

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**TEKERLEKLİ OTONOM ROBOTLAR İÇİN
GÖRÜNTÜ İŞLEME TABANLI BİR HARİTALAMA
ALGORİTMASI GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Müh. Berkan ÇAKIR

Anabilim Dalı : Mekatronik Mühendisliği

Programı : Mekatronik Mühendisliği

HAZİRAN 2008

**TEKERLEKLİ OTONOM ROBOTLAR İÇİN
GÖRÜNTÜ İŞLEME TABANLI BİR HARİTALAMA
ALGORİTMASI GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Müh. Berkan ÇAKIR

(518051005)

Tezin Enstitüye Verildiği Tarih : 30 Nisan 2008

Tezin Savunulduğu Tarih : 11 Haziran 2008

Tez Danışmanı : Yrd. Doç. Dr. Levent OVACIK (İTÜ)

Jüri Üyeleri : Prof. Dr. Ata MUĞAN (İTÜ)

Yrd. Doç. Dr. Erdinç ALTUĞ (İTÜ)

HAZİRAN 2008

ÖNSÖZ

Günümüz dünyasında insanların yerini alabilecek ve verilen görevleri en iyi şekilde yerine getirebilecek makinelere fazlasıyla ihtiyaç duyulmaktadır. Bu makinelerin başında robotlar gelmektedir. Robotlar farklı görevler için tasarlanmış donanımların bir araya getirilmesiyle oluşturulan ve kendi üzerlerinde bulunan yönetim birimleri vasıtasıyla verilen görevleri yerine getiren üst düzey makinelerdir. Endüstriyel üretim, tıp, uzay araştırmaları gibi birçok alanda kullanılmakla beraber kullanılan ortama göre boyut ve kapasite farklılıkları göstermektedirler. Günümüzde sıklıkla kullanılan ve üzerinde birçok çalışma yapılan robot türlerinden biri otonom mobil robotlardır. Otonom mobil robotlar bulunduğu ortam içerisinde kendi başına hareket eden ve ortamdaki nesneleri algılayan küçük boyutlu gezgin robotlardır. Bu tür robotlar içinde bulunduğu ortam ile ilgili bilgileri toplayabilmek ve saklayabilmek, istendiğinde bu bilgileri başka bir birime gönderebilmek üzere tasarlanırlar. Özellikle insanların ulaşamayacağı veya insan sağlığını tehdit edebilecek ortamlarla birlikte gezegen yüzeylerinin araştırılması gibi zorluk derecesi yüksek görevleri yerine getirebilmekte ve bulunduğu ortam hakkında faydalı bilgiler vermektedirler. Bu faydalı bilgilerin başında ortamın haritası gelmektedir. Robotik haritalama üzerinde şimdiye kadar birçok çalışma yapılan ve farklı türden yaklaşımları gerektiren bir alandır. Bu çalışmada, seçilen tekerlekli otonom bir mobil robot aracılığı ile görüntü işleme teknikleri kullanılarak robotun konum bilgisinin doğru bir şekilde alınması ve ortamın haritasının çıkarılması hedef alınmıştır. Bu hedefler doğrultusunda birçok makale ve çalışma incelenerek en güncel yöntemlerle görüntü işleme tabanlı bir haritalama algoritması geliştirilmiştir.

Bu çalışmanın hazırlık aşamalarından başlayarak her konuda desteklerini gördüğüm danışmanım Sayın Yrd. Doç. Dr. Levent Ovacık'a ve geliştirdiği mobil robotu hiçbir karşılık beklemeden kullanmamı sağlayan Yüksek Mühendis Muhittin Atilla'ya teşekkürü bir borç bilirim. Ayrıca her zaman yanımda olan Zübeyde Doğan'a, desteklerini esirgemeyen başta Melih Zenciroğlu olmak üzere arkadaşlarıma ve eğitim hayatım boyunca bana maddi manevi her türlü desteği veren aileme sonsuz teşekkürlerimi sunarım.

Berkan ÇAKIR

HAZİRAN 2008

İÇİNDEKİLER

KISALTMALAR	iv
TABLO LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖZET	viii
SUMMARY	ix
1. GİRİŞ	1
1.1. Robot Navigasyonu	2
1.2. Motivasyon ve Araştırma Konusu	4
1.3. Geçmişte Yapılan Çalışmalar	6
2. GÖRÜNTÜ İŞLEME TABANLI HARİTALAMA	10
2.1. Hareket Modeli	10
2.2. Görsel Odometri	12
2.2.1. Öznitelik Belirleme	14
2.2.2. Öznitelik Eşleştirme	19
2.2.3. Optik Akış	22
2.2.4. Yer Değiştirme ve Dönme Hareketlerinin Hesaplanması	32
2.3. Engel Algılama	34
3. DONANIM ve YAZILIM	38
3.1. Donanım	38
3.2. Yazılım	40
4. DENEYSEL SONUÇLAR ve GENEL DEĞERLENDİRME	43
4.1. Doğrusal Yol Testi	43
4.2. Lokalizasyon Testi	46
4.3. Haritalama Testi	47
4.4. Genel Değerlendirme	50
KAYNAKLAR	52
ÖZGEÇMİŞ	55

