x52 piksel boyutunda orijinal görüntü

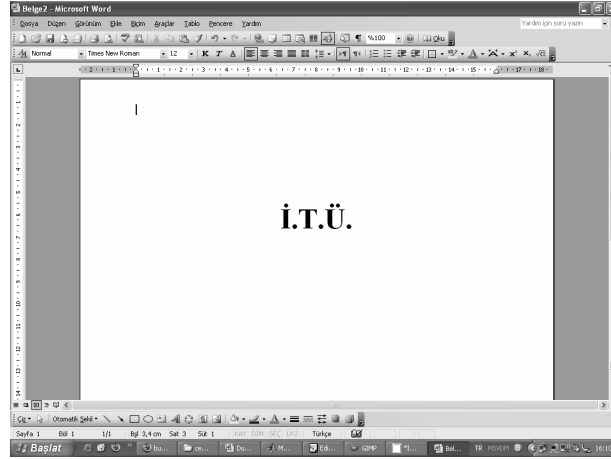
Tablo 5.2: Kare şeklinde 52 adımlık kaydırma sonucunda kamera çözünürlüğüne bağlı olarak elde edilen görüntüler

Kullanılan Kamera Çözünürlüğü	Kamera Görüntüsü	Yüksek Çözünürlüklü Görüntü
4x4		
13x13		



Örnek 2:

Şekil 5.8 bir uygulama ekranının gerçek görüntüsünü temsil etsin. Şekil 5.9 ise aynı ekranın uzaktaki gri-ölçekli bir kamera ile yakalanmış düşük çözünürlüklü görüntüsü olsun.

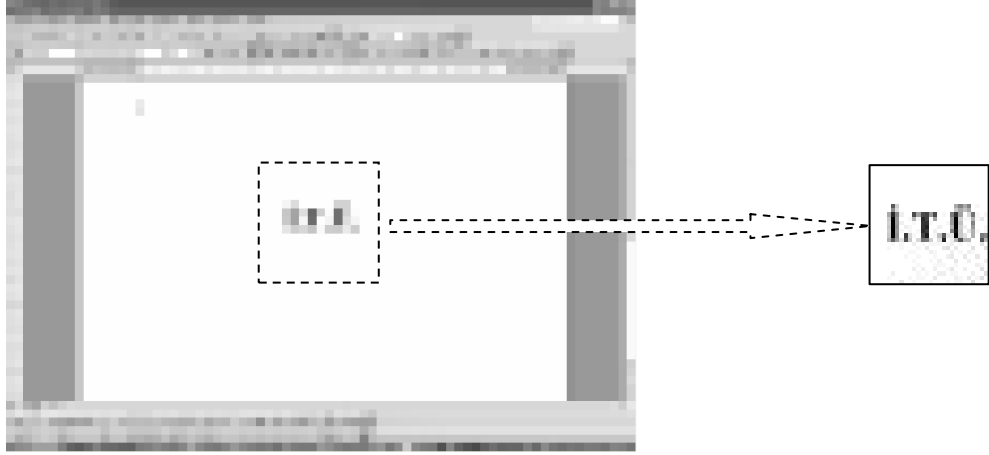


Şekil 5.8: Ekranın gerçek görüntüsü



Şekil 5.9: Ekranın kamera ile yakalanmış görüntüsü

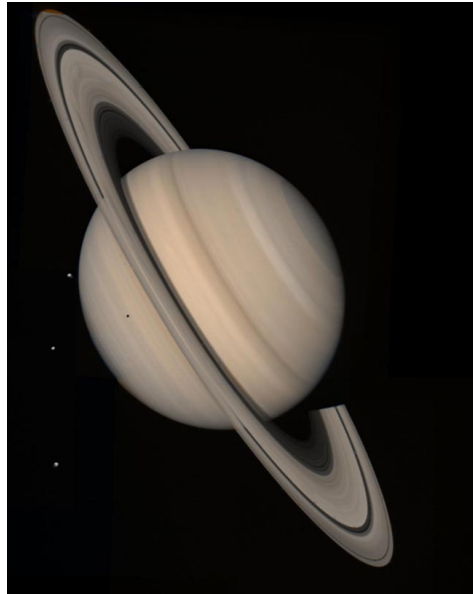
Düşük çözünürlüklü kamera ile tarama işlemi sonunda yakalanan görüntüler önerilen yöntem ile işlendiğinde Şekil 5.10'daki sonuç elde edilmektedir. Şekilden de görüldüğü gibi görüntü içindeki belli bir noktaya odaklanıp, sadece o bölgenin yüksek çözünürlüklü görüntüsü elde edilmeye çalışılmıştır.



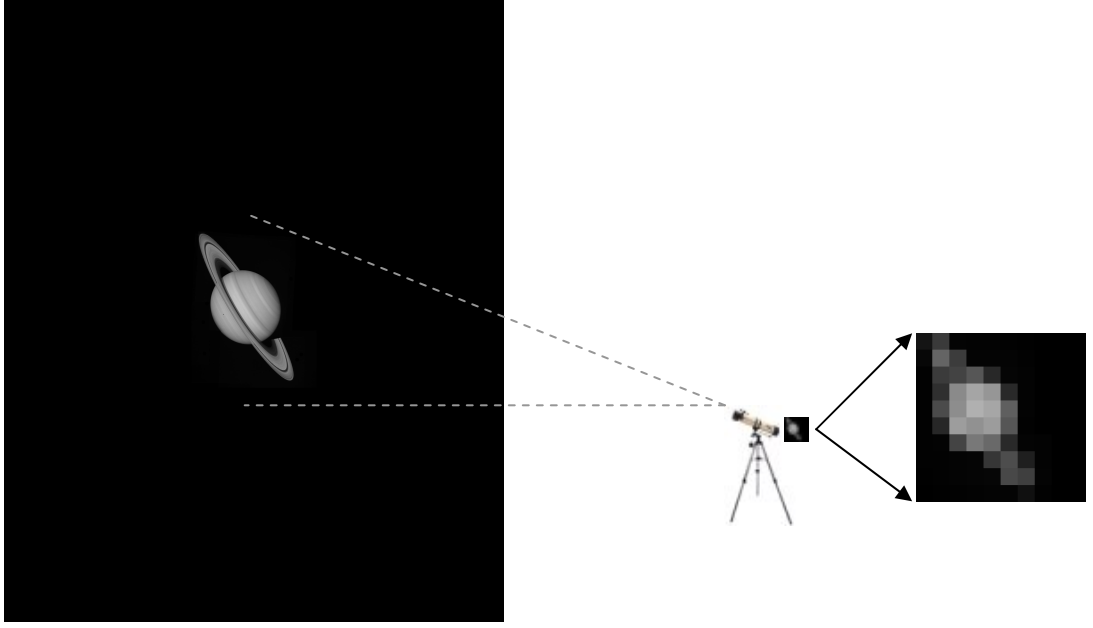
Şekil 5.10: Ekranın yüksek çözünürlüklü görüntüsünün elde edilmesi

Örnek 3:

Şekil 5.11 Satürn gezegeninin gerçek görüntüsünü temsil etsin. Gezegeni teleskopla bakılıp, elde edilen görüntü dijital bir kamera ile yakalanırsa Şekil 5.12'deki görüntü elde edilmektedir.

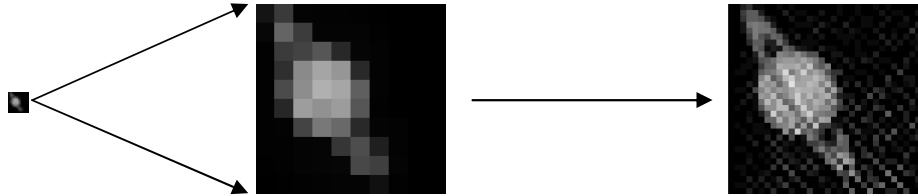


Şekil 5.11: Satürn gezegeninin gerçek görüntüsü



Şekil 5.12: Satürn gezegeninin teleskop ve kamera ile yakalanmış görüntüsü

Şekil 5.12'den de görüleceği üzere teleskopun büyütme limitlerinin olması ve kamera çözünürlüğünün düşük olmasından dolayı algılanabilmekten çok uzak bir görüntü yakalanmıştır. Önerilen yöntemdeki tarama işlemi sonucunda elde edilen düşük çözünürlüklü görüntüler işlendiğinde Şekil 5.13'deki yüksek çözünürlüklü görüntü elde edilmektedir.



Şekil 5.13: 32x32 adımda görüntünün taranması sonucunda elde edilen çözünürlük iyileştirmesi

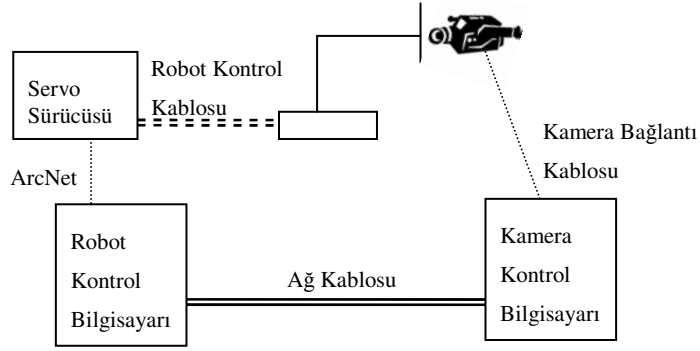
6. DENEYSEL ÇALIŞMA

Deneysel çalışmada mercekli göz yapısının dışında bileşik gözlere sahip böcek gözü modeli laboratuvar koşullarında gerçekleştirilmiştir. Şekil 3.2’de görüldüğü üzere böcek gözleri birçok küçük gözcükten (bazı böcek türlerinde bu sayı 30000 olabilmektedir) oluşmaktadır ve her gözcük bakılan doğrultudaki bölgenin bir bölümünün verisini yakalayabilmektedir. Yakalanan veriler beyinde birleştirilerek tek bir görüntü elde edilmektedir. Çalışmada bileşik gözü modellerken her gözcüğün bulunması gereken yerde 1 tane görüntü sensörü kullanma imkânı (belli aralıklarda konumlanmış yüzlerce görüntü sensörü yerleştirmek) olmadığı için kamerayı veya incelenecek bölgeyi hareket ettirerek model gerçekleştirilmiştir.

Çalışmada gerçekleştirilmek istenen model bileşik göz yapılarına benzetmekle birlikte gözcüklerin incelenen bölge üzerinde kapsadığı görüntü alanları bakımından farklılık göstermektedir. Şekil 3.8’de görüldüğü üzere her gözcük bir önceki gözcüğün kapsadığı bölgeye yeni veriler eklemektedir. Farklı kapsam bölgeleri olan gözcükleri modelleyebilmek için artımlı bir robot koluna bağlanmış dijital kamera bir düzlem üzerinde hareket ettirilmektedir ve gözcüklerin konumuna geldiği zaman görüntü yakalayıp saklamaktadır. Tüm görüntüler alındıktan sonra gereken işlemler yapılarak incelenen bölgenin yüksek çözünürlüklü görüntüsü elde edilmektedir.

6.1 Çalışmanın Mimarisi

Yapılan çalışmada Şekil 6.1’de görüldüğü üzere 2 adet PC, 1 adet servo sürücü, 1 adet Mitsubishi Pa10-7C robotu, 1 adet ISA ArcNet iletişim kartı, 1 adet Euresys Framegrabber kartı ve 1 adet CCD kamera kullanılmıştır. Bilgisayarlar yerel ağ kablosuyla birbirlerine bağlıdır. Tablo 6.1 ve Tablo 6.2’de kullanılan bilgisayarlarda kurulu donanım ve yazılım bilgileri görülmektedir.



Şekil 6.1: Çalışmanın mimarisi

Tablo 6.1: Robot kontrol bilgisayarı donanım ve yazılım listesi

Donanım/Yazılım Adı	İşlevi
PIII 800Mhz	İşlemci
ISA ArcNet İletişim Kartı	Servo sürücü ile iletişim
RedHat 8.0 Linux	İşletim sistemi
GSL (GNU Scientific Library)	Hızlı matris ve vektör işlemleri
Pa10 Linux Sürücüsü	C fonksiyonlarıyla robotu sürmek
MatrixLibrary	GSL kullanılarak yazılmış matris ve vektör sınıfları
InverseKinematicLibrary	Robot eklemlerinin hız kontrolünde kullanılan ters kinematik işlemleri
RobotLibrary	Üst düzey robot kontrol işlemleri
StringLibrary	Genel karakter katarı işlemleri
SocketLibrary	Temel TCP/IP iletişim işlemleri
CameraSocketLibrary	Kameranın TCP/IP üzerinden kontrolü
TimerLibrary	1 micro sn duyarlı zamanlayıcı kontrolü

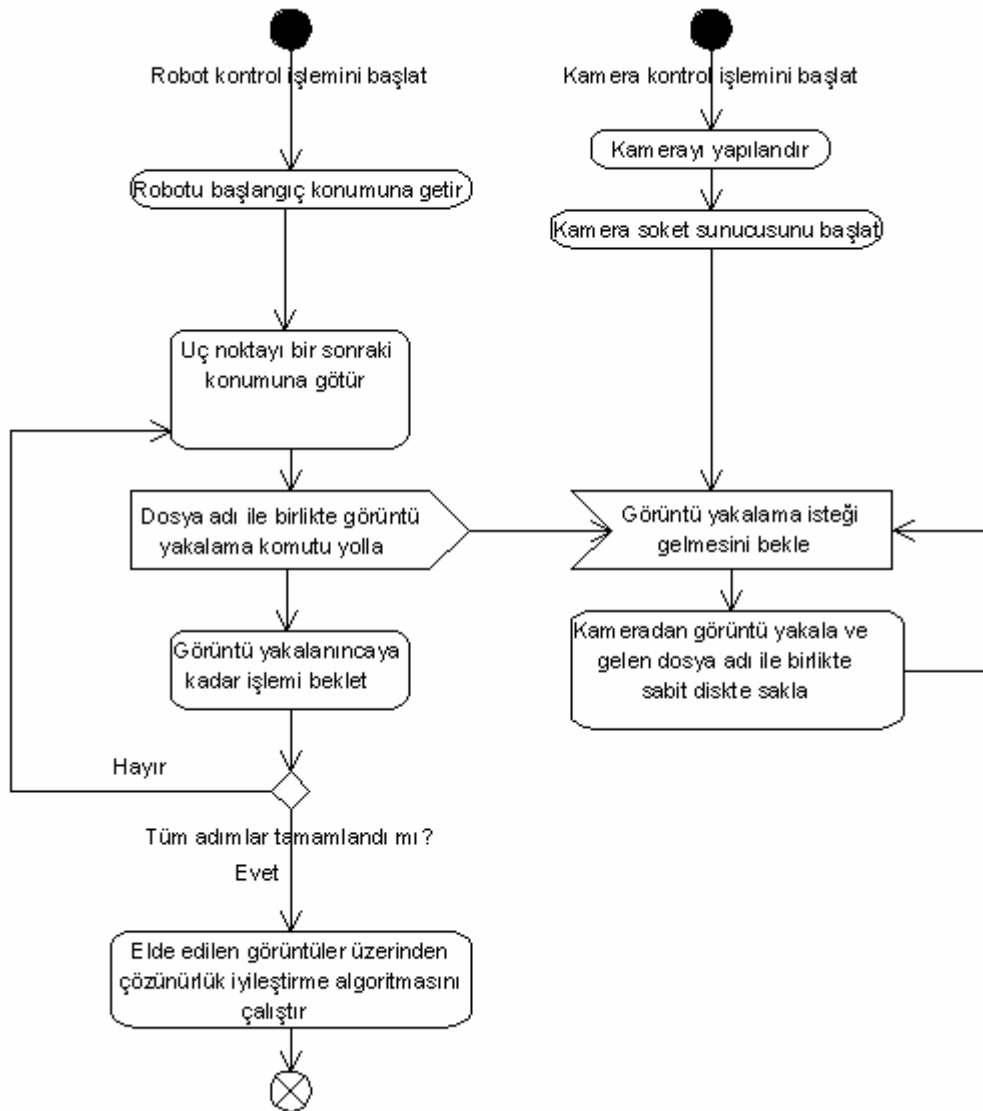
Tablo 6.2: Kamera kontrol bilgisayar donanım ve yazılım listesi

Donanım/Yazılım Adı	İşlevi
PIV 2.4Ghz	İşlemci
Euresys Framegrabber	Kamerayı sürmek
RedHat 8.0 Linux	İşletim sistemi
MultiCam	C fonksiyonlarıyla framegrabber kartını sürmek
OpenCV (Open Computer Vision)	Hızlı görüntü işleme işlemleri
FrameGrabberLibrary	Framegrabber kartının uygulamaya özel üst düzey kontrolü
ImagingLibrary	OpenCV kullanarak temel görüntü işlemleri
StringLibrary	Genel karakter katarı işlemleri
SocketLibrary	Temel TCP/IP iletişim işlemleri
CameraSocketLibrary	Kameranın TCP/IP üzerinden kontrolü
TimerLibrary	1 micro sn duyarlı zamanlayıcı kontrolü
SuperResolution	Yüksek çözünürlüklü görüntüleme modülü

6.2 Çalışmanın Akış Diyagramı

Şekil 6.2’den görüldüğü üzere sistemin tüm kontrolü robot kontrol bilgisayarındadır. İşlem başlatılmadan önce kamera kontrol bilgisayarındaki kamera kontrol soket programı çalıştırılmalıdır. Bu program ile TCP/IP üzerinden uzaktaki bilgisayara bağlı dijital kameraya dosya adı ile birlikte görüntü yakala komutu yollanabilmektedir. Kamera kontrol bilgisayarındaki uygulama başlatıldığı zaman

öncelikle resim sayısallaştırıcı yapılandırmaktadır, böylece soket üzerinden istek geldiğinde istenen bölgeyle ilgili görüntüler yakalanmakta ve sabit diske kaydedilmektedir. Robot kontrol bilgisayarındaki uygulama başlatıldığı zaman robotun uç noktası öncelikle başlangıç konumuna getirilmektedir ve daha önceden belirlenen bir tarama ızgarası üzerinde algoritmaya uygun kaydırma oranlarında hareket etmekte ve her kaydırma adımı için kamera kontrol bilgisayarına görüntü yakalama komutu yollamaktadır. Tüm tarama işlemi tamamlandıktan sonra çözünürlük iyileştirme algoritması çalıştırılarak incelenen bölgenin yüksek çözünürlüklü görüntüsü elde edilmektedir.



Şekil 6.2: Çalışmanın akış diyagramı

6.3 Robot Kontrolü Çalışması

6.3.1 Çalışmanın Amacı

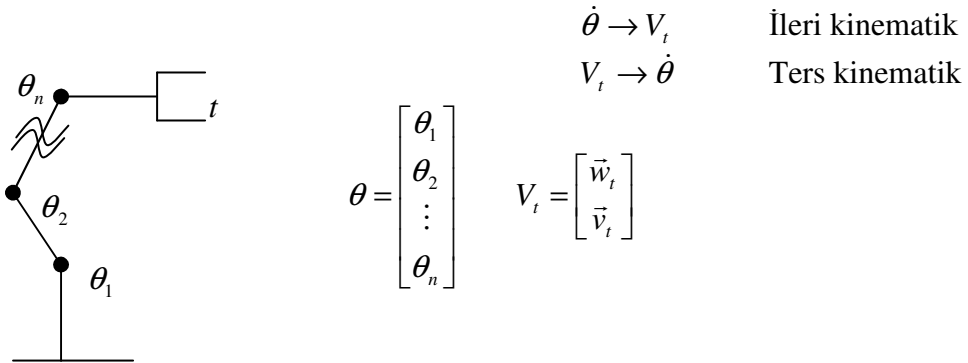
Robotik çalışmasını yapmaktaki amaç, böcek gözünün modellenmesi sırasında bileşik göz yapılarındaki farklı bölgelere yönelmiş her bir gözcüğün aldığı görüntü verisini yakalamaktır. Bunu gerçekleyebilmek için robotun ucuna bağlanan bir dijital kameranın veya incelenecek görüntünün uzaydaki bir düzlem üzerinde tarama ızgarasındaki adımlara bağlı olarak hareket ettirilmesini sağlamak gerekmektedir. Tarama ızgarasındaki adım aralıkları çok küçük (yaklaşık 0.1mm) olacağından sistemin hassasiyetinin oldukça yüksek olması sağlanmalıdır.

Bu bölümde simülasyon çalışmaları Matlab ortamında gerçekleştirilmiştir ve deneysel çalışmalar için PA10-7C robot kolu kullanılmıştır.

6.3.2 Robot Kolunun Bir Düzlem Üzerinde Hareket Ettirilmesi

İleri kinematik denklemleri eklem noktalarındaki açı bilgilerinden uç noktanın konumunu ve dönüşünü hesaplamaktadırlar. Ters kinematik denklemleri ise ileri kinematiktekilerin tersi olarak uç noktanın bulunması istenen konum ve yöneliminden açı değerlerini hesaplamaktadır. Artımlı manipülatörlerde ters kinematik problemleri için sonsuz çözüm vardır. Bu problemlerde belli optimizasyon kriterlerine göre çözüm aranır.

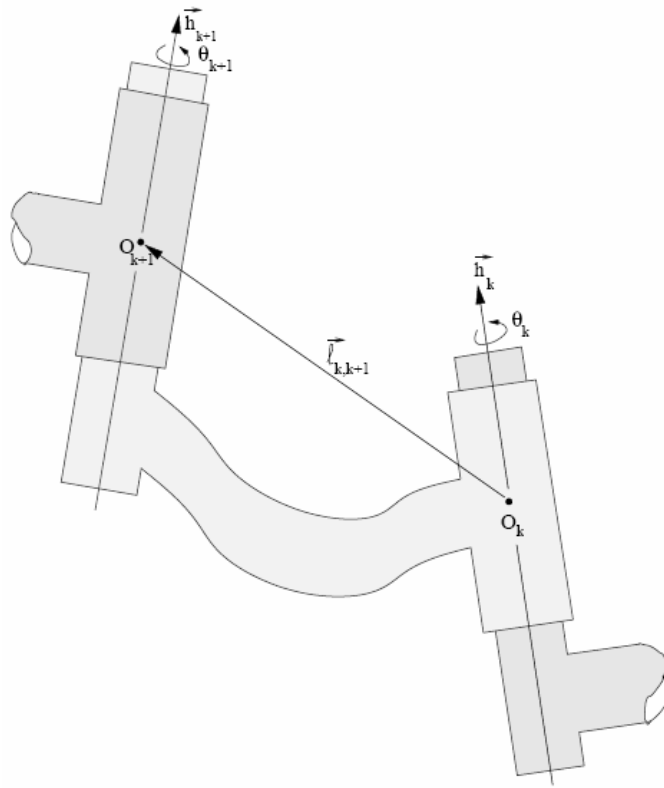
Şekil 6.3’de görüldüğü gibi t uç nokta, θ eklem açıları ve V_t uç nokta hız vektörü olmak üzere



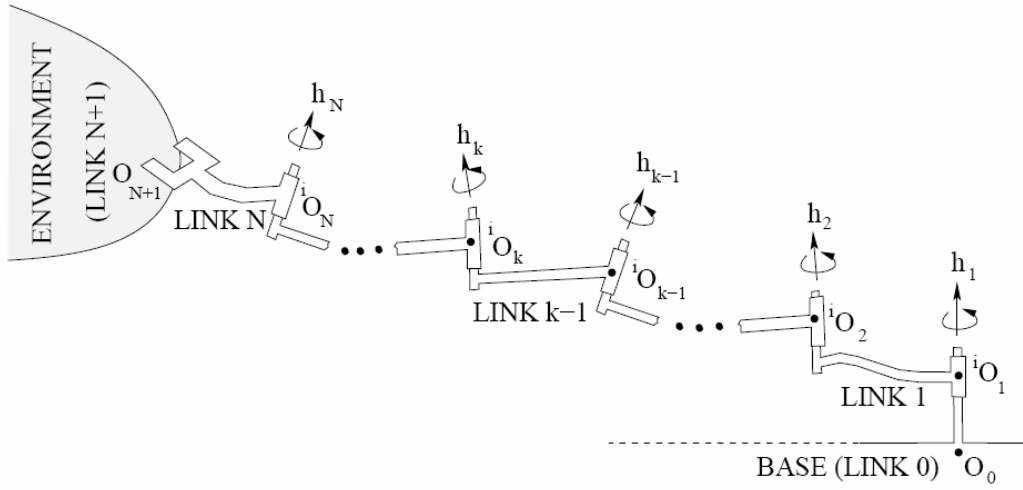
Şekil 6.3: Robot kolu eklem açıları ve uç nokta hız vektörü

6.3.3 Seri Manipülator için Ters Kinematik Algoritması

Burada anlatılan algoritma vektörel gösterimlerden yararlanmaktadır. Gövde koordinat eksen takımı O ile gösterilmiştir. Bu değişkenin altında yazan indis onun hangi parçaya ait olduğunu belirtmektedir. Örneğin O_3 üçüncü linkten sonra gelen gövde eksen takımıdır. h_k k . eklemin dönme eksenine paralel birim vektördür. Buradaki tanımlamalar genel kurallar göz önünde bulundurularak yapılmıştır. Parçalar arasındaki her link bir DOF (serbestlik derecesi) ekleme sahiptir. Eğer birden çok DOF eklem olması durumunda varsayımdaki h_r 'ye kolon başına birer birim vektör daha eklemek yeterli olacaktır. H_k ve θ_k k . eklemdaki uzamsal hareket eksenini ve o eklem için eklem açısıdır. Şekil 6.4'de görüldüğü üzere $l_{k,k+1}$ O_k 'den O_{k+1} 'e doğru bir vektördür. Lineer ve açısal hız vektörleri de v_k ve w_k ile gösterilmektedir. Uç noktanın hız vektörü V_k ile ifade edilmektedir ve (w_k^T, v_k^T) şeklinde tanımlanmıştır. Burada T işlemi transpoze operatörüdür. Son olarak vektör çarpımının ters bakışlı (skew-symmetric) lineer operatörü şapka ile gösterilmiştir, örneğin $\hat{p} = \bar{p} \times$ olmak üzere ifade edilmiştir.



Şekil 6.4: k linkine ilişkin vektörler



Şekil 6.5: N linkli seri manipülatör

Bu bölümde hareket denklemleri seri olarak bağlanmış bir manipülatör için türetilenlerdir. Taban eksen takımına göre manipülatörün açısal ve lineer hızları $k-1$. linkten k . linke aktarılmaktadır. k . linke ait açısal ve lineer hızlar Denklem 6.1'den elde edilmektedir.

$$V_k = \phi_{k,k-1} V_{k-1} + H_k \dot{\theta}_k \quad (6.1)$$

Denklem 6.1'de

$$\phi_{k,k-1} = \begin{bmatrix} {}^3I & {}^3O \\ -\hat{l}_{k-1,k} & {}^3I \end{bmatrix}$$

$\phi_{k,k-1}$, O_{k-1} 'den O_k 'ya hızları aktaran lineer bir operatördür ve “propagasyon operatörü” olarak adlandırılır. Hız vektörü, propagasyon vektörü ve eklem açısı matrisi Denklem 6.2'de görüldüğü gibi hesaplanabilir.

$$V = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \quad V \in \Re^{6n \times 1} \quad (6.2a)$$

$$\phi = \begin{bmatrix} {}_6I & {}_6O & \cdots & {}_6O \\ \phi_{2,1} & {}_6I & \cdots & {}_6O \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n,1} & \phi_{n,2} & \cdots & {}_6I \end{bmatrix} \quad \phi \in \Re^{6n \times 6n} \quad (6.2b)$$

$$H = \begin{bmatrix} H_1 & & & 0 \\ & H_2 & & \\ & & \ddots & \\ 0 & & & H_n \end{bmatrix} \quad H \in \Re^{6n \times 6n} \quad (6.2c)$$

Denklemlerdeki n değişkeni robotun serbestlik derecesi sayısıdır. Bu değişkenlere göre link noktalarının hızları Denklem 6.3'den elde edilebilir.

$$V = \phi H \dot{\theta} \quad (6.3)$$

$\sigma_i = [{}_6O \quad \cdots \quad \phi_{n,n-1}]$ uç propagasyon operatörü her iki tarafın da soldan Jakobyen ile çarpılmasıyla elde edilmektedir.

$$V_i = J \dot{\theta} \quad (6.4)$$

Denklem 6.4'de

$$J = \sigma_i^{\Delta} \phi H \quad J \in \Re^{6 \times n} \text{ olmaktadır.}$$

Şu ana kadar Jakobyenin türetilmesinin genel formülü gösterilmiştir. Bu metot her tipten seri manipülatöre uygulanabilir. Manipülatörün artımlı olması için n sayısının 6'dan büyük olması gerekmektedir. Dolayısıyla test kinematik problemi Denklem 6.5'in en küçük kareler yöntemiyle çözülmesine indirgenebilir

$$\dot{\theta} = J^{\#} V_i \quad (6.5)$$

Denklem 6.5'de

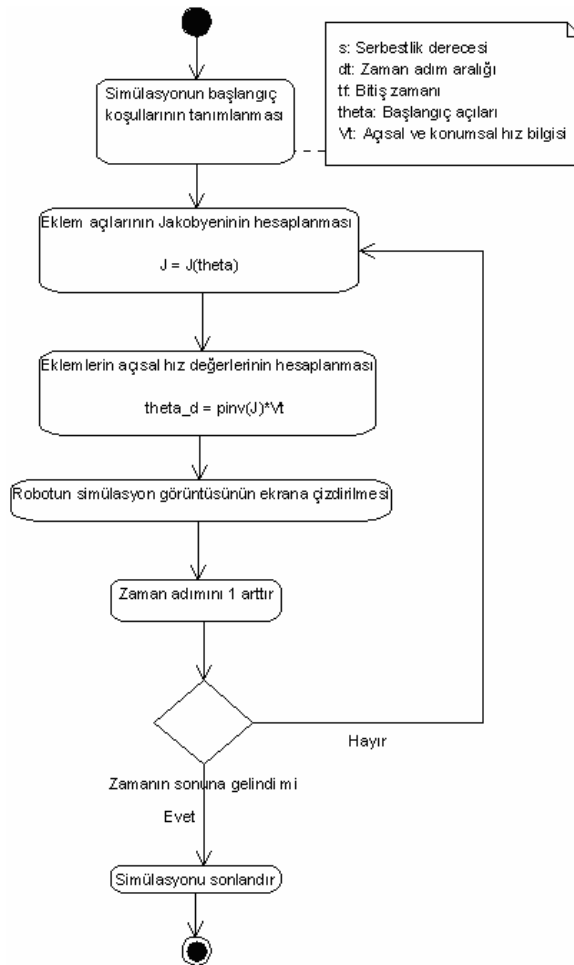
$$J^{\#} = J^T (JJ^T)^{-1} \text{ olmaktadır.}$$

Metodun çalışabilmesi için robotun tekil bir konumda olmaması gerekmektedir. Bu kısıt robotun tarayacağı alan içerisinde herhangi bir tekil bölge olmamasının garanti edilmesiyle aşılabilmektedir. Dolayısıyla robotun ilk başlayacağı konum işlemin devamı açısından önem taşımaktadır.

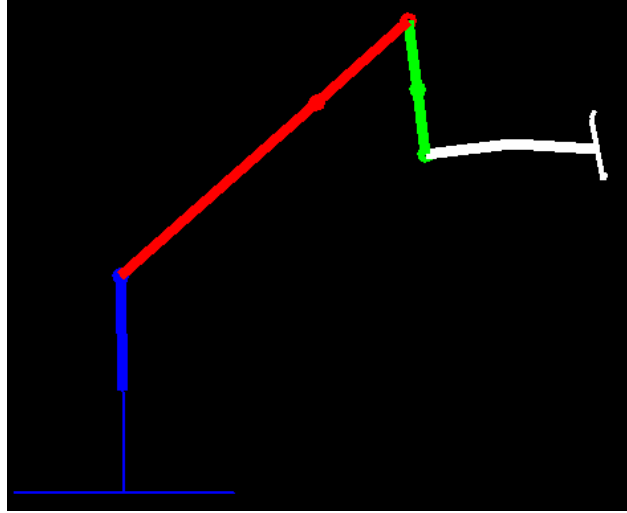
6.3.4 Gerçeklenen Çalışma

6.3.4.1 Simülasyon Çalışması

Robotun uç noktasının bir düzlem üzerinde hareket ettirilmesi robotik kontrol için bir ters kinematik problemidir. Bu sebeple önceki bölümlerde teorisi açıklanan ters kinematik yönteminin simülasyon çalışması Matlab kullanılarak yapılmıştır. Çalışmanın akış diyagramı Şekil 6.6'da görülmektedir. Programın kodları ek olarak verilmiştir.



Şekil 6.6: Ters kinematik çalışmasının akış diyagramı



Şekil 6.7: Robot kolunun simülasyondaki başlangıç konumu

6.3.4.2 Deneyisel Çalışma

Bu bölümde yapılan çalışmalar ters kinematik hesaplarıyla robotun eklem noktalarının hız vektörlerinin hesaplanması ve elde edilen vektörlerden robotun uygun bir şekilde sürülmesi esasına dayanmaktadır. Ters kinematik hesapları oldukça karmaşık matris işlemleri yapılmasını gerektirmektedir. Robotu her 40ms de bir sürebilmek için matris işlemlerinin çok hızlı yapılabildiği GSL (GNU Scientific Library) kütüphanesinden yararlanılmıştır.

6.3.4.3 Kullanılan Kütüphaneler

a) GSL Kütüphanesi

GNU Bilimsel Kütüphane (GNU Scientific Library), C ve C++ programcıları için hazırlanmış bir sayısal (numerical) kütüphanesidir. GNU Halk Lisansı (GNU Public License) ile dağıtılan açık bir yazılımdır. Kütüphane rasgele sayı üretici, özel fonksiyonlar ve en küçük kareler oturtma işlemleri gibi oldukça geniş matematiksel rutinler içermektedir. Ayrıntılı bir test süitiyle birlikte 1000'in üzerinde fonksiyon içermektedir. Kütüphanenin sağladığı modüller aşağıda listelenmiştir:

- | | |
|------------------------|---------------------------|
| — Complex Numbers | — N-Tuples |
| — Roots of Polynomials | — Monte Carlo Integration |
| — Special Functions | — Simulated Annealing |
| — Vectors and Matrices | — Differential Equations |
| — Permutations | — Interpolation |

- Sorting
- BLAS Support
- Linear Algebra
- Eigensystems
- Fast Fourier Transforms
- Quadrature
- Random Numbers
- Quasi-Random Sequences
- Random Distributions
- Statistics
- Histograms
- Numerical Differentiation
- Chebyshev Approximation
- Series Acceleration
- Discrete Hankel Transforms
- Root-Finding
- Minimization
- Least-Squares Fitting
- Physical Constants
- Discrete Wavelet Transforms
- IEEE Floating-Point

b) PA10 Sürücüsü

PA10-7C robot kolu için İTÜ Robotik Laboratuvarı'nda geliştirilmiş bir sürücüdür. ISA ArcNet kartı üzerinden robotu kontrol etmektedir ve Linux'un ISA kernel sistem çağrılarını kullanmaktadır. Genel bir ifadeyle robotun her eklemının hız ve tork bilgisini okumakta ve her eklemi için hız, tork veya kilit parametrelerini yollayabilmektedir. Bunun dışında robota komut yollamadan önce her eklemının açısını kontrol edip yanlış bir açıdan kaynaklanacak eklem kilitlenmesini önlemektedir.

6.3.4.4 Geliştirilen Kütüphaneler

a) MatrixLibrary

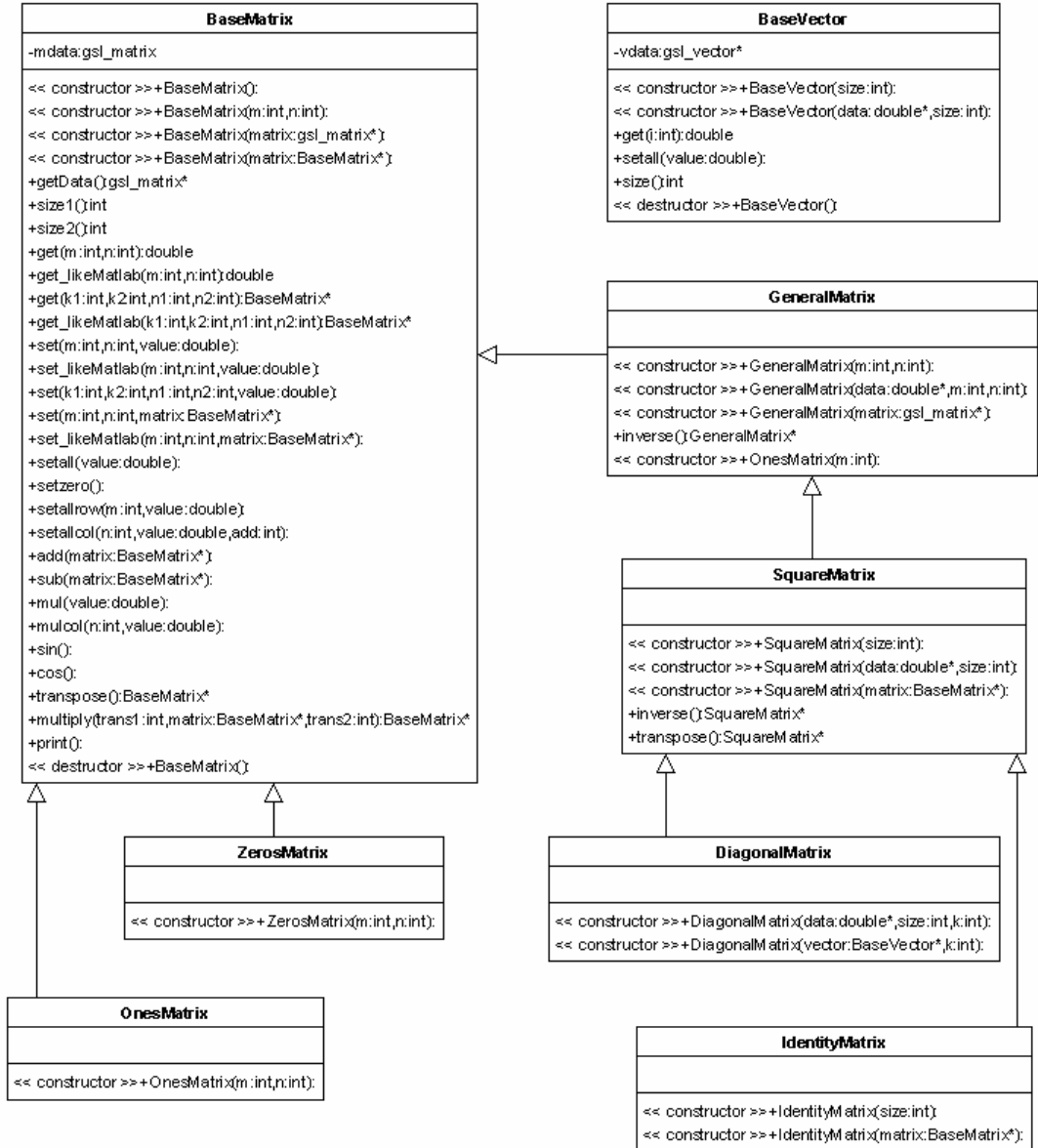
Amacı, GSL kütüphanesindeki matris işlemlerini nesneye yönelimli olarak kullanılabilir hale getirerek ters kinematik hesaplarındaki bazı özel Matlab fonksiyonlarını (örn. pinv) gerçekleştirmektir. Kütüphanenin sınıf ilişkileri Şekil 6.8'da verilmiştir. Daha detaylı bilgi için EK 4A'ya bakınız.

b) InverseKinematicLibrary

Amacı, ters kinematik işleminde kullanılacak hesapları yapacak nesneye yönelimli genel bir kütüphane oluşturmaktır. Robotun fiziksel özellik bilgileriyle oluşturulan nesne hareket edilmek istenen lineer ve açısal hız vektörlerini parametre olarak aldıktan sonra hareket esnasında her adımdaki eklem hızlarının ne olacağı bilgisini hesaplayabilmektedir. Alınan bu veriler robota iletilerek istenen hareket yaptırılmış

olmaktadır. Kütüphanenin sınıf ilişkileri Şekil 6.9’de verilmiştir. Daha detaylı bilgi için EK 4B’ye bakınız.