KISALTMALAR

MCU	: Mikro kontrolör Ünitesi (İng. Micro Controller Unit)
PIC	: Programlanabilir Ara yüz Kontrolörü (İng. Programmable Interface Controller)
USB	: Genel Seri Hat (İng. Universal Serial Bus)
RF	: Radyo Frekansı (İng. Radio Frequency)
I/O	: Giriş / Çıkış (İng. Input / Output)
SLAM	: Eşzamanlı Lokalizasyon ve Haritalama (İng. Simultaneous Localization and Mapping)
GPS	: Küresel Konum Belirleme Sistemi (İng. Global Positioning System)
INS	: Ataletsel Navigasyon Sistemi (İng. Inertial Navigation System)
NCC	: Normalize Çapraz Korelasyon (İng. Normalized Cross Correlation)
IMU	: Ataletsel Ölçüm Ünitesi (İng. Inertial Measurement Unit)

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1 : 1m uzunluğunda doğrusal bir hat boyunca robotun ölçtüğü mesafe örnekleri.....	44

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 : Lokalizasyon sırasında oluşan hatanın birikmesi sonucu elde edilen haritadaki kayma [3]	3
Şekil 1.2 : Navigasyon üçgeni ve önem dereceleri [2]	3
Şekil 1.3 : Tekerlekli otonom robot [4]	6
Şekil 2.1 : Global hareket eksen ve robotun kendi eksen	10
Şekil 2.2 : Tezde kullanılan görsel odometri algoritmasının akış şeması ..	15
Şekil 2.3 : Görüntü üzerinde köşe ve kenarlar	16
Şekil 2.4 : Özdeğerlerin anlamlandırılması	18
Şekil 2.5 : Prizma, piramit ve küpten oluşan bir görüntüdeki köşe noktalarının bulunması	19
Şekil 2.6 : Bir koridorun çekilmiş görüntüsü üzerinde bulunan köşe noktaları	20
Şekil 2.7 : Benzer iki görüntü üzerindeki özniteliklerin eşleştirilmesi	20
Şekil 2.8 : İki farklı görüntü üzerinde bulunan köşe noktaları	21
Şekil 2.9 : Görüntüler üzerindeki köşelerin eşleştirilmesi	22
Şekil 2.10 : Yüksek eşleşme oranına sahip köşe noktaları	22
Şekil 2.11 : 3-boyutlu uzaydaki bir hareket vektörünün kamera eksenleri üzerinde 2-boyutlu olarak gösterimi	23
Şekil 2.12 : Kameranin hareketi referans alınarak optik akış vektörlerinin hareket yönlerinin anlamlandırılması : a) ileri yönde hareket, b) kendi etrafında dönme hareketi, c) sola doğru hareket	24
Şekil 2.13 : Küçük zaman aralığında alınan iki görüntüden ilki üzerindeki bir noktanın diğer görüntü üzerindeki konumu ve yer değiştirme miktarı	25
Şekil 2.14 : Optik akışı incelenen bir noktanın görüntüler üzerinde arandığı pencerelerin büyüklüğünün piramit şeklinde gösterimleri [28]	28
Şekil 2.15 : Hedef robot platformu	29
Şekil 2.16 : Görüntü koordinatlarından yer düzlemine göre dikey koordinatın hesaplanmasını sağlayan dönüşüm	29
Şekil 2.17 : Robotun x yönünde aldığı mesafeyi hesaplamak için kullanılan yaklaşım: a) yataydaki görme açısı, b) görüntü düzlemi ve gerçek düzleme göre pozisyon	31
Şekil 2.18 : Robotun kendi hareket eksenlerine göre anlık hız ve yer değiştirme vektörleri	33
Şekil 2.19 : Robotun global düzleme ve kendi düzlemine göre yer değiştirmesi	34
Şekil 2.20 : Algılanan engelin robotun kendi hareket eksenine göre konumu	36
Şekil 2.21 : Bir engelin lokal ve global eksen takımında gösterilmesi	36
Şekil 3.1 : Robot gövdesi ve tekerlekler [4]	39

Şekil 3.2	: Robotun Atilla tarafından inşa edilen ve kullanılan son hali [4]	40
Şekil 3.3	: Tezde kullanılan robot ve kamera platformu	41
Şekil 4.1	: x- eksenı üzerinde L mesafesi kadar gitmesi istenen robotun algıladıđı mesafeler ve ölçüm hataları	43
Şekil 4.2	: Robotun el ile 50cm uzunluğunda doğrusal bir hat üzerinde ilerletilmesiyle ölçülen doğrusal yoldan sapma miktarları	45
Şekil 4.3	: Robotun el ile 50cm uzunluğunda doğrusal bir hat üzerinde ilerletilmesiyle ölçülen doğrusal mesafeler	46
Şekil 4.4	: 1m x 1m ölçülerinde kare şeklinde bir yol üzerinde robotun oluşturduđu harita	47
Şekil 4.5	: Robotun hareketini ve ortaya koyduđu haritayı test etmek için oluşturulan koridor	47
Şekil 4.6	: Robotun 1 m uzunluğunda ve L şeklinde oluşturulan koridor boyunca yaptıđı harekete ilişkin: a) ortam haritası, b) ideal şekil	49
Şekil 4.7	: a) Ultrasonik sensordan çıkan dalgaların engellerin kesişim bölgelerinde yansıması ve b) koridor boyunca ilerleyen robotun köşe noktalarındaki yansılardan dolayı algıladıđı hatalı noktalar	49
Şekil 4.8	: 1m x 1m alan içerisinde konumlandırılmış cisimler arasında: a) robotun izlediđi yol ve oluşturduđu harita, b) idealde robotun izlemesi gereken yol ve ortamın haritası	50