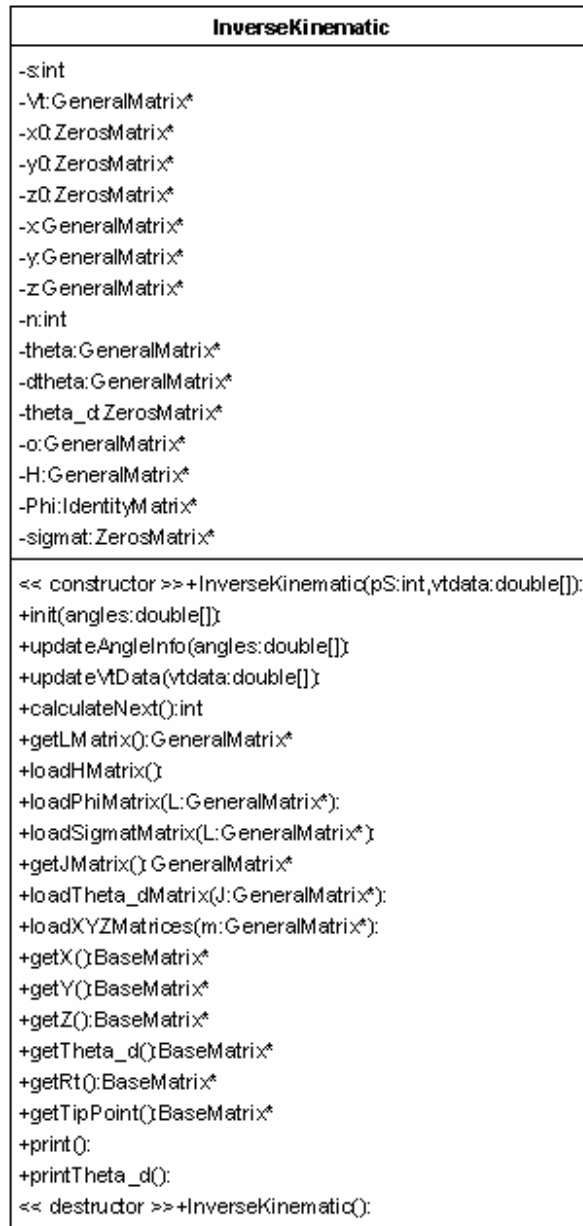
Kütüphane geliştirilirken robotun tekil nokta kontrolleri dikkate alınmamıştır. Çünkü çalışma uzayı böyle bir problem oluşturmayacak şekilde seçilmiştir. Çalışmada robotik kontrol modülünden beklenen tek özellik robotu istenen bir doğrultudan sapmadan oldukça hassas adımlarla ilerletebilmesidir.



Şekil 6.8: Matris kütüphanesi sınıf diyagramı

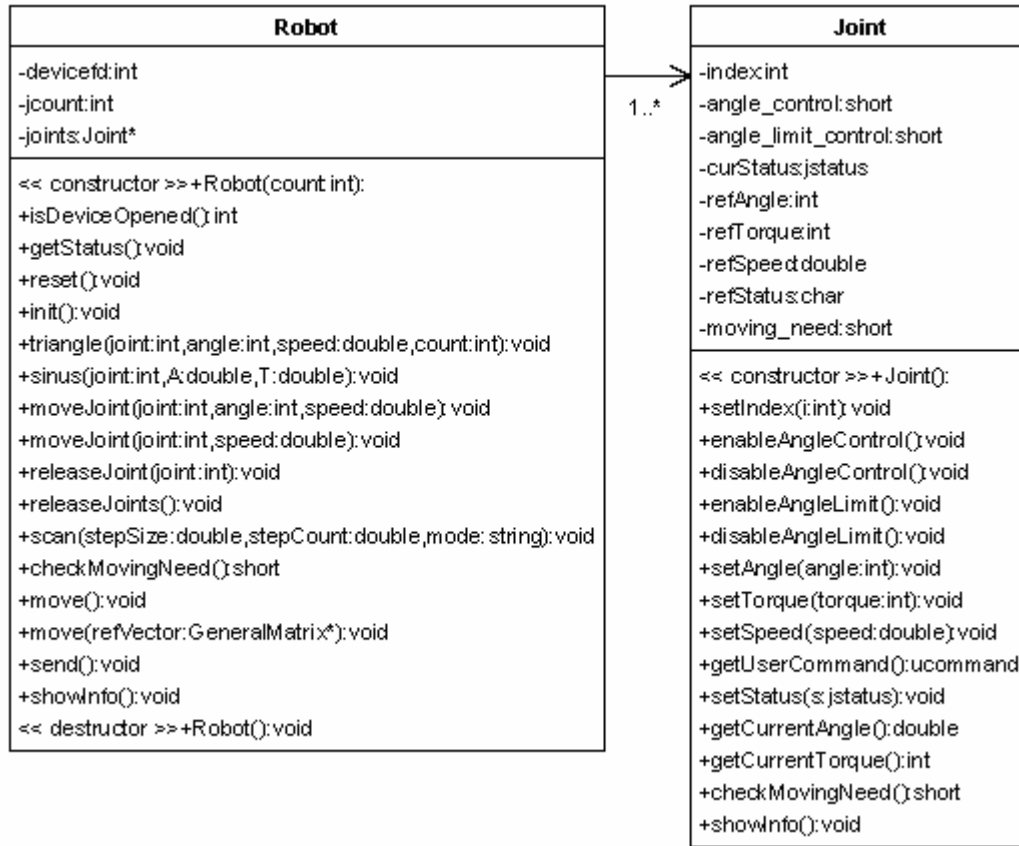
c) RobotLibrary

Amacı, PA10 robotunu bir nesne olarak modelleyip çalışmada kullanılacak temel kontrollerin ve hareketlerin yapılabilceği bir kütüphane oluşturmaktır. Robot ile haberleşmeyi bilgisayarın ISA slotuna bağlanan bir ArcNet kartı için yazılan robot sürücüsü üzerinden yapmaktadır. Çalışma sırasında robotun yanlış sürülmesinden kaynaklanacak eklem limitini aşma problemlerini önlemek amacıyla her eklem için kritik açı limitleri tanımlanmıştır. Kütüphanenin sınıf ilişkileri Şekil 6.10'da verilmiştir. Daha detaylı bilgi için EK 4C'ye bakınız.

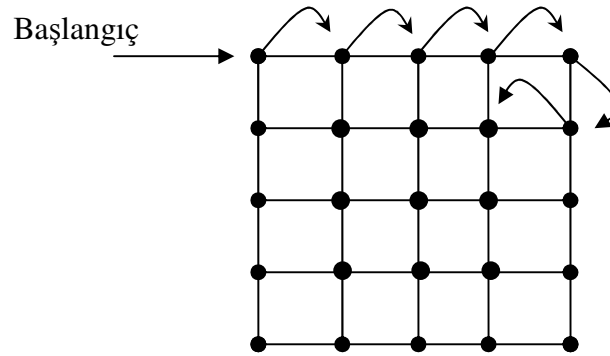


Şekil 6.9: Ters kinematik kütüphanesi sınıf diyagramı

Gerçeklenen yöntem kameranın ya da incelenecek bölgenin Şekil 6.11’de görüldüğü gibi belirlenen bir tarama ızgarasındaki noktaları doğru bir şekilde takip etmesi ve her noktaya ilişkin görüntü alması esasına dayanmaktadır. Alt piksel değişimlerini yakalayabilmek amacıyla robotun uç noktasının çok hassas bir şekilde hareket etmesi ve doğru konumlanması gerekmektedir. Çalışmada test kinematik hesapları sonrasında zaman içinde robotun mekanik aksamından kaynaklanan küçük hataların oluştuğu gözlemlenmiştir. Bu hataları giderebilmek amacıyla PI kontrolden yararlanılmıştır. PI kontrolde belirlenen Ki ve Kp parametreleri ile robotun uç noktasının bulunması istenen konum etrafında ihmal edilebilir düzeyde çok küçük salınımlar yapması sağlanmaktadır. Bu sayede oluşan hatalar minimize edilerek daha sağlıklı bir konum kontrolü sağlanmış olmuştur.

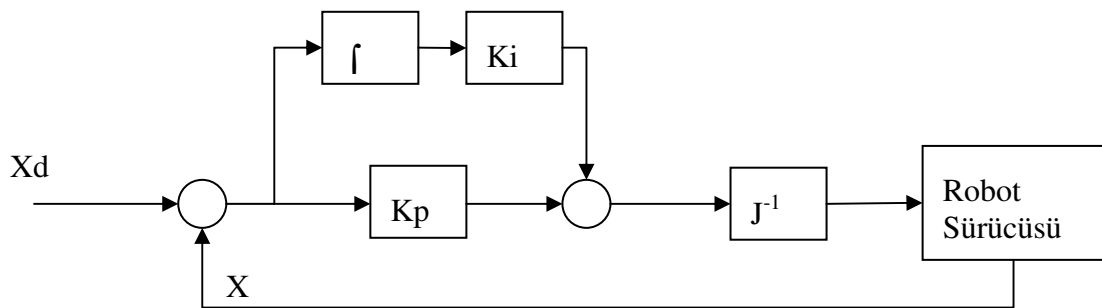


Şekil 6.10: Robot kütüphanesi sınıf diyagramı

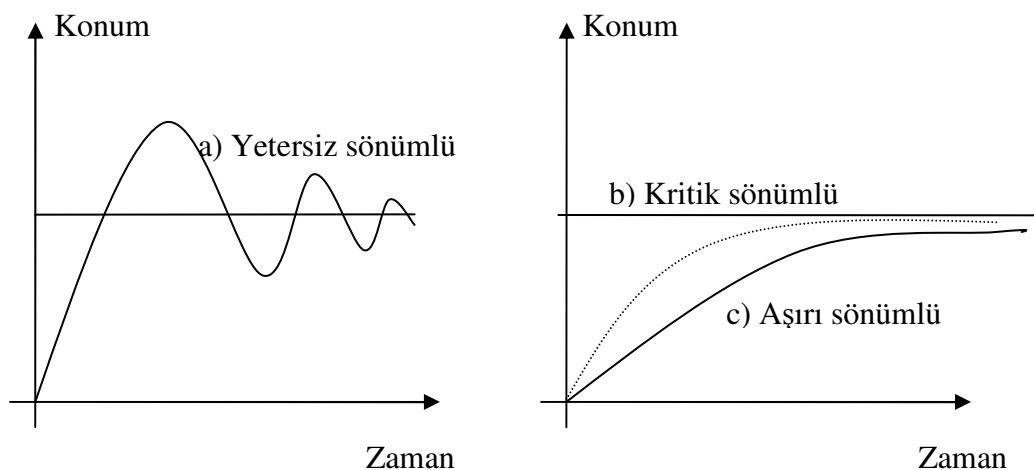


Şekil 6.11: Robotun uç noktasının tarama ızgarası üzerindeki dolaşımı

Şekil 6.12’de görülen Xd bulunulması istenen konum, Kp ve Ki kalibrasyon parametreleri ve X ise robotun o andaki gerçek konum bilgisidir. Çalışmada uygun Ki ve Kp değerleri bulunarak Şekil 6.13c’de görülen kritik sönümlü hareket sağlanmaya çalışılmıştır.



Şekil 6.12: PI kontrol devresi



Şekil 6.13: PI kontrolde K_i ve K_p 'ye bağlı hareket modelleri

d) TimerLibrary

Amacı, işletim sisteminin sistem saatini kullanarak istenilen süre kadar kod çalışmasını bekletmek ve belirtilen bir başlangıç zamanına göre geçen süreyi hesaplamaktır. Kütüphane mikro saniye hassasiyetinde zamanı ölçebilmektedir. Robotun her 40ms’de bir sürülmesi gerektiği düşünüldüğünde kütüphane ters kinematik hesaplarının işlem yüklerinin analiz edilmesinde de kullanılmıştır. Kütüphanenin sınıf ilişkileri Şekil 6.14’da verilmiştir. Daha detaylı bilgi için EK 4D’ye bakınız.

Timer
-start_tv:struct tim eval -end_tv:struct tim eval -running:int
<< constructor >>+Timer(): +start():void +end():void +elapsedMicrosec():long +elapsedMilisec():long +waitTo(milisecond:int):void +wait(milisecond:int):void +startTimeToString():string +endTimeToString():string

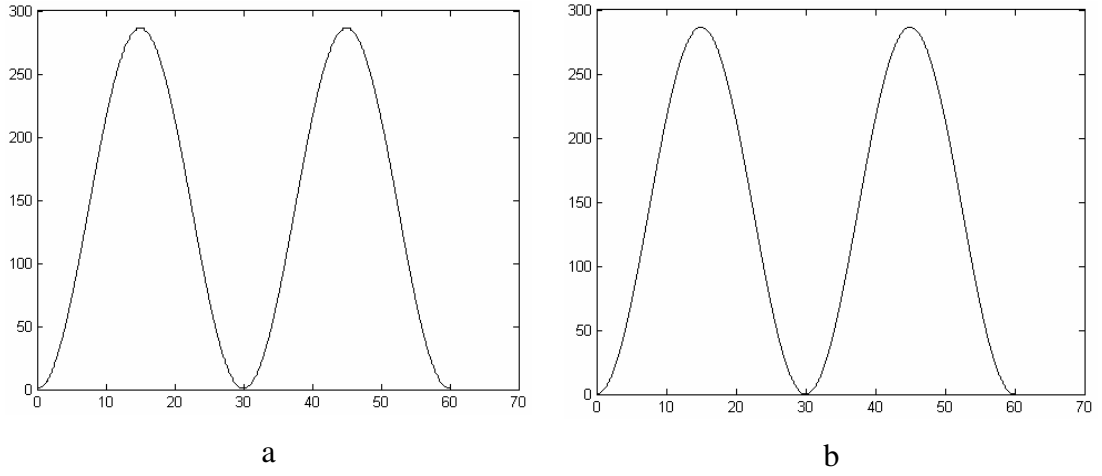
Şekil 6.14: Zamanlayıcı kütüphanesi sınıf diyagramı

6.3.4.5 Hassasiyet Denemeleri

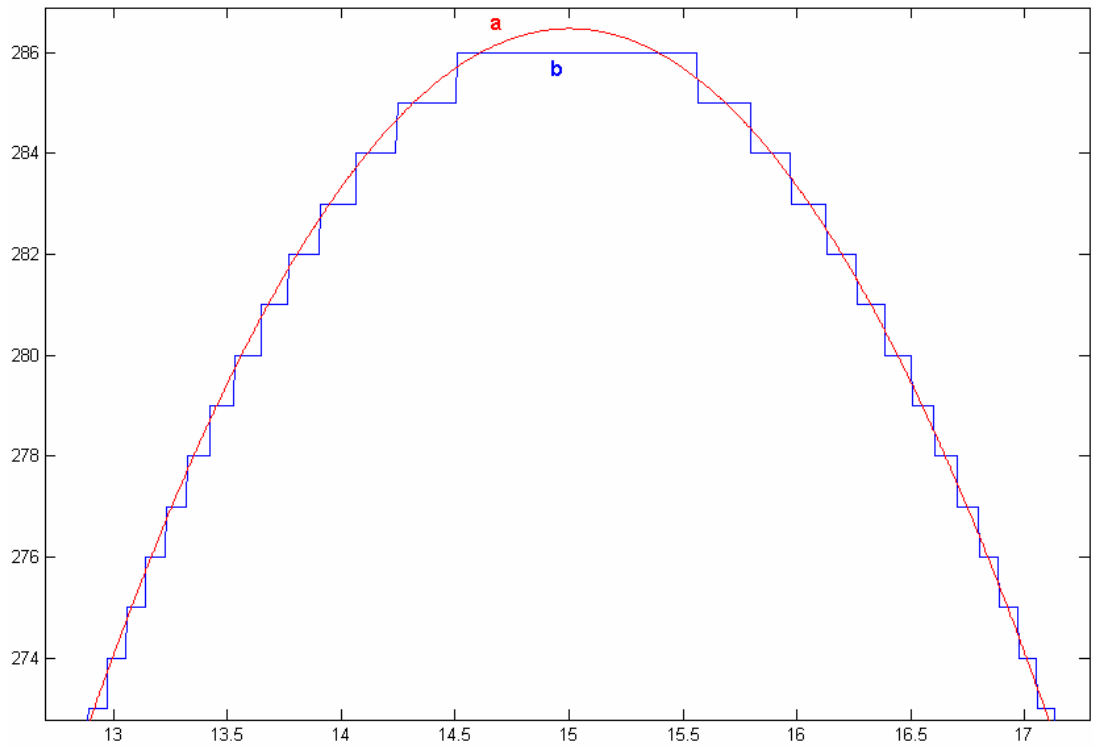
a) Eklem Açısıl Kontrol Testi

Yapılan bu denemede robot kolunun seçilen eklemine $A\sin(2\pi/T)$ hareketi yaptırılmaktadır. A hareketin genliğini, T ise hareketin periyodunu belirlemektedir. Robot kolunun açı kontrol hassasiyeti 1^0 ’dir. 6. eklem için deneme sonucu Şekil 6.15’de verilmiştir.

$A=30, T=30$ için:



Şekil 6.15: a) Robota yollanan komutlardan teorik hesaplanan konum değerleri; b) Eklemden alınan açı değerleri



Şekil 6.16: Teorik ve gerçek veriler üst üste bindirilmiştir. a) Teorik hesaplanan konum; b) Eklemden alınan açısal konum

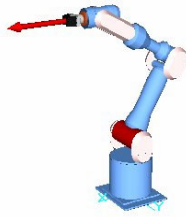
Deney sonuçlarından da görüleceği üzere robot eklemi verilen açısal konumu başarıyla takip etmiştir.

b) Ters Kinematik Hassasiyet Testi

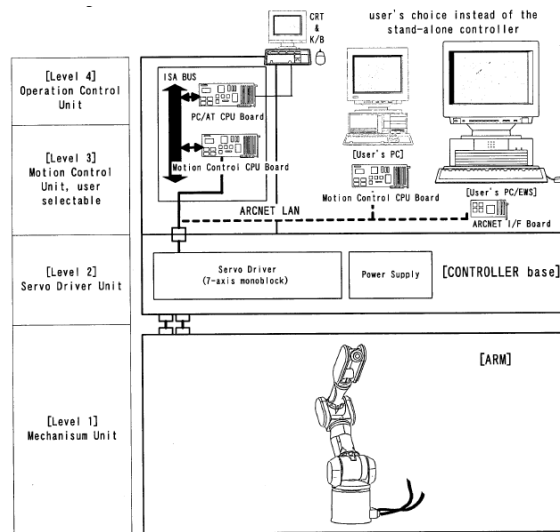
Yapılan bu denemede robot kolu başlangıç konumuna getirildikten sonra tek düzlemde hareket ettirilmiştir ve tepe noktanın taban eksen koordinatlarına göre vektör değişimleri gözlemlenmiştir.

6.3.5 PA10-7C Robot Kolunun Yapısı

Mitsubishi tarafından genel amaçlı üretilen 7 serbestlik dereceli artımlı bir robot koludur. 35 kg ağırlığıyla benzerlerine göre oldukça hafif olan PA10 10kg ağırlık taşıyabilmektedir. Yağmur, toz, su ve sıcaklık gibi zorlu çevre şartlarına dayanıklı olarak imal edilmiştir. Buna ek olarak Şekil 6.17’de görüldüğü üzere açık sistem yapısında modellenmiştir, bu sayede sadece üretim, montaj ve fabrikasyon konularının dışında da farklı birçok iş için rahatlıkla programlanabilir ve kullanılabilir hale getirilmiştir. Robotun mekanik yapısı EK 1’de verilmiştir.



Şekil 6.17: PA10-7C robotunun 3 boyutlu görüntüsü



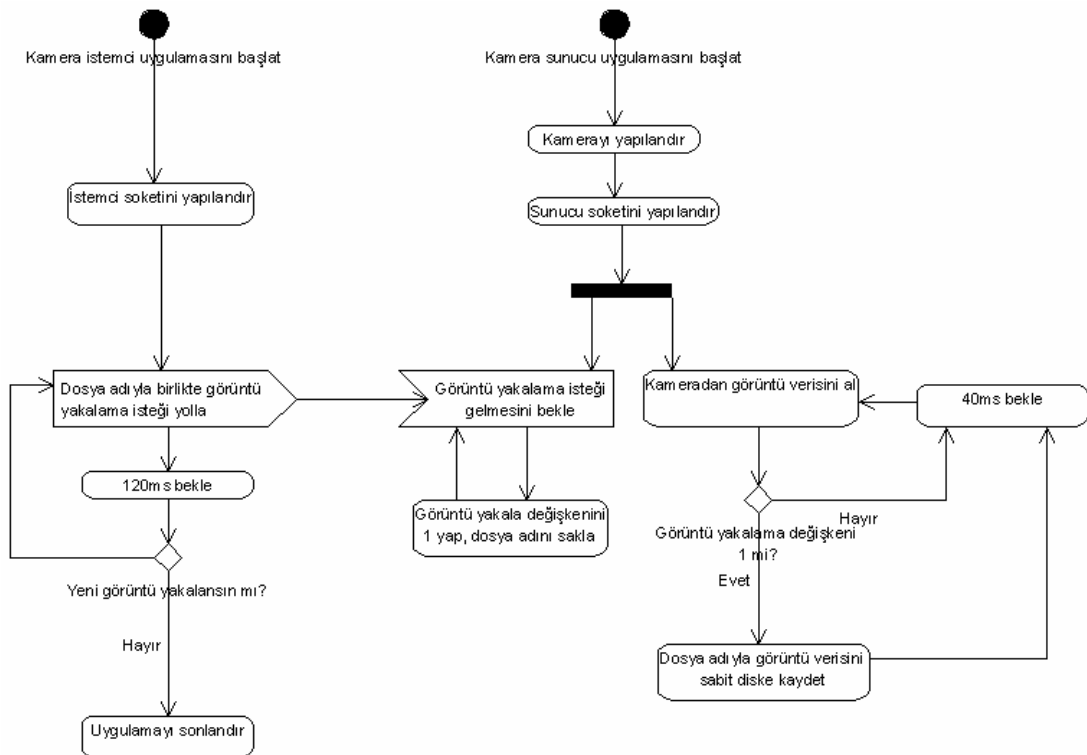
Şekil 6.18: PA10 robotunun açık sistemi

6.4 Kamera Kontrolü Çalışması

6.4.1 Çalışmanın Amacı

Kamera kontrolü çalışmasını yapmaktaki amaç uzaktaki bir bilgisayardan komut yollayarak kameradan görüntü yakalanmasını sağlamaktır. Bu sayede robotu kontrol eden bilgisayar ile kamerayı kontrol eden bilgisayar birbirinden ayrılabilir. Çalışmada bilgisayarlar arasındaki haberleşme için Şekil 6.19’da görülen protokol kullanılmaktadır. Bu protokol ile kamera TCP/IP protokolüyle ağ üzerinden yönetilebilir olmaktadır.

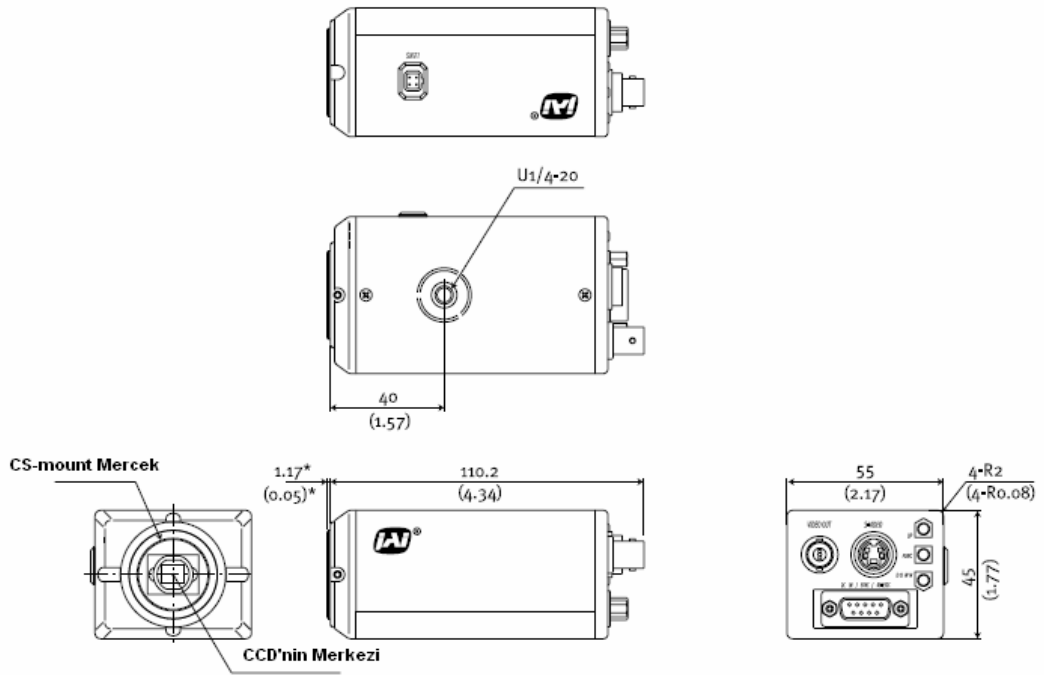
Yapılan deneysel çalışmalar için C++ ile yazılan kütüphaneler, hassasiyet denemelerindeki verileri analiz etmek için Matlab kullanılmıştır.



Şekil 6.19: Kamera kontrolü protokolü

6.4.2 Kullanılan Kameranın Yapısı

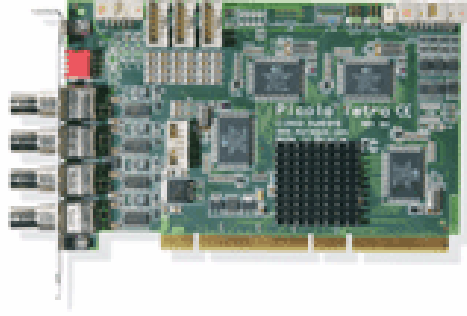
Yapılan çalışmalarda JAI firmasının ürettiği CV-S3200 model yüksek çözünürlüklü, renkli ve PAL/NTSC formatlarında görüntü alınabilen kamera kullanılmıştır. Bu kamera JAI firmasının yeni DSP işlemcileriyle çalışmaktadır ve oldukça yüksek sinyal gürültü oranına ve veri akışına sahiptir. Kendi RAM'larına sahip kamera DSP işlemcisiyle birlikte %100 yatay çözünürlük sağlamaktadır. Kameranın şematiği Şekil 6.20'de görülmektedir. Teknik özellikleri EK 2'de verilmiştir.



Şekil 6.20: Kullanılan kameranın şematiği

6.4.3 Kullanılan Resim Sayısallaştırıcının (Frame Grabber) Yapısı

Yapılan çalışmada Euresys firmasının ürettiği Pico Tetra model resim sayısallaştırıcı kartı kullanılmıştır. Pico kartları standart kameralarla uyumlu çalışan video yakalama kartlarıdır. Güvenlik, gözetleme ve bilgisayarla görme gibi ileri teknoloji uygulamalarında kullanılması amacıyla üretilmiştir. Kamera tarafından sağlanan çoğu veriyi iyi bir görüntü elde edilmesi için kısa zamanda sayısallaştırır. Maliyeti oldukça ucuz olan bu kartlar aynı anda paralel olarak birçok kameradan gelen veriyi gerçek zamanlı olarak yakalayabilmektedirler.



Şekil 6.21: Euresys marka Pico Tetra model resim sayısallaştırıcı

6.4.4 MultiCam Sürücüsü

Üst seviye bir kart sürücüsü olan MultiCam, Euresys marka resim sayısallaştırıcıların işlemlerini yöneten bir yazılım katmanıdır. Etkin ve kolay kullanılabilir soyutlama katmanı ile C fonksiyonlarıyla donanımın kontrol edilebilmesini sağlamaktadır. Sürücü sadece Euresys kartlarına özeldir, başka kartlarla çalışmamaktadır.

Donanımı yöneteceği parametreler 4 sınıf altında toplanmıştır; Kart (board), Kanal (channel), Yüzey (surface), Konfigürasyon (configuration). Kart sınıfındaki parametreler resim seçinin Giriş/Çıkış özelliklerini tanımlamak için kullanılır. Kanal, MultiCam sisteminin ana bileşenidir. Görüntü yakalama işlemi sırasında verinin kameradan alınıp bilgisayarın belleğinde saklanıncaya kadar gereken donanım ve yazılım kontrollerini yapar. Yüzey, alınan 2 boyutlu görüntülerin bilgisayarda saklandığı bellek alanıdır; donanımsal tampon bellekler de vardır. Konfigürasyon sınıfındaki parametreler Euresys kartlarını kullanmak isteyen bilgisayardaki başka yazılım veya donanım bilgilerini tanımlar.

6.4.5 Gerçeklenen Çalışma

6.4.5.1 DeneySEL Çalışma

Bu bölümde yapılan çalışmalar ağ üzerinden kamera görüntüsünün yakalanabilmesi ve sabit diske kaydedilmesidir. Ayrıca çevresel gürültüler analiz edilmiş ve farklı ışık kaynakları kullanılarak elde edilen sonuçlardaki iyileşmeler gözlemlenmiştir.

Kamera 40ms’de yeni bir görüntü yakalayabilmektedir, ancak yakalanan görüntünün sabit diske kaydedilmesi oldukça büyük işlem zamanları gerektirmektedir. Bu

sebeple harcanan süreyi azaltmak için kameradan yakalanan tüm görüntünün değil de, yakalanan görüntünün bir bölümü kesilip diske kaydedilmektedir. Böylece diske yazılmak istenen veri boyu azaldığından istemci tarafındaki bekleme süresi 120ms'ye kadar düşürülmüştür.

6.4.5.2 Kullanılan Kütüphaneler

a) OpenCV Kütüphanesi

OpenCV (Open Source Computer Vision) Intel firması tarafından geliştirilen açık kaynak kodlu görüntü işleme kütüphanesidir. Birçok karmaşık görüntü işleme algoritmalarını oldukça hızlı ve başarılı bir şekilde gerçekleyebilen 300'ün üzerine C fonksiyonları ve C++ sınıflarından oluşmaktadır. Platform bağımlılığı oluşturmamaktadır. Buna rağmen Intel tabanlı işlemcilere bağımlılık oluşturmaktadır. Intel IPP kütüphanelerine (Integrated Performance Primitives) ara yüz sağlamaktadır.

b) MultiCam Sürücüsü

Sürücü hakkında detaylı bilgi Bölüm 6.4.4'de verilmiştir.

6.4.5.3 Geliştirilen Kütüphaneler

a) ImagingLibrary

Amacı, OpenCV kütüphanesindeki görüntü işlemlerini nesneye yönelimli olarak kullanılabilir hale getirmektir. Bu sayede çözünürlük iyileştirme algoritmasındaki işlemler kolaylıkla gerçekleştirilmiş olmaktadır. Kütüphanenin sınıf ilişkileri Şekil 6.22'de verilmiştir. Daha detaylı bilgi için EK 4E'ye bakınız.

b) FrameGrabberLibrary

Amacı, Euresys marka resim sayısallaştırıcının (frame grabber) MultiCam adındaki sürücüsündeki işlemleri nesneye yönelimli olarak kullanılabilir hale getirmektir. Bu sayede kamerayı ön koşullandırma ve görüntü yakalama işlemleri kolaylıkla gerçekleştirilmiş olmaktadır. Kütüphanenin sınıf ilişkileri Şekil 6.23'de verilmiştir. Daha detaylı bilgi için EK 4F'ye bakınız.

Image
-data:IplImage* -matrix_data:CvMat*
<< constructor >>+Image() << constructor >>+Image(name:const char*); << constructor >>+Image(size:CvSize,depth:int,channel:int,bgcolor:int); << constructor >>+Image(image:IplImage*); << constructor >>+Image(size:CvSize,buffer:uchar*); +createMatrixData(size:CvSize):void +getData():IplImage* +getSize():CvSize +getPixelStep():int +getPixelPointer(x:int,y:int):uchar* +getPixel(x:int,y:int):uchar +getMatrixValue(x:int,y:int):double +putPixel(x:int,y:int,value:int):void +setMatrixValue(x:int,y:int,value:double):void +show():void +paste(x:int,y:int,image:Image*):void +fill(value:int):void +convertToBinary(threshold:int):void +save(filename:const char*):void +exportImage(filename:const char*):void +marshall(filename:const char*):void << destructor >>+Image();

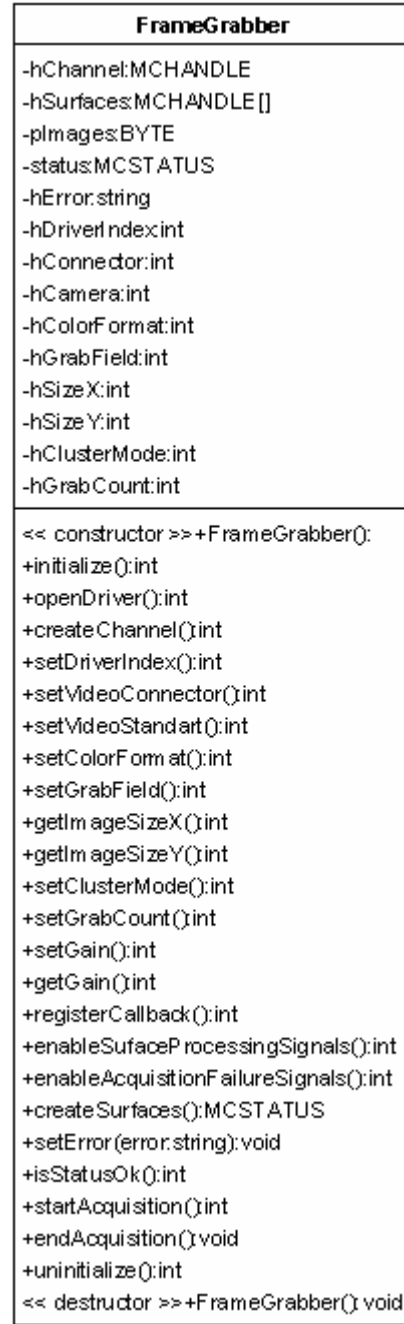
Şekil 6.22: Görüntü kütüphanesi sınıf diyagramı

c) SocketLibrary

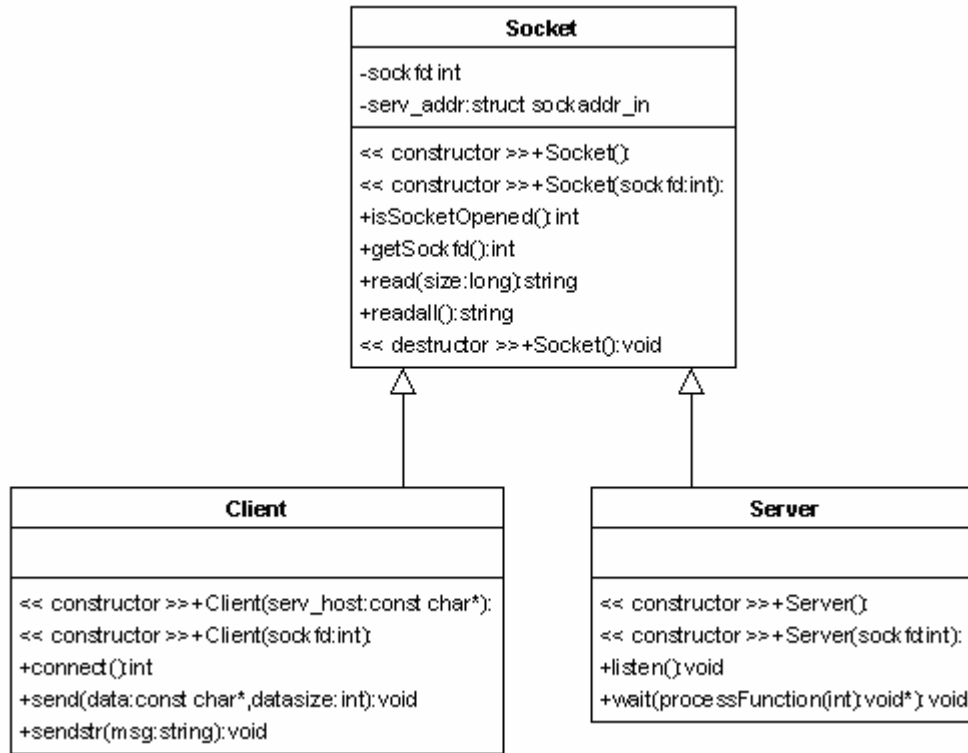
Amacı, TCP/IP haberleşmesinin nesneye yönelik olarak kullanılabilmesini sağlamaktır. Genel amaçlı olarak hazırlanan bu kütüphane robot kontrol bilgisayarı ve kamera kontrol bilgisayarı arasındaki iletişimde kullanılacaktır. Kütüphanenin sınıf ilişkileri Şekil 6.24’da verilmiştir. Daha detaylı bilgi için EK 4G’ye bakınız.

d) CameraSocketLibrary

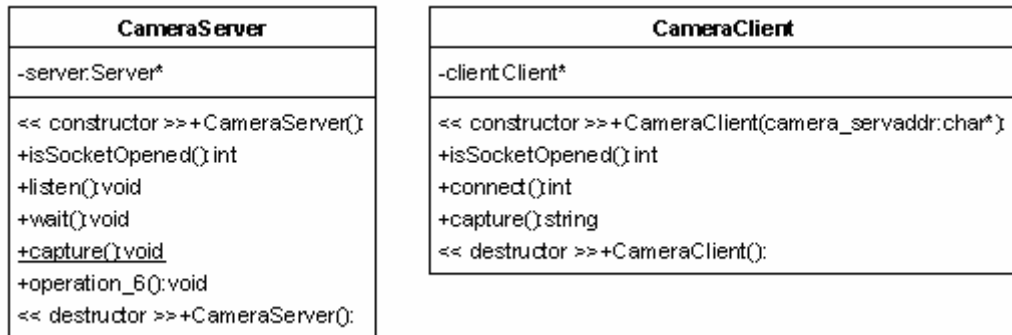
Amacı, kameranın TCP/IP üzerinden nesneye yönelik olarak kullanılabilmesini sağlamaktır. Kütüphane bir sunucu ve bir istemci bileşenden oluşmaktadır. İstemci sunucuya dosya adı ile birlikte görüntü yakalama isteğinde bulur, sunucu da gelen isteği karşılar ve kameradan aldığı görüntüyü belirtilen dosya adında dosya sistemine saklar. Kütüphanenin sınıf ilişkileri Şekil 6.25’de ve çalışma yöntemi Şekil 6.26’de verilmiştir. Daha detaylı bilgi için EK 4H’ye bakınız.



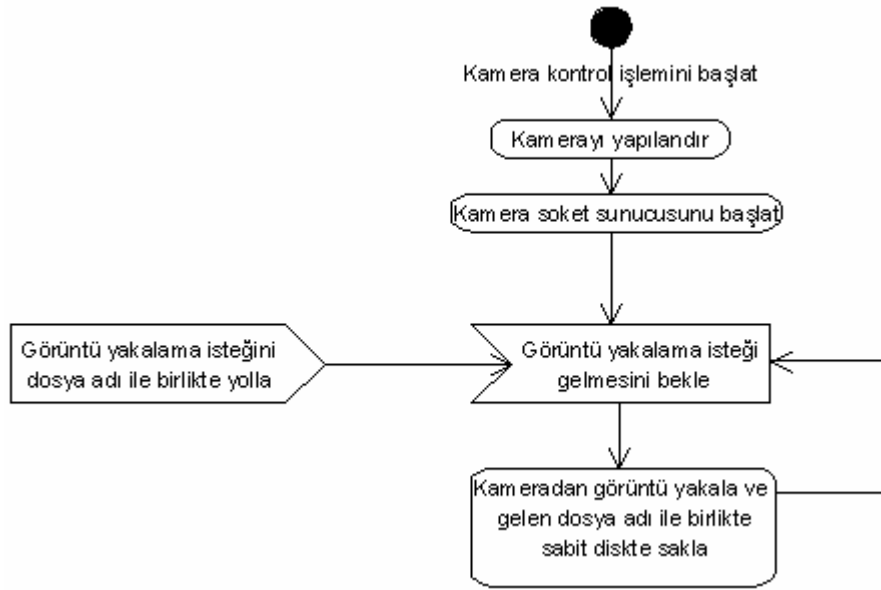
Şekil 6.23: Resim sayısallaştırıcı kütüphanesi sınıf diyagramı



Şekil 6.24: Soket kütüphanesi sınıf diyagramı



Şekil 6.25: Kamera soket kütüphanesi sınıf diyagramı



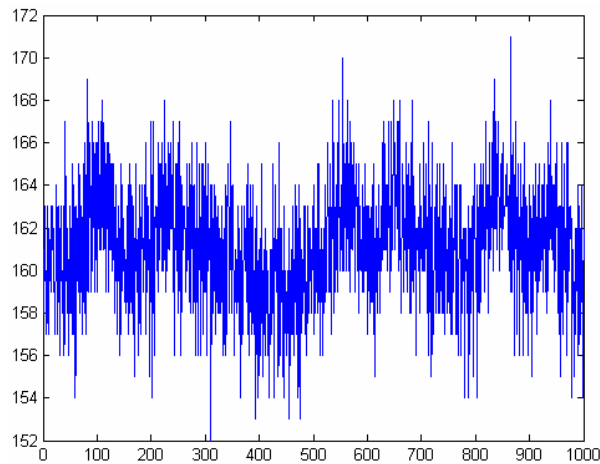
Şekil 6.26: Kamera haberleşme diyagramı

6.4.5.4 Hassasiyet Denemeleri

a) Farklı Işık Kaynakları Testi

Bu çalışmada incelenecek bölgenin aydınlatılması için 3 farklı ışık kaynağı kullanılmıştır. Her ışık kaynağı için durağan kamera ve çalıştırılmayan robot kolu koşullarında 1000 tane örnek alınmıştır. Alınan örneklerden hesaplanan piksel değişimleri aşağıda verilmiştir.

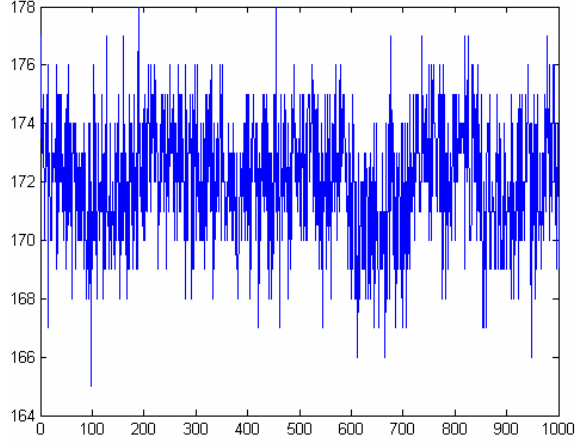
1) Flüoresan Lamba



Şekil 6.27: Flüoresan lamba ile gözlemlenen piksel değişimi

Flüoresan lamba kullanıldığında piksel ortalama değeri 161.1450 ve piksel değişim varyansı 9.4314 olarak hesaplanmıştır. Varyansın yüksek çıkması flüoresan lambanın şebeke frekansından dolayı titreşmesinden kaynaklanmıştır.

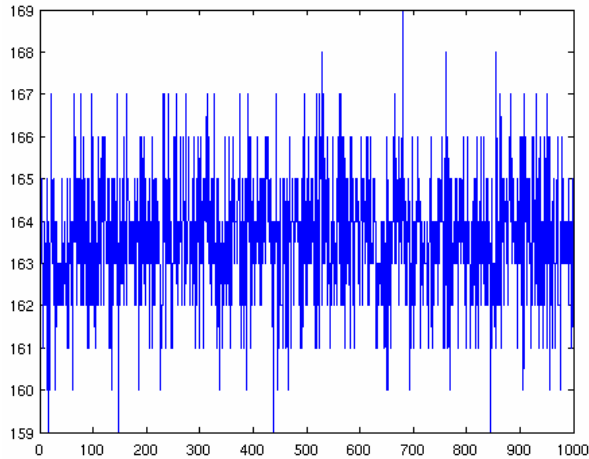
2) 220V 100W Akkor Lamba



Şekil 6.28: Akkor lamba ile gözlemlenen piksel değişimi

Akkor lamba kullanıldığında piksel ortalama değeri 172.0760 ve piksel değişim varyansı 4.8811 olarak hesaplanmıştır. Varyansın flüoresan lambaya göre daha düşük çıkmasının sebebi akkor lambadaki çalışma sıcaklığı sebebiyle titreşim ataletinin yüksek olmasından kaynaklanmaktadır.

3) 12V 50W Halojen Lamba

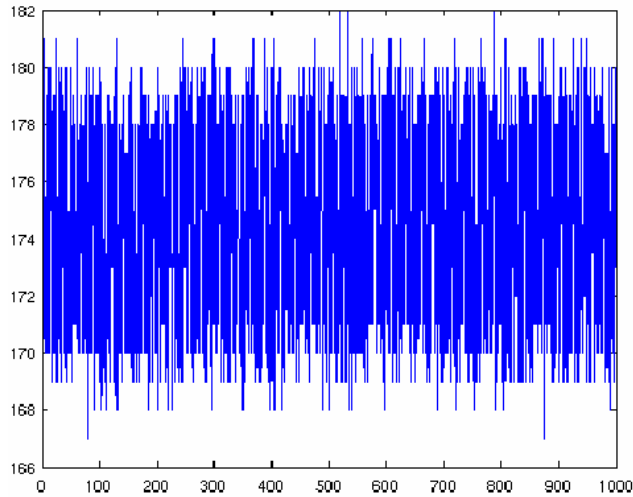


Şekil 6.29: Halojen lamba ile gözlemlenen piksel değişimi

Halojen lamba kullanıldığında piksel ortalama değeri 163.5870 ve piksel değişim varyansı 2.4829 olarak hesaplanmıştır. Varyansın akkor lambaya göre düşük çıkmasının sebebi doğrusal akımdaki sabit gerilimden kaynaklanmaktadır.

b) Robot Kolu Çalışma Titreşimi Testi

Bu çalışmada robot kolunun çalışma sırasındaki mekanik titreşimleri incelenmiştir. Işık kaynağı olarak halojen lamba kullanılmıştır.



Şekil 6.30: Halojen lamba ile robot çalışırken gözlemlenen piksel değişimi

İnceleme sonrasında piksel ortalama değeri 174.4290 ve piksel değişim varyansı 19.5085 olarak hesaplanmıştır. Bunun sebebi robotun hassas konumlaması için kullanılan PI kontrolden kaynaklanmıştır.

Şekil 6.30'da görüldüğü gibi halojen lamba kullanılarak piksel ortalama değeri hesaplanmasında elde edilen değer burada yüksek çıkmasının sebebi, ışık kaynağının düzlem üzerine düzgün ışık yayamamasından kaynaklanmıştır.

7. DENEYSEL ÇALIŞMA SONUÇLARI

7.1 Karşılaşılan Problemler

Deneysel çalışmada simülasyon çalışmasından farklı olarak donanımsal ve çevresel faktörlerden kaynaklanan bazı problemlerle karşılaşmıştır. Bu problemler şu şekilde sayılabilir:

a) Görüntü sensorunun sabit bir noktadan farklı değerler yakalaması

Şekil 6.27’de görüldüğü üzere sabit duran görüntü, ışık, kamera ve çalıştırılmayan robot kolu koşullarında bile kameranın tek pikselinden yakalanan değerlerin her yakalamada farklı bir değer aldığı gözlemlenmiştir. Buradaki değer değişimleri kullanılan ışık kaynağına ve kamera sensorunun kalitesine bağlıdır.

b) Robot hareketindeki PI kontrolün titreşim yaratması

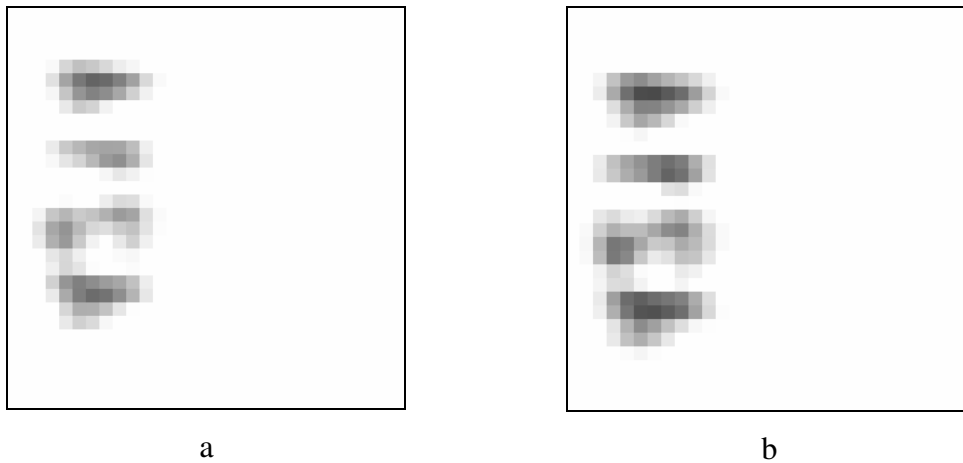
Yapılan çalışmada algoritma gereği görüntünün belli adım aralıklarında hareket ettirilerek alt piksel değerlerinin yakalanması amaçlanmıştır. Alt piksel değerleri yakalanması için adım aralıklarının çok küçük olması gerekmektedir ve hassas biçimde robot kolunun o noktada bulunması sağlanmalıdır. Robotun mekanik aksamından dolayı, eklemler sabit bir açıda tutulmak istense bile zaman içerisinde gözle görünür kayma ve dönme hataları meydana gelmiştir. Bu hataları yok edebilmek ve hassas bir hareket sağlayabilmek için ters kinematik hesaplarına PI kontrol de dâhil edilmiştir. PI kontrol hassas bir konumlama sağlamasına rağmen yapısı gereği tepe noktasının kabul edilebilir bir hata aralığında salınım yapmasına neden olmaktadır. Bu salınımlar Şekil 6.30’da görüldüğü üzere kamera pikselinden elde edilen değer değişiminin daha da artmasına neden olmuştur.

c) CCD sensorların algıladıkları ışık enerjisini tam sayı olarak sayısallaştırmaları (0-255)

Deneysel çalışmada kamera grilik seviyesi algılama kipinde çalıştırılmıştır. Bu kipte her piksel algılanan parlaklık ölçüsünde 0 ile 255 arasında bir tamsayı değere atanmaktadır. Çalışmada yapılan küçük kaydırmalar kameradan yakalan görüntü piksellerinde küçük değişimlere neden olmaktadır; çözünürlük iyileştirme için anlamlı alt piksel değerleri görüntüde oluşan bu küçük değişimler kullanılarak hesaplanmaktadır. Ancak pikseller tamsayı değerlerini alabildiğinden, oluşan ondalıklı değişimler göz önünde bulundurulamamaktadır. Dolayısıyla ele alınamayan ondalık değerleri, nicemleme (quantizing) hatalarına neden olmaktadır ve bu da sonuçta elde edilen yüksek çözünürlüklü görüntü kalitesini olumsuz etkilemektedir.

d) Bölge üzerine düşen ışık miktarının farklılık göstermesi

Kullanılan ışık kaynağı bölge üzerine aynı oranda ışık düşürmemektedir. Flüoresan kullanıldığında ışık yukarıdan geldiği için küçük titreşimler bölge üzerindeki gölgelenmeleri arttırmaktadır. Bölgenin karşısına yerleştirilen akkor ve halojen lamba ise merkeze daha çok ve kıyılara doğru gidildikçe azalan bir ışık yaymaktadır. Bu durumda görüntü kaydırıldığında yüzey üzerine düşen ışık miktarındaki değişikliklerden dolayı görüntünün o bölümünden çok farklı değerler elde edilmektedir. Bahsedilen etki Şekil 7.1’de görülmektedir.



Şekil 7.1: Bölge üzerine düşen ışık miktarının farklılık göstermesi, a) Işık miktarı daha fazla; b) Işık miktarı daha az

e) Kamera lensinden kaynaklanan görüntünün bombeleşmesi

Şekil 7.2’de görüldüğü üzere kamera lensi görüntüyü bombeli bir şekilde yakalamaktadır. Kısa mesafeli hareketler için problem yaratmayan bu özellik, uzun mesafe tarama denemelerinde görüntüdeki şekil bozukluğundan dolayı problem yaratmıştır.

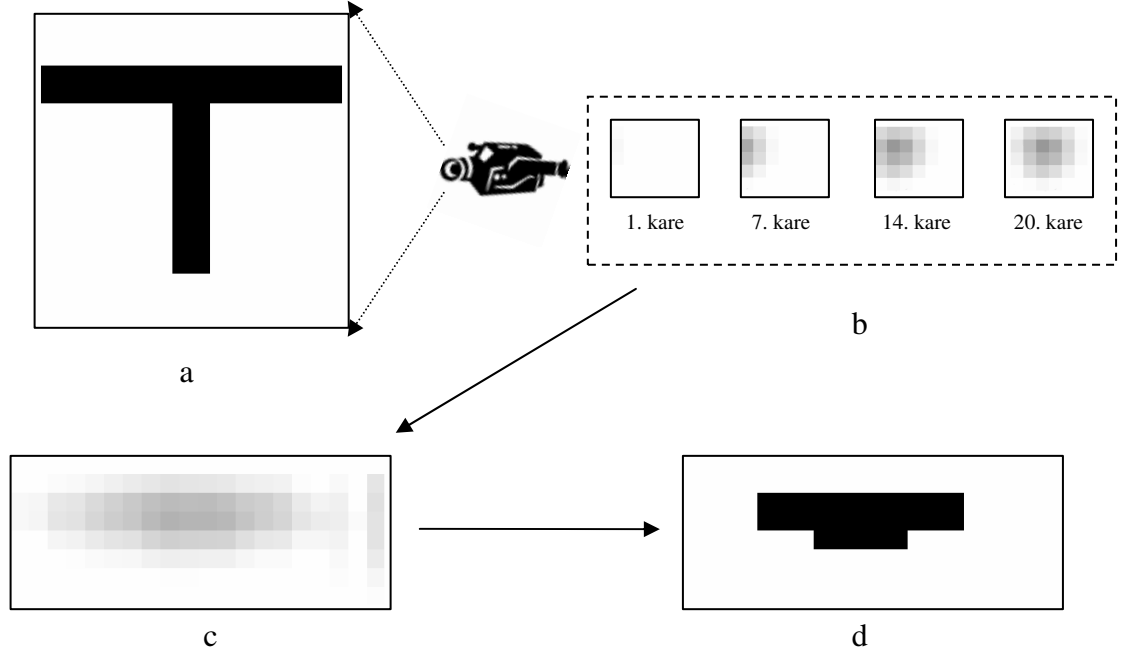


Şekil 7.2: Kamera lensinden kaynaklanan görüntü bombeleşmesi

7.2 Elde Edilen Sonuç

Deneysel çalışmada tek yönde kaydırma gerçekleştirilmiştir. Şekil 7.3a’da görülen şekil kamera ile 8x8 piksel olarak yakalanabilmiştir. Robot kolu yatay yönde 20 adım kaydırılarak tek yönde 8 piksel boyutundaki görüntü baştan sona taranmıştır. Şekil 7.3b’deki gibi elde edilen düşük çözünürlüklü görüntüler işlenerek Şekil 7.3c’de görülen yüksek çözünürlüklü görüntü elde edilmiştir. Deneysel çalışmada karşılaşılan gürültüleri minimize etmek amacıyla elde edilen görüntü üzerinde farklı eşik değerleriyle analiz yapılmıştır. Eşik değerinin değişimi ile görüntüdeki piksel değişimleri incelenmiştir. Ortaya çıkan sonuca göre 219–224 aralığında en uzun eşik

değeri değişimine karşılık görüntü piksellerinin değişmediği ve bu aralıkta, Şekil 7.3d’de görüldüğü üzere orijinal görüntüdeki şeklin algılanabilir olduğu gözlenmiştir.



Şekil 7.3: Deneysel çalışma sonuçları, a) Gerçek görüntü; b) Kamera görüntüsü; c) Oluşturulan yüksek çözünürlüklü görüntü; d) Yüksek çözünürlüklü görüntüye 220 eşik değeri uygulanması

Sonuçtan da anlaşılacağı üzere tek yönde kaydırma ile sadece yatay yönde çözünürlük iyileştirilmesi gerçekleştirilmiştir ve kameranın düşük çözünürlüklü olarak yakaladığı görüntüden algılanamayan şekil, çözünürlük iyileştirme işleminden sonra algılanabilir olmuştur.

Tablo 7.1’de kamera ile yakalanan görüntü ve çözünürlük iyileştirme işlemi sonucunda elde edilen görüntü için uygulanan farklı eşik değerlerinin sonuçları görülmektedir.

Tablo 7.1: Farklı eşik değerleri için kamera görüntüsünün ve elde edilen sonucun karşılaştırılması

Uygulanan Eşik Değeri	Kameradan Tek Kare ile Yakalanan Görüntü	Yöntemin Sonucunda Elde Edilen Görüntü
160		
180		
200		
220		
240		

8. SONUÇLAR VE TARTIŞMA

Yapılan tez çalışmasında bileşik gözlerin çalışma prensibinden esinlenerek bir çözünürlük iyileştirme yaklaşımı ortaya çıkartılmıştır. Simülasyon ortamında ve deneysel olarak sonuçları gösterilen bu yaklaşım görüntüyü belli adım aralıklarında tarayarak kameradan alınan düşük çözünürlüklü görüntüler üzerinde çözünürlük iyileştirmesi yapmaktadır.

Çalışmada model olarak kullanılan bileşik göz yapısına sahip böcek gözleri doğa koşullarında birçok avantaja sahiptirler. Örneğin odaklama problemleri yoktur, yakın mesafede çok küçük detayları algılayabilirler, panoramik bir görüntü sağlarlar, kırılma güçleridir, hareket değişimlerini çok hassas bir şekilde algılayabilirler, oldukça ufak olarak ve düz bir şekilde üretilebilirler. Tüm bu avantajlarından dolayı askeri ve sivil alanda yapay bileşik göz modelleri üretme üzerine çalışılmaktadır.

Ortaya atılan yöntem bileşik gözlerin görünüşü yakalama ve birleştirme özelliğini kullanmaktadır. Bileşik göz yapılarında farklı yönlerde bakan binlerce gözcük belli bir bölge üzerinden oldukça düşük çözünürlüklü veriler toplamaktadır ve bu veriler beyinde birleştirilerek bölgenin daha iyi çözünürlüklü görüntüsü ortaya çıkartılmaktadır. Farklı bölgelere yönelmiş gözcükler laboratuvar ortamında dijital kamera ve 7 serbestlik dereceli bir robot kolu kullanılarak modellenmiştir. Robot kolunun ucuna bağlanan görüntü belli konumlara getirilerek kamera ile gözcüklerde olduğu gibi bölge üzerinden farklı bakış açılarında görüntü verisi alınmakta ve bunlar geliştirilen algoritma ile böcek beyninde olduğu gibi işlenip birleştirilmektedir.

Böcek gözü modelinin bir diğer avantajı ölçeklenebilme özelliği sağlamasıdır. Birbirleriyle özdeş gözcüklerin sayısı ve yerleşimi değiştirilerek incelenen bölge üzerinden daha geniş (büyük adımlarla daha çok bölge tarama) veya daha yüksek (küçük adımlarla daha dar bir bölge tarama) çözünürlüklü bir görüntü elde edilebilmektedir.

8.1 Simülasyon Sonuçları

Geliştirilen yöntemin simülasyon çalışması Matlab ortamında gerçekleştirilmiştir. Bu ortamda hem kare tarama hem de tek yönde tarama çalışması yapılmıştır. Elde edilen sonuçlar aşağıdaki gibidir:

- Tarama sırasında adım aralıkları azaldıkça çözünürlük iyileştirme oranının daha da arttığı gözlemlenmiştir. Adım aralıkları arttırıldıkça gerçek görüntünün daha kaba bir formu elde edilmiştir.
- Tarama sırasında kullanılan kameranın sensor sayısı arttıkça elde edilen görüntünün çözünürlük iyileştirme oranının arttığı gözlemlenmiştir. Sensor sayısı gerçek görüntünün piksel sayısına eşit olduğunda, gerçek görüntünün birebir aynısı elde edilmiştir.
- Piksellerdeki tam sayı nicemlemesinden (quantization) dolayı elde edilen görüntüde gürültüler meydana gelmektedir. Kameranın sensor sayısı arttıkça oluşan hata piksellere dağılıp azalacağından elde edilen görüntüdeki gürültüler azalmaktadır.
- Yöntemde yüksek çözünürlük pikselleri döngüsel olarak elde edilmektedir ve bir piksel değeri kendisinden önceki piksellere bağımlı olarak hesaplanmaktadır. Yöntemin veri yakalama özelliğinden dolayı bir piksel değeri hesabında hata oluşması sadece kendine komşu piksellerde bozulmalara neden olmaktadır ancak hatalar ötelenmemektedir.
- Yöntemden ikili görüntülerde gri-seviyeli görüntülere göre daha başarılı sonuçlar elde edilmiştir.

8.2 Deneysel Sonuçlar

Geliştirilen yöntemin deneysel çalışması İTÜ Kontrol Mühendisliği Robotik Laboratuvarında yapılmıştır. Deneylerde elde edilen sonuçlar aşağıdaki gibidir:

- Kullanılan PA10-7C robotu ters kinematik hesaplarıyla istenilen bir yönde 0.1mm hassasiyetle hareket ettirilebilmektedir

— Robot konumlama hassasiyeti için kullanılan PI kontrol kamera ile elde edilen piksel değerlerinde aşırı titreşimlere neden olmaktadır

— Doğrusal akımla beslenen bir ışık kaynağı kullanılarak robot koluna bağlanmış siyah-beyaz bir resim tek yönde hareket ettirilmiştir ve alınan düşük çözünürlüklü görüntüler işlenerek yüksek çözünürlüklü görüntü elde edilmiştir. Düşük çözünürlüklü görüntülerde ne olduğu algılanamayan resim, yüksek çözünürlüklü görüntüde algılanabilir olmuştur.

8.3 İleriye Yönelik Çalışma Önerileri

Çalışılan yöntemde böcek gözlerinin yapısı modellenmiştir. Bu modelde farklı bölgelere yönelmiş gözcüklerin veri topladığı veriler işlenerek bir tane yüksek çözünürlüklü görüntü elde edilmektedir. Çalışmada gözcük modelinin düzlemsel olarak belli kayma adımlarında bölgeyi kapsadığı kabul edilmiştir. Gerçek böcek gözlerinde gözcükler küresel bir yapı üzerinde bulunmaktadır; bölge üzerinde hem düzlemsel hem de açısal kayıklıkları vardır. Dolayısıyla ileriki çalışmada bölgenin küresel bir şekilde taranması denenebilir.

Yapılan çalışmada görüntü sensörü olarak dijital bir kamera kullanılmıştır. Dijital kamera sensorları daha önce de belirtildiği gibi tam sayı piksel değerleri üretmektedir. Pikselleri tam sayı olarak okumak ondalık değerlerin ele alınamamasına neden olmaktadır. Bu da çözünürlük iyileştirmesi sırasında gürültülere neden olmaktadır. İleriki çalışmada dijital kamera kullanmak yerine analog değer alınabilecek bir görüntü sensörü ile çalışılabilir. Böylece yöntemde önerilen döngülü algoritma elektronik bir devre ile gerçekleştirilebilir ve nicemlemeden kaynaklanan hatalar olmadan çözünürlük iyileştirmesi sağlanabilir.

Yapılan çalışmada alt piksel kayma hareketleri PA10-7C robot kolu kullanılarak gerçekleştirilmiştir. Robot kolu hareketi sırasında oluşan titreşim sonuçtaki başarıyı doğrudan etkilemektedir. Bu sebeple robot kontrolü için başka bir yöntem bulunabilir veya alt piksel hareketleri için başka bir sistem tasarlanabilir. Mesela x-y düzlemi üzerinde hareket edebilen bir kızak servo motorlarla daha titreşimsiz olarak kontrol edilerek daha iyi sonuçlar elde edilebilir.

Yapılan alıřmada elde edilen sonucu etkileyen bir faktör de kullanılan ışık kaynağıdır. Deneylerde flüoresan, akkor ve halojen olmak üzere üç tip lamba kullanılmıştır. Bu üç ışık kaynağının da kendine has problemlerinden Bölüm 6.4.5.4’de bahsedilmiştir. İleriki alıřmada bölge üzerine daha homojen ışık yayabilen bir ışık kaynağı kullanılması daha iyi sonuçlar elde edilmesini sağlayacaktır.

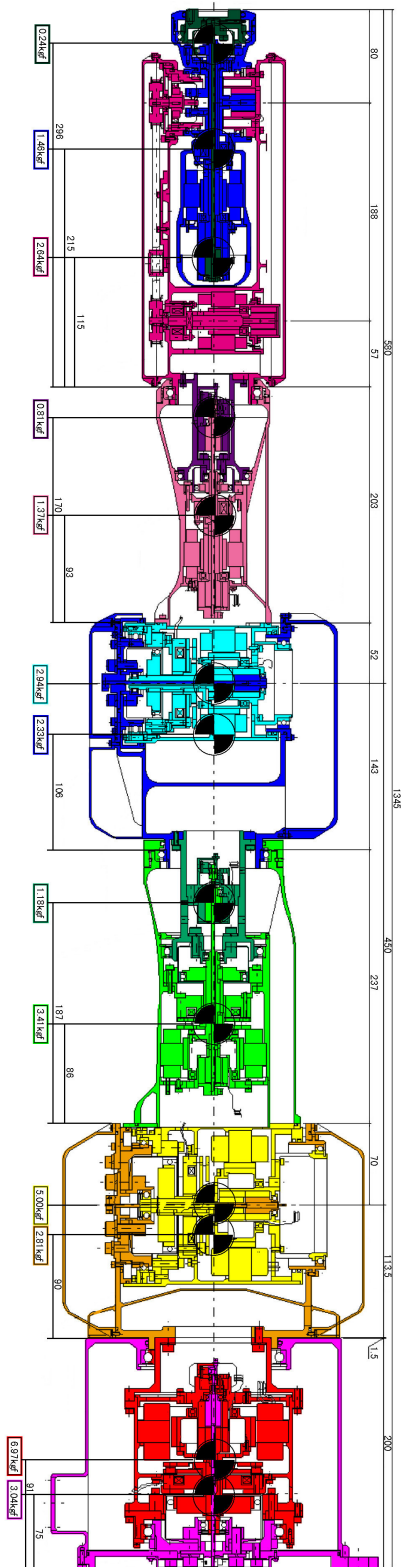
KAYNAKLAR

- [1] **Komatsu, T., Aizawa, K., Igarashi, T. and Saito, T.**, February 1993. Signal-Processing Based Method for Acquiring Very High Resolution Image with Multiple Cameras and its Theoretical Analysis, Proc. Inst. Elec. Eng., vol. **140**, no. 1, pp. 19-25
- [2] **Moon Gi Kang and Chaudhuri S.**, May 2003. Super-resolution Image Reconstruction, Signal Processing Magazine, IEEE, Volume **20**, Issue 3, pp.19 – 20
- [3] **Farsiu, S., Robinson, M.D., Elad, M. and Milanfar P.**, October 2004. Fast and Robust Multiframe Super Resolution, Image Processing, IEEE Transactions, Volume **13**, Issue 10, pp. 1327 – 1344
- [4] **Borman, S., and Stevenson, R.L.**, 1998. Spatial Resolution Enhancement of Low-Resolution Image Sequences. A Comprehensive Review with Directions for Future Research, Lab. Image and Signal Analysis, University of Notre Dame, Tech. Rep.
- [5] **Baker, S. and Kanade, T.**, September 2002. Limits on Super-Resolution and How to Break Them, Volume 24, Issue **9**, pp.1167 – 1183
- [6] **Komatsu, T., Igarashi, T., Aizawa, K. and Saito, T.**, December 1993. Very High Resolution Imaging Scheme with Multiple Different-Aperture Cameras, Sinal Processing: Image Commun., vol. **5**, pp. 511-526,
- [7] **Choi, C.**, April 2004. 'Insect Eyes' can Improve Optical Systems
Ağ üzerinden erişim:
<http://www.washtimes.com/upi-breaking/20040402-015549-5848r.htm>
- [8] **Parmentier, J.**, April 2000. Insect Eye, Micscape Magazine
Ağ üzerinden erişim:
<http://www.microscopy-uk.org.uk/mag/artapr00/inseye.html>
- [9] **Duparre, J. W., Dannberg, P., Schreiber, P., Brauer, A., Nussbaum, P., Heitger, F., Tünnermann, A., Völkel, R., Eisner, M. and Scharf T.**, February 2005. Microoptical Telescope Compund Eye, Optics Express, vol. **13**, no. 3, pp. 889
- [10] **Duparre, J. W., Dannberg, P., Schreiber, P., Brauer, A., Nussbaum, P., Heitger, F. and Tünnermann, A.**, 2004. Ultra-thin Camera Based on Artificial Apposition Compound Eyes, Proc. of 10th Microopt. Conf., W. Karthe,G. D. Khoe, and Y. Kokubun, eds., pages E–2, ISBN: 3-8274-1603-5

- [11] **Chaudhuri, S.**, 2001. Super-Resolution Imaging, Hingham, MA, USA: Kluwer Academic Publishers, p 7.
Ağ üzerinden erişim:
<http://site.ebrary.com/lib/istanbulteknik/Doc?id=10048274&ppg=22>
- [12] **Blackwell, J. A.**, March 2003. CCD Imaging: Q&A
Ağ üzerinden erişim
<http://www.regulusastro.com/regulus/papers/ccdfaq/index.html>
- [13] **Hunt, B. R.**, 1995. Super-resolution of Images : Algorithms, Principles and Performance, International Journal of Imaging Systems and Technology, vol. 6, pp. 297– 304,
- [14] **Parker, J. A.**, Kenyon, R. V., and Troxel, D. E., 1983. Comparison of Interpolating Methods for Image Resampling, IEEE Trans. on Medical Imaging, vol. 2, pp. 31– 39
- [15] **Park, S. C., Park, M. K. and Kang, M. G.**, May 2003. Super-Resolution Image Reconstruction, IEEE Signal Processing Magazine, p 21-36
- [16] **Gevrekci, M. and Gunturk, B.K.**, September 2005. Image Acquisition Modeling for Super-resolution Reconstruction, Image Processing, 2005. ICIP 2005. IEEE International Conference, Volume 2, 11-14, Page(s):II - 1058-61
- [17] **Hu He and Kondi, L.P.**, September 2005. A Regularization Framework for Joint Blur Estimation and Super-resolution of Video Sequences, Image Processing, 2005. ICIP 2005. IEEE International Conference, Volume 3, Page(s):III - 329-32
- [18] **Eren, P. E., Sezan, M. I. and Tekalp A M.**, October 1997. Robust, Object Based High Resolution Image Reconstruction from Low Resolution Video, IEEE Trans. on Image Processing, vol. 6, pp. 1446– 1451
- [19] **Clark, J. J., Palmer, M R. and Laurence P.D.**, 1985. A Transformation Method for the Reconstruction of Functions from Nonuniformly Spaced Samples, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, pp. 1151-1165
- [20] **Altunbasak, Y., Patti, A.J. and Mersereau, R.M.**, April 2002. Super-Resolution Still and Video Reconstruction from MPEG-coded Video, Circuits and Systems for Video Technology, IEEE Transactions on Volume 12, Issue 4, Page(s):217 – 226
- [21] **Borman, S. and Stevenson, R.L.**, 1999. Super-resolution from image sequences-A review, in Proc. 1998 Midwest Symp. Circuits and Systems, pp. 374-378
- [22] **Tsai, R. Y., and Huang, T. S.**, 1984. Multiframe Image Restoration and Registration, Advances in Computer Vision and Image Processing, JAI Press Inc., pp. 317– 339.
- [23] **Tekalp, A.M., Ozkan, M.K. and Sezan, M.I.**, March 1992. High-resolution Image Reconstruction from Lower-Resolution Image Sequences and Space-Varying Image Restoration, Acoustics, Speech, and Signal

- Processing, 1992. ICASSP-92., 1992 IEEE International Conference on Volume **3**, Page(s):169 - 172
- [24] **Kim, S. P. and Su, W.Y.**, October 1993. Recursive High-Resolution Reconstruction of Blurred Multiframe Images, IEEE Trans. on Image Processing, vol. **2**, pp. 534– 539
 - [25] **Papoulis, A.**, November 1977. Generalized Sampling Theorem, IEEE Trans. On Circuits and Systems, vol. **24**, pp. 652– 654
 - [26] **Brown, J. L.**, February 1981. Multi-channel Sampling of Low Pass Signals, IEEE Trans. on Circuits and Systems, vol. **CAS-28**, no. 2, pp. 101– 106
 - [27] **Irani, M. and Peleg, S.**, May 1991. Improving resolution by image registration, CVGIP: Graphical Models and Image Proc., vol. **53**, pp. 231-239
 - [28] **Stark, H. and Oskoui, P.**, 1989. High resolution image recovery from image-plane arrays, using convex projections, J. Opt. Soc. Am. A, vol. **6**, pp. 1715-1726
 - [29] **Schultz, R.R. and Stevenson, R.L.**, June 1996. Extraction of High-Resolution Frames from Video Sequences, Image Processing, IEEE Transactions on Volume **5**, Issue 6, Page(s):996 – 1011
 - [30] **Tom, B.C. and Katsaggelos, A.K.**, October 1995. Reconstruction of a High-Resolution Image by Simultaneous Registration, Restoration, and Interpolation of Low-Resolution Images, Proc. 1995 IEEE Int. Conf. Image Processing, vol. **2**, Washington, DC, pp. 539-542
 - [31] **Kimball, J. W.**, December 2004. The Compound Eye
Ağ üzerinden erişim
<http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/C/CompoundEye.html>
 - [32] **BioMEDIA Associates**, 2004. What Animal has a More Sophisticated Eye, Octopus or Insect?
Ağ üzerinden erişim
<http://ebiomedica.com/gall/eyes/octopus-insect.html>

EK 1: PA10-7C Robot Kolunun Mekanik Yapısı

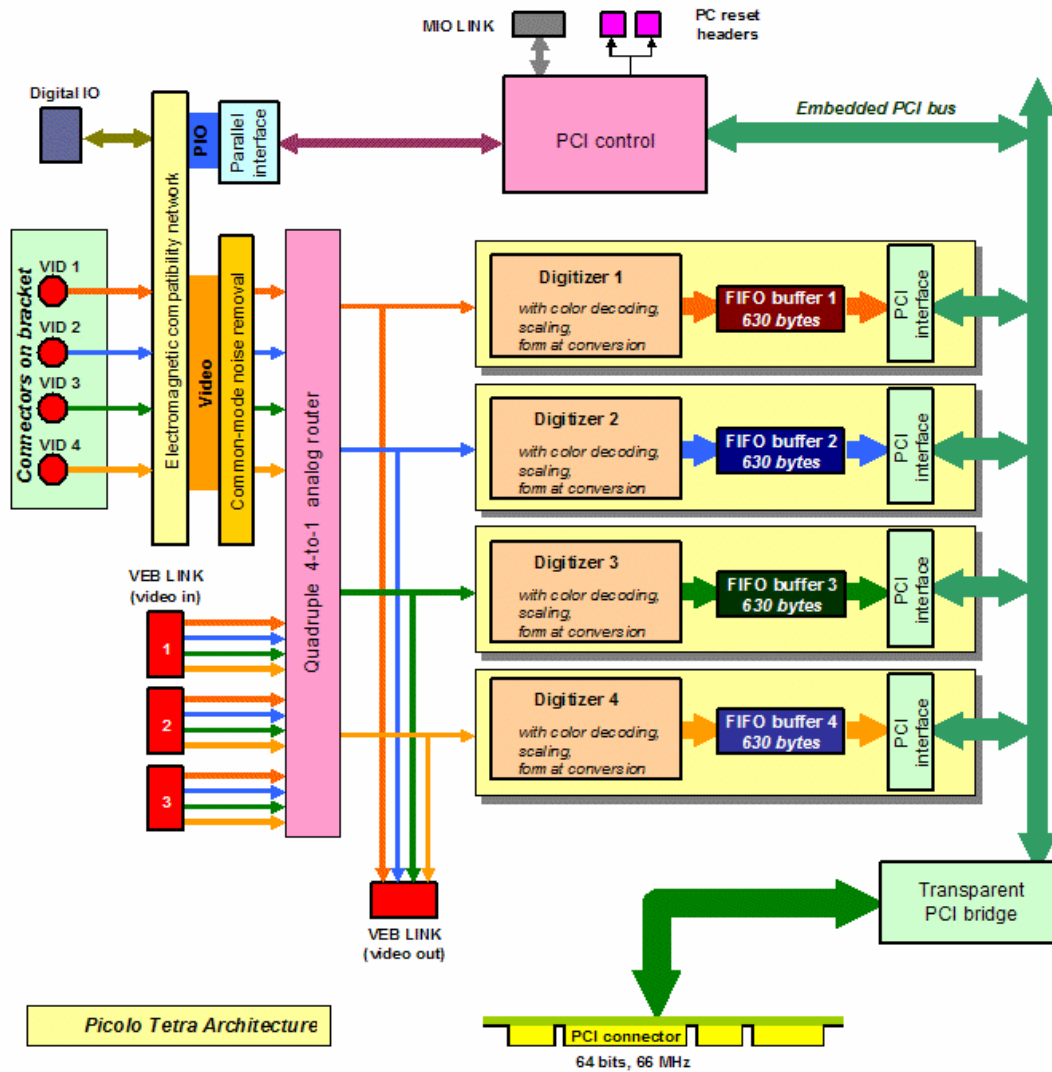


EK 2: JAI CV-S3200 Marka Kameranın Teknik Özellikleri

Specifications	PAL	NTSC
Scanning system	625 lines 25 frames/sec. 2:1 interlaced	525 lines 30 frames/sec. 2:1 interlaced
CCD sensor	Color 1/2" and 1/3" IT EXview HAD CCD™	
Sensing area 1/2" (CV-S3200) 1/3" (CV-S3300)	6.6 (h) x 4.8 (v) mm 4.9 (h) x 3.7 (v) mm	
Effective pixels	752 (h) x 582 (v)	768 (h) x 494 (v)
Cell size 1/2" (CV-S3200) 1/3" (CV-S3300)	8.6 (h) x 8.3 (v) μ m 6.5 (h) x 6.25 (v) μ m	8.4 (h) x 9.8 (v) μ m 6.35 (h) x 7.4 (v) μ m
Sensitivity on sensor	0.09 Lux (AGC on, 50% video, 2856K) 0.025 Lux (with 4 fields integration)	
S/N ratio	>50 dB (AGC off, Gamma 1.0)	
Video output	1.0 Vpp, 75 Ohm. VBS and Y/C	
Gain mode	Fixed, AGC 12 dB, AGC 24 dB, Manual	
Gain range Fixed AGC 12 dB AGC 24 dB Manual (on-screen menu) Manual (RS 232C)	0 dB 0 to +12 dB 0 to +24 dB 0 to +24 dB 0 to +24 dB	
Gamma	0.45 - 0.6 - 1	
White balance	3200K, 4600K, 5600K, Auto, Manual, One-push	
Iris control	Direct drive or video controlled	
Iris direct drive	$\pm$ Drive and $\pm$ Damp. Level adjust	
Iris video output	1Vpp video and +9V DC	
Back light compensation	Off, Pattern 1, pattern 2, Pattern 3	
Synchronization	Internal X-tal or external lock	
External lock input	1 Vpp. VBS or composite sync.	
Scanning reversal	Off, On (horizontally mirrored)	
Shutter	Off (1/50 PAL, 1/60 NTSC), 1/125, 1/250, 1/500, 1/1000, 1/2000, 1/4000, 1/10,000 sec., CCD Iris	
Shutter flickerless	1/120 sec.	1/100 sec.
Extended sensitivity (Integration)	Off, Sensitive 1, Sensitive 2, Exposure time 2 to 64 fields	
Camera setup	On-screen menu or RS 232C	
Functions for on-screen menu and RS 232C	Shutter, Integration, Gain, Gamma, Back light, Scan reversal, White balance, Iris level control	
Functions RS 232C only	Y/C reversal	
Operating temperature	-10°C to +50°C*	
Humidity	20-80% non-condensing	
Power	12V DC $\pm$ 10%, 3W	
Lens mount	CS-mount	
Flange back adjust	$\pm$ 1 mm	
Dimensions	45 x 55 x 110.2 mm (HxWxD)	
Weight	350 g	

* Recommended temp.: 0°C to 40°C

EK 3: Euresys Picolo Tetra Mimarisi



EK 4: Geliştirilen Kütüphaneler

A) Matris Kütüphanesi

class BaseMatrix

— Özellikleri

- `gsl_matrix *mdata;`
GSL kütüphanesi tarafından tanımlanmış tiptir. Matris verisini işaret eder.

— Metotları

- `BaseMatrix();`
Yeni bir boş matris nesnesi oluşturur
- `BaseMatrix(int m,int n);`
m satırlı ve n sütunlu yeni bir matris nesnesi oluşturur
- `BaseMatrix(gsl_matrix *matrix);`
matrix işaretçisinin gösterdiği veriden yeni bir matris nesnesi oluşturur
- `BaseMatrix(BaseMatrix *matrix);`
matrix nesnesinin gösterdiği veriden yeni bir matris nesnesi oluşturur
- `gsl_matrix* getData();`
Nesnenin matris verisinin işaretçisini döndürür
- `int size1();`
Nesnenin satır boyunu döndürür
- `int size2();`
Nesnenin sütun boyunu döndürür
- `double get(int m,int n);`
Nesnenin m. satır ve n. sütunundaki veriyi döndürür. Satır ve sütunlar sıfır indeksinden başlamaktadır
- `double get_likeMatlab(int m,int n);`
Nesnenin m. satır ve n. sütunundaki veriyi döndürür. Satır ve sütunlar Matlab programında olduğu gibi bir indeksinden başlamaktadır
- `BaseMatrix* get(int k1,int k2,int n1,int n2);`
Nesnenin k1. satır ve k2. sütunundan başlayarak n1 satırlık ve n2 sütunluk yeni bir alt matris nesnesi döndürür. Satır ve sütunlar sıfır indeksinden başlamaktadır
- `BaseMatrix* get_likeMatlab(int k1,int k2,int n1,int n2);`
Nesnenin k1. satır ve k2. sütunundan başlayarak n1 satırlık ve n2 sütunluk yeni bir alt matris nesnesi döndürür. Satır ve sütunlar Matlab programında olduğu gibi bir indeksinden başlamaktadır

- `void set(int m,int n,double value);`
Nesnenin m. satır ve n. sütununa value verisini yazar. Satır ve sütunlar sıfır indeksinden başlamaktadır
- `void set_likeMatlab(int,int,double);`
Nesnenin m. satır ve n. sütununa value verisini yazar. Satır ve sütunlar Matlab programında olduğu gibi bir indeksinden başlamaktadır
- `void set(int k1,int k2,int n1,int n2,double value);`
Nesnenin k1. satır ve k2. sütunundan başlayarak n1 satırlık ve n2 sütunluk alt matrise value değerini yazar. Satır ve sütunlar sıfır indeksinden başlamaktadır
- `void set(int m,int n,BaseMatrix *matrix);`
Nesnenin m. satır ve n. sütunundan başlayarak matrix verisini alt matris olarak yazar. Satır ve sütunlar sıfır indeksinden başlamaktadır
- `void set_likeMatlab(int m,int n,BaseMatrix *matrix);`
Nesnenin m. satır ve n. sütunundan başlayarak matrix verisini alt matris olarak yazar. Satır ve sütunlar Matlab programında olduğu gibi bir indeksinden başlamaktadır
- `void setall(double value);`
Nesnenin tüm elemanlarına value değerini yazar
- `void setzero();`
Nesnenin tüm elemanlarına sıfır değerini yazar
- `void setallrow(int m,double value);`
Nesnenin m. satırına value değerini yazar
- `void setallcol(int,double);`
Nesnenin n. sütununa value değerini yazar
- `void add(BaseMatrix *matrix);`
Nesneyle matrix verisini toplar
- `void sub(BaseMatrix *matrix);`
Nesneden matrix verisini çıkarır
- `void mul(double value);`
Nesnenin tüm elemanlarını value ile çarpar
- `void mulcol(int n,double value);`
Nesnenin n. sütununu value ile çarpar
- `void sin();`
Nesnenin tüm elemanlarını sinüs değerleriyle değiştir
- `void cos();`
Nesnenin tüm elemanlarını kosinüs değerleriyle değiştir
- `BaseMatrix* transpose();`
Nesnenin transpozisini alarak yeni bir matris nesnesi döndürür

- `BaseMatrix* multiply(int trans1, BaseMatrix *matrix, int trans2);`
Nesneyi matrix nesnesiyle çarpıp ve yeni bir matris nesnesi döndürür.
Trans1 ve trans2 parametreleri çarpım işleminden önce matris nesnelerinin transpozelerinin alınıp alınmayacağı ile ilgilidir

- `void print();`
Nesne elemanlarını ekrana yazdırır
- `~BaseMatrix();`
Nesneyi bellekten siler

class BaseVector

— Özellikleri

- `gsl_vector *vdata;`
GSL kütüphanesi tarafından tanımlanmış tiptir. Vektör verisini işaret eder.

— Metotları

- `BaseVector(int i);`
i boyutunda yeni bir boş vektör nesnesi oluşturur
- `BaseVector(double *data, int i);`
Elemanları data olan i boyutunda yeni bir boş vektör nesnesi oluşturur
- `double get(int i);`
Nesnenin i. elemanını döndürür
- `void setall(double value);`
Nesnenin tüm elemanlarına value yazar
- `int size();`
Nesnenin boyunu döndürür
- `~BaseVector();`
Nesneyi bellekten siler

class GeneralMatrix(BaseMatrix)

— Metotları

- `GeneralMatrix(int m, int n);`
m satırlı ve n sütunlu yeni bir matris nesnesi oluşturur
- `GeneralMatrix(double *data, int m, int n);`
Elemanları data olan m satırlı ve n sütunlu yeni bir matris nesnesi oluşturur
- `GeneralMatrix(gsl_matrix *matrix);`
matrix işaretçisinin gösterdiği veriden yeni bir matris nesnesi oluşturur
- `GeneralMatrix(BaseMatrix *matrix);`
matrix işaretçisinin gösterdiği veriden yeni bir matris nesnesi oluşturur

- GeneralMatrix* inverse();
Nesnenin ters matrisini hesaplayarak yeni bir matris nesnesi döndürür

class SquareMatrix(GeneralMatrix)

— Metotları

- SquareMatrix(int size);
size satırlı ve size sütunlu yeni bir matris nesnesi oluşturur
- SquareMatrix(double *data,int size);
Elemanları data olan size satırlı ve size sütunlu yeni bir matris nesnesi oluşturur
- SquareMatrix(BaseMatrix *matrix);
matrix işaretçisinin gösterdiği veriden yeni bir matris nesnesi oluşturur
- SquareMatrix* inverse();
Nesnenin ters matrisini hesaplayarak yeni bir matris nesnesi döndürür
- SquareMatrix* transpose();
Nesnenin transpozisini hesaplayarak yeni bir matris nesnesi döndürür

class DiagonalMatrix(SquareMatrix)

— Metotları

- DiagonalMatrix(double *data,int size,int k);
Köşegenleri data olan size+k boyutunda yeni bir matris nesnesi oluşturur
- DiagonalMatrix(BaseVector *vector,int k);
Köşegenleri vektör nesnesinin elemanları olan size+k boyutunda yeni bir matris nesnesi oluşturur

class IdentityMatrix(SquareMatrix)

— Metotları

- IdentityMatrix(int size);
Köşegenlerini 1 olan size satırlı ve size sütunlu yeni bir matris nesnesi oluşturur
- IdentityMatrix(BaseMatrix *matrix);
matrix işaretçisinin gösterdiği veriden yeni bir matris nesnesi oluşturur

class ZerosMatrix(BaseMatrix)

— Metotları

- ZerosMatrix(int m,int n);
Elemanları sıfır olan m satırlı ve n sütunlu bir matris oluşturur

class OnesMatrix(BaseMatrix)

— Metotları

- `OnesMatrix(int m,int n);`
Elemanları 1 olan m satırlı ve n sütunlu bir matris oluşturur

B) Ters Kinematik Kütüphanesi

class InverseKinematic

— Özellikleri

- `int s;`
Robotun serbestlik derecesidir
- `GeneralMatrix *Vt;`
Uç noktanın açısal ve lineer hız vektörüdür. 6×1 boyutunda bir matris ile ifade edilir. İlk 3 eleman referans koordinat eksenindeki açısal hızını, son 3 eleman ise referans koordinat eksenindeki doğrusal hızını tanımlar.
- `ZerosMatrix *x0;`
Başlangıç koşuluna ilişkin link eksen takımının x bileşenlerini tanımlar. $3 \times s$ boyutunda bir matris ile ifade edilir
- `ZerosMatrix *y0;`
Başlangıç koşuluna ilişkin link eksen takımının y bileşenlerini tanımlar. $3 \times s$ boyutunda bir matris ile ifade edilir
- `ZerosMatrix *z0;`
Başlangıç koşuluna ilişkin link eksen takımının z bileşenlerini tanımlar. $3 \times s$ boyutunda bir matris ile ifade edilir
- `GeneralMatrix *x;`
Çalışma anındaki link eksen takımının x bileşenlerini tanımlar. $3 \times s$ boyutunda bir matris ile ifade edilir
- `GeneralMatrix *y;`
Çalışma anındaki link eksen takımının y bileşenlerini tanımlar. $3 \times s$ boyutunda bir matris ile ifade edilir
- `GeneralMatrix *z;`
Çalışma anındaki link eksen takımının z bileşenlerini tanımlar. $3 \times s$ boyutunda bir matris ile ifade edilir
- `int n;`
Robot kolunun her eklemi için tutulan açısal ve lineer hız matrisinin boyunu tutar. $n = 6*s$ değerini alır
- `GeneralMatrix *theta;`
Robot kolu eklemlerinin çalışma anındaki açısal değerini tutar. $s \times 1$ boyutunda bir matris ile ifade edilir

- GeneralMatrix *dtheta;
Robot kolu eklemlerinin çalışma anındaki son yaptığı açısal hareket değerlerini tutar. $s \times 1$ boyutunda bir matris ile ifade edilir
- ZerosMatrix *theta_d;
Robot kolu eklemlerinin çalışma anındaki açısal hız değerlerini tutar. $s \times 1$ boyutunda bir matris ile ifade edilir
- ZerosMatrix *H;
Robot kolu eklemlerinin dönme eksen matrisidir. $6s \times s$ boyutunda bir matris ile ifade edilir
- IdentityMatrix *Phi;
Robot kolu linklerinin propagasyon matrisidir. $6s \times 6s$ boyutunda bir matris ile ifade edilir
- ZerosMatrix *sigmat;
Robot kolunun uç nokta propagasyon matrisidir. $6 \times 6s$ boyutunda bir matris ile ifade edilir

— Metotları

- InverseKinematic(int pS, double vtdata[]);
pS serbestlik dereceli ve vtdata hareketi için eklem açıları değerlerini hesaplayacak yeni bir ters kinematik nesnesi yaratır
- void init(double angles[]);
Hesaplamalara başlamadan önceki başlangıç açılarını belirler
- void updateAngleInfo(double angles[]);
Robottan alınan gerçek anlık açı değerlerini belirler. Böylece hesaplar yapılırken gerçek değerler üzerinde çalışılır
- void updateVtData(double vtdata[]);
Çalışma anında yeni bir vtdata hareket bilgisini yükler
- int calculateNext();
Robota yollanacak eklem açı bilgilerini hesaplar
- GeneralMatrix *getLMatrix();
Robotun link vektörlerini döndürür
- void loadHMatrix();
Dönme eksen matrisini yeniden hesaplar
- void loadPhiMatrix(GeneralMatrix *L);
Link vektörlerine göre propagasyon matrisini yeniden hesaplar
- void loadSigmatMatrix(GeneralMatrix *L);
Link vektörlerine göre uç nokta propagasyon matrisini yeniden hesaplar

- GeneralMatrix *getJMatrix();
Ters kinematik hesabında kullanılacak Jakobyeni hesaplar ve döndürür.
Burada hesaplanan Jakobyen = $\sigma \Phi H$ dir.

- void loadTheta_dMatrix(GenericMatrix *J);
Robota yollanacak anlık hız bilgilerini yeniden hesaplar
- void loadXYZMatrices(GenericMatrix*);
Link eksen takımının x,y,z bileşenlerini yeniden hesaplar
- BaseMatrix* getX();
Link eksen takımının x bileşenini döndürür
- BaseMatrix* getY();
Link eksen takımının y bileşenini döndürür
- BaseMatrix* getZ();
Link eksen takımının z bileşenini döndürür
- BaseMatrix* getTheta_d();
Robot kolunun her ekleminin anlık hız bilgisini döndürür
- BaseMatrix* getRt();
Robot kolunun uç noktasının koordinat ekseninin x,y,z bileşenlerindeki değerleri döndürür
- BaseMatrix* getTipPoint();
Robot kolunun uç noktasının anlık koordinatını döndürür
- void print();
Anlık x, y, z matris değerlerini ekrana basar
- void printTheta_d();
Robot eklemlerine yollanacak anlık açısal hızları ekrana basar
- ~InverseKinematic();
Nesneyi bellekten siler

C) Robot Kütüphanesi

class Robot

— Özellikleri

- int devicefd;
Bilgisayarın ISA slotuna bağlı ArcNet kartını sürmek için açılan dosya işaretçisidir
- int jcount;
Robotun eklemler sayıdır. Çalışmada kullanılan PA10-7C robotunda bu değer yedidir

- Joint *joints;
Robotun her eklemi için oluşturulan Joint nesne dizisi işaretçisidir

— Metotları

- Robot(int count);
count kadar eklemi olan yeni bir robot nesnesi yaratır
- int isDeviceOpened();
ArcNet kartının açılıp açılmadığı bilgisini döndürür
- void getStatus();
Robotun eklemlerinin açı ve torque bilgilerini okur
- void reset();
Robotu dik pozisyona getirir
- void init();
Robotu çözünürlük iyileştirme çalışmasındaki başlangıç konumuna getirir
- void triangle(int joint,int angle,double speed,int count);
Belirtilen eklemi belirtilen açı miktarında artı ve eksi yönde belirtilen hızda ve belirtilen sayıda hareket ettirir
- void sinus(int joint,double A,double T);
Belirtilen ekleme $A \sin\left(\frac{2\pi}{T}t\right)$ hareketini yaptırır
- void moveJoint(int joint,int angle,double speed);
Belirtilen eklemi belirtilen açıya belirtilen hız ile hareket ettirir
- void moveJoint(int joint,double speed);
Belirtilen eklemi belirtilen hızda hareket ettirir
- void releaseJoint(int joint);
Belirtilen eklemin mekanik frenlerini açar
- void releaseJoints();
Tüm eklemlerin mekanik frenlerini açar
- string scan(double stepSize,int stepCount,string mode);
Robotu ters kinematik hesaplarıyla bulunduğu konumda belirtilen adım miktarı ve sayısı parametreleriyle hareket ettirir. *mode* parametresi hareketin yönünü ve şeklini belirtmeye yarar
- short checkMovingNeed();
Robotun istenen bir konuma gelip gelmediğini kontrol eder
- void move();
Eklemler için koşullandırılan hareket parametrelerini istenen konuma gelene kadar robota yollar
- void move(GeneralMatrix *refVector);
Robotun uç noktasını koordinat ekseninde belirtilen noktaya getirir

- void send();
ISA kartı üzerinden robota hareket komutlarını yollar
- void showInfo();
Robot eklemlerinin o andaki konumlarını ekrana yazar
- ~Robot();
Robot nesnesini bellekten siler

class Joint

— Özellikleri

- int index;
Eklem robot kolunda kaçınıcı sırada olduğu bilgisini tutar. En alttaki eklem sıfır indeksinden başlamaktadır
- short angle_control;
Robota yollanacak eklem ilişkin hareket parametresini oluştururken istenen açıda olup olmadığı kontrolünün yapılıp yapılmayacağını belirtir
- short angle_limit_control;
Robota yollanacak eklem ilişkin hareket parametresini oluştururken açılar için tanımlanmış yazılım limitlerinin dikkate alınıp alınmayacağını belirtir
- jstatus curStatus;
Eklem anlık açı ve tork bilgisini tutar
- int refAngle;
Eklem gelmesini istediğimiz açı değerini tutar
- int refTorque;
Eklem hareketi sırasında yollanacak tork bilgisini tutar
- double refSpeed;
Eklem hareketi sırasında yollanacak açısal hız bilgisini tutar
- char refStatus;
Eklem tork modunda mı yoksa hız modunda mı çalışacağını belirtir
- short moving_need;
Eklem belirtilen açıya gelip gelmediği bilgisini tutar

— Metotları

- Joint();
Yeni bir eklem nesnesi yaratır
- void setIndex(int i);
Eklem robot kolundaki sırasını tanımlar
- void enableAngleControl();
Hareket bilgisi oluştururken istenen açıya gelindi kontrolünün yapılmasını sağlar

- void disableAngleControl();
Hareket bilgisi oluştururken istenen açıya gelindi kontrolünün yapılmamasını sağlar
- void enableAngleLimit();
Hareket bilgisi oluştururken sınır açıya gelindi kontrolünün yapılmasını sağlar
- void disableAngleLimit();
Hareket bilgisi oluştururken sınır açıya gelindi kontrolünün yapılmamasını sağlar
- void setAngle(int angle);
Eklem gitmesi istenen açı değerini tanımlar
- void setTorque(int torque);
Eklem hareketi sırasında yollanacak tork bilgisini tanımlar
- void setSpeed(double speed);
Eklem hareketi sırasında yollanacak açısal hız bilgisini tanımlar
- ucommand getUserCommand();
Robota yollanacak eklem hareket bilgisini döndürür
- void setStatus(jstatus s);
Robotun çalışma anındaki tork ve açısal hız bilgisini tanımlar
- double getCurrentAngle();
Eklem o andaki açı değerini döndürür
- int getCurrentTorque();
Eklem o andaki tork değerini döndürür
- short checkMovingNeed();
Eklem istenen açıya gelip gelmediğini kontrol eder
- void showInfo();
Eklem açı ve tork bilgisini ekrana yazar

D) Zamanlayıcı Kütüphanesi

class Timer

— Özellikleri

- struct timeval start_tv;;
Zamanlayıcının başlatıldığı zamanını tutar
- struct timeval end_tv;;
Zamanlayıcının durdurulduğu zamanını tutar
- int running;
Zamanlayıcının çalışıp çalışmadığı bilgisini tutar

— Metotları

- `Timer();`
Yeni bir zamanlayıcı nesnesi yaratır
- `void start();`
Zamanlayıcıyı başlatır
- `void end();`
Zamanlayıcıyı durdurur
- `long elapsedMicrosec();`
Zamanlayıcının başlangıç ve bitiş zamanları arasındaki süre farkını mikro saniye cinsinden döndürür
- `long elapsedMilisec();`
Zamanlayıcının başlangıç ve bitiş zamanları arasındaki süre farkını milisaniye cinsinden döndürür
- `void waitTo(int milisecond);`
Zamanlayıcının başlangıcından itibaren belirtilen milisaniye kadar süre geçene kadar bekler
- `void wait(int milisecond);`
Belirtilen milisaniye kadar süre geçene kadar bekler
- `string startTimeToString();`
Zamanlayıcının başlangıç zamanını ekrana basar
- `string endTimeToString();`
Zamanlayıcının durdurulma zamanını ekrana basar

E) Görüntü Kütüphanesi

class Image

— Özellikleri

- `IplImage* data;`
OpenCV kütüphanesi tarafından tanımlanmış tiptir. Görüntü verisini işaret eder.
- `CvMat* matrix_data;`
OpenCV kütüphanesi tarafından tanımlanmış tiptir. Görüntü verisini ondalıklı sayılardan oluşan bir matris şeklinde işaret eder.

— Metotları

- `Image();`
Yeni bir boş görüntü nesnesi oluşturur
- `Image(const char *filename);`
Dosya sisteminde filename ile belirtilen resim dosyasından yeni bir görüntü nesnesi oluşturur

- `Image(CvSize size,int depth,int channel,int bgcolor);`
size boyunda, depth (8bit, 16bit, v.b.) piksel derinliğinde, channel kanallı ve arka fon rengi bgcolor olan yeni bir görüntü nesnesi oluşturur
- `Image(IplImage *image);`
image işaretçisinin gösterdiği veriden yeni bir görüntü nesnesi oluşturur
- `Image(CvSize size,uchar *buffer);`
size boyunda ve buffer işaretçisinin gösterdiği verilerden oluşan yeni bir görüntü nesnesi oluşturur
- `void createMatrixData(CvSize size);`
size boyunda görüntü verilerini tutacak yeni bir matris nesnesi oluşturur
- `void setMatrixValue(int x,int y,double value);`
Matrisin x. satır ve y. sütununa value değerini yükler
- `double getMatrixValue(int x,int y);`
Matrisin x. satır ve y. sütununa ilişkin veriyi geri döndürür
- `IplImage* getData();`
Görüntü verisinin işaretçisini döndürür
- `CvSize getSize();`
Görüntünün boyunu döndürür
- `int getPixelStep();`
Görüntü verisindeki her piksel verisi arasındaki bellek aralığını byte cinsinden döndürür
- `uchar* getPixelPointer(int x,int y);`
Görüntü verisinde x. satır ve y. sütuna ilişkin piksel verisinin işaretçisini döndürür
- `uchar getPixel(int x,int y);`
Görüntüdeki x. satır ve y. sütuna ilişkin piksel değerini döndürür
- `void putPixel(int x,int y,int value);`
Görüntüdeki x. satır ve y. sütuna ilişkin piksele value değerini yükler
- `void show();`
Görüntüyü yeni bir pencerede gösterir
- `Image* crop(int x,int y,int width,int height);`
Görüntünün x. satır ve y. sütun pikselinden başlayarak width genişliğinde ve height yüksekliğindeki bir bölgeden yeni bir görüntü nesnesi oluşturur ve döndürür
- `void paste(int x,int y,Image *image);`
Görüntünün x. satır ve y. sütun pikselinden başlayarak image işaretçisinin gösterdiği görüntü piksellerini yapıştırır
- `void fill(int value);`
Görüntüdeki tüm piksellere value yükler

- void convertToBinary(int threshold);
Görüntüyü threshold değerine göre siyah-beyaz hale getirir
- void save(const char *filename);
Görüntüyü filename adı ile dosya sisteminde saklar
- void exportImage(const char *filename);
Nesneyi filename adı ile dışa aktarır
- void marshall(const char* filename);
filename adındaki dosya verisinden yeni bir görüntü nesnesi yükler
- ~Image();
Nesneyi bellekten siler

F) Resim Sayısallaştırıcı Kütüphanesi

class FrameGrabber

— Özellikleri

- MCHANNEL hChannel;
Resim sayısallaştırıcıda oluşturulan kanal işaretçisini tutar
- MCHANNEL hSurfaces[NSURFACES];
Yakalanan görüntülerin işleneceği yüzeyleri işaret eder
- BYTE *pImages[NSURFACES];
Yakalanan görüntülerin tutulacağı tampon bellekleri işaret eder
- MCSTATUS status;
İklendirme (initilization) adımları sırasında oluşan hata durumunu tutar
- string hError;
İklendirme (initilization) adımları sırasında oluşan hataları tutar
- int hDriverIndex;
Resim sayısallaştırıcının sürücü indeksini tutar
- int hConnector;
Kameranin resim sayısallaştırıcının kaçınıcı konektörüne bağlandığı bilgisini tutar
- int hCamera;
Kameradan alınacak görüntünün video standardı bilgisini tutar (PAL veya NTSC)
- int hColorFormat;
Kameradan alınacak görüntünün renk biçimi bilgisini (Y8, RGB24, v.b.) tutar
- int hGrabField;
Kameradan alınacak görüntünün kavrama alanı (örn. tüm kare) bilgisini tutar

- `int hSizeX;`
Kameradan alınacak görüntünün piksel cinsinden genişliğini tutar
- `int hSizeY;`
Yakalanan görüntünün piksel cinsinden yüksekliğini tutar
- `int hGrabCount;`
Yakalama işlemi başlatıldığında arka arkaya kaç tane görüntü yakalanacağı bilgisini tutar. -1 değeri sürekli görüntü yakalamasını sağlar

— Metotları

- `FrameGrabber();`
Yeni bir resim sayısallaştırıcısı nesnesi oluşturur
- `int initialize();`
Resim sayısallaştırıcısının ön tanımlı değerlere göre ilklendirilmesini sağlar
- `int openDriver();`
Sürücünün açılmasını sağlar, geriye başarımlı durumu döndürür
- `int createChannel();`
Resim sayısallaştırıcısının yeni bir kanal oluşturmasını sağlar, geriye başarımlı durumu döndürür
- `int setDriverIndex();`
Resim sayısallaştırıcısının sürücü indeksini yükler, geriye başarımlı durumu döndürür
- `int setVideoConnector();`
Kameranin hangi konektöre bağlı olduğunu ayarlar, geriye başarımlı durumu döndürür
- `int setVideoStandart();`
Kameradan alınacak verinin video standardını ayarlar, geriye başarımlı durumu döndürür
- `int setColorFormat();`
Kameradan alınacak verinin renk biçimini ayarlar, geriye başarımlı durumu döndürür
- `int setGrabField();`
Kameradan alınacak verinin kavrama alanını ayarlar, geriye başarımlı durumu döndürür
- `int getImageSizeX();`
Kameradan alınacak görüntünün piksel cinsinden genişlik bilgisini alır, geriye başarımlı durumu döndürür
- `int getImageSizeY();`
Kameradan alınacak görüntünün piksel cinsinden yükseklik bilgisini alır, geriye başarımlı durumu döndürür

- `int setGrabCount();`
Görüntü yakalama işlemi başlatıldığında arka arkaya kaç tane görüntü yakalanacağını ayarlar, geriye başarımlı durumu döndürür
- `int setGain(int);`
Kameradan alınacak görüntülerdeki kazanç oranını ayarlar (ön tanımlı değeri 1000'dir, 1100 %110 kazanç anlamına gelmektedir), geriye başarımlı durumu döndürür
- `int getGain();`
Ayarlanan kazanç değerini döndürür
- `int registerCallback();`
Resim sayısallaştırıcının görüntü yakalandıktan sonra tetikleyeceği fonksiyonu ayarlar, geriye başarımlı durumu döndürür
- `int enableSurfaceProcessingSignals();`
Görüntü yakalama sırasındaki yüzey işleme sinyallerini etkinleştirir, geriye başarımlı durumu döndürür
- `int enableAcquisitionFailureSignals();`
Görüntü yakalama işlemi sırasında meydana gelecek hata sinyallerini etkinleştirir, geriye başarımlı durumu döndürür
- `MCSTATUS createSurfaces();`
Yakalanan görüntülerin işleneceği yüzeyleri ve tampon belleği oluşturur, geriye başarımlı durumu döndürür
- `void setError(string);`
İklendirme adımları sırasında ortaya çıkacak hata mesajını yükler
- `int isStatusOk();`
İklendirme adımlarında hata olup olmadığı bilgisini döndürür
- `int startAcquisition();`
Kameradan görüntü yakalama işlemini başlatır
- `void endAcquisition();`
Kameradan görüntü yakalama işlemini durdurur
- `int uninitialize();`
Resim sayısallaştırıcı için ayrılan bellekleri siler
- `~FrameGrabber();`
Nesneyi bellekten siler

G) Soket Kütüphanesi

class Socket

— Özellikleri

- `int sockfd;`
Açılan socketin işaretçisini tutar
- `struct sockaddr_in serv_addr;`
Sunucunun adresini tutar

— Metotları

- `Socket();`
Yeni bir socket nesnesi yaratır
- `Socket(int sockfd);`
Belirtilen socket üzerinden yeni bir socket nesnesi yaratır
- `int isSocketOpened();`
Socketin açılıp açılmadığını kontrol eder
- `int getSockfd();`
Socket işaretçisini döndürür
- `string read(long size);`
Sokete gelen verinin *size* kadar byte'ını okur ve döndürür
- `string readall();`
Sokete gelen verinin tamamını okur ve döndürür
- `~Socket();`
Soketi kapatarak nesneyi bellekten siler

class Server

— Metotları

- `Server();`
Yeni bir sunucu socket nesnesi yaratır
- `Server(int sockfd);`
Belirtilen socket üzerinden yeni bir sunucu socket nesnesi yaratır
- `void listen();`
Sunucu socketini dinlemeye başlar
- `void wait(void (*processFunction)(int));`
Sunucu socketine yeni bir paket gelmesini bekler. Paket geldiğinde *processFunction* fonksiyonunun çalışmasını sağlar

class Client

— Metotları

- `Client(const char *serv_host);`
Belirtilen sunucuyla konuşacak yeni bir istemci soket nesnesi yaratır
- `Client(int sockfd);`
Belirtilen soket üzerinden yeni bir istemci soket nesnesi yaratır
- `int connect();`
Sunucu soketine bağlanır, bağlantı durumunu döndürür
- `void send(const char *data,int datasize);`
Sunucu soketine belirtilen veriyi yollar
- `void sendstr(string msg);`
Sunucu soketine belirtilen veriyi yollar

H) Kamera Soket Kütüphanesi

class CameraServer

— Özellikleri

- `Server *server;`
TCP soket sunucusu nesnesi işaretçisi, haberleşme işlemleri için kullanılmaktadır

— Metotları

- `CameraServer();`
Yeni bir kamera soket sunucusu nesnesi oluşturur
- `int isSocketOpened();`
Kamera sunucu soketinin açılıp açılmadığını kontrol eder
- `void listen();`
Kamera sunucu soketini dinlemeye başlar
- `void wait();`
Kamera sunucu soketine yeni bir paket gelmesini bekler. Paket gelince sınıfın `capture` metodunu çalıştırır
- `static void capture(int sockfd);`
Soket üzerinden gelen veriyi dosya adı olarak ayarlar ve kameradan yakalanan görüntü verisinin dosya adıyla sabit diske kaydedilmesini sağlar
- `~CameraServer();`
Kamera soket sunucusunu kapatarak nesneyi bellekten siler

class CameraClient

— Özellikleri

- Client *client;
TCP soket istemcisi nesnesi işaretçisi, haberleşme işlemleri için kullanılmaktadır

— Metotları

- CameraClient(char *camera_servaddr);
camera_servaddr ile belirtilen kamera soket sunucusuna bağlanan yeni bir kamera soket istemci nesnesi oluşturur
- int isSocketOpened();
Kamera istemci soketinin açılıp açılmadığını kontrol eder
- int connect();
Kamera sunucu nesnesine bağlanmaya çalışır, geriye başarımlı durumunu döndürür
- void capture(string filename);
Kamera soket sunucusuna filename verisini yollayarak yeni bir görüntü yakalamasını söyler
- ~CameraClient();
Kamera soket istemcisini kapatarak nesneyi bellekten siler

ÖZGEÇMİŞ

Evrin ÖZÇELİK 15 Kasım 1980 yılında Çanakkale’de dünyaya geldi. 1998 yılında Çanakkale M.P. Anadolu Lisesi’nden mezun olduktan sonra aynı yıl kazandığı İTÜ Bilgisayar Mühendisliği’nde lisans öğrenimine başladı. 4 sene süren lisans eğitiminden 2002 yılında mezun oldu ve İTÜ Yazılım Geliştirme Merkezi’nde Kuzgun adındaki bir web projesine dâhil oldu. Aynı yıl İTÜ Bilgisayar Mühendisliği’nde ağ sistemleri üzerine yüksek lisans eğitime başladı. 1 yıl sonra iş çalışmalarıyla daha paralel olacağını düşündüğü yapay zeka üzerine İTÜ Bilgisayar Mühendisliği’nde yüksek lisans eğitime başladı. Kuzgun projesinin Arı Teknokent tarafından kabul edilmesiyle kurulan bir anonim şirket olan Anka Bilgi Teknolojileri’ne ortak oldu ve 3 sene İnternet Teknolojileri bölüm yöneticiliği yaptı. 2005 yılında Anka Bilgi Teknolojileri’nden ayrılarak uluslar arası bir boyut kazanan Rakun adında bir web projesine başladı. Şu anda bu projeyle IBM Türk ile ortak çalışmalar sürdürmektedir.