TEKERLEKLİ OTONOM ROBOTLAR İÇİN GÖRÜNTÜ İŞLEME TABANLI BİR HARİTALAMA ALGORİTMASI GELİŞTİRİLMESİ

ÖZET

Bu tez çalışmasında, üzerinde bulunan eyleyiciler ve algılayıcılar sayesinde içerisinde bulunduğu ortamda yarı otonom olarak hareket edebilen tekerlekli mobil bir robot ve bu robot üzerinde bulunan algılayıcılardan ve kameradan gelen verileri işleyen bir masaüstü bilgisayar ile görüntü işleme tabanlı bir haritalama algoritması geliştirilmiştir. Mobil robot, masaüstü bilgisayara kablosuz radyo frekans hattı ile bağlanmakta ve üzerinde bulunan algılayıcılardan gelen verileri masaüstü bilgisayara eş zamanlı olarak gönderebilmektedir. Masaüstü bilgisayar ise robottan gelen verileri değerlendirerek gerektiğinde robota komut gönderebilmektedir. Bu şekilde yüksek hacimli işlemler için robot yerine masaüstü bilgisayar kullanılabilir. Robot üzerinde bulunan kamera ile robotun hareket ettiği yüzeyden alınan görüntüler masaüstü bilgisayara iletilerek robotun konum bilgisini hesaplamak için gerekli olan veriler sağlanmaktadır. Masaüstü bilgisayar, kameradan aldığı görüntüler ile optik akış vektörlerini kullanarak robotun anlık konumunu hesaplamakta ve robotun içerisinde bulunduğu ortamdaki engellere ilişkin mesafe bilgileriyle ortamın haritasını çıkarmaktadır.

Tez temel olarak dört bölümden oluşmaktadır. İlk bölümde robot navigasyonu, robot navigasyonunun temel bileşenleri ile karşılaşılabilen problemlere değinilmiştir. Bu bölümde ayrıca tezin motivasyonu ve bu alanda yapılan benzer çalışmalardan bahsedilerek okuyucuya genel bir bakış açısı kazandırılmıştır. İkinci bölümde bu çalışma için geliştirilen görsel odometri sisteminden detaylı olarak bahsedilmiştir. İlk olarak robotun hareket modeli ele alınmış daha sonra görüntü işleme tabanlı lokalizasyon ve haritalama için kullanılan yöntemlerden detaylıca açıklanmıştır. Bölüm sonunda kameradan alınan görüntülerden elde edilen optik akış vektörleriyle robotun gerçek hareketini hesaplamak ve ultrasonik sensordan alınan engellere ilişkin mesafe bilgileriyle robotun içinde dolaştığı ortamın haritasını çıkarmak için oluşturulan matematiksel model açıklanmıştır. Üçüncü bölümde tezde kullanılan robot platformun donanımı ve kullanılan yazılım araçları tanıtılmıştır. Tezin son bölümünde ise haritalama algoritmasının doğruluğunu sınamak amacıyla yapılan testler ve bu testlerden elde edilen sonuçlar yorumlanmıştır.

DEVELOPING A VISION BASED MAPPING ALGORITHM FOR AUTONOMOUS WHEELED ROBOTS

SUMMARY

In this present thesis, a mobile wheeled robot and a PC are used for developing a vision based mapping algorithm. The mobile robot is able to move semi-autonomously in its environment with the help of actuators and sensors on it and the PC processes the data acquired both from the sensors and the camera which are located on the robot. The mobile robot has been connected to the PC with a wireless radio frequency link and it can also send the data acquired from its sensors to the PC simultaneously. On the other hand, the PC assesses the data received from the robot. Also it is able to send commands to the robot whenever it is required. In this way, it is possible to use the PC instead of the robot for intensive computations. The camera, which is located on the robot, takes the images of the surface on which the robot moves. These images are transmitted to the PC in order to obtain the data required for estimating the location of the robot. The PC uses optical flow vectors on the images taken from camera for calculating the immediate position of the robot. It also processes the map of the environment in which robot moves by using the distances of the obstacles located in that environment.

The thesis consists of four chapters mainly. In the first chapter, the navigation of the robot, the basic components of the navigation process, and the problems concerning the navigation has been stated. Furthermore a general point of view is brought into the reader explaining the basic motive of the thesis, and the similar works on this present subject. In the second chapter, the visual odometry system developed within this system is presented in detail. First, the kinematic model of the robot is explained, then the methods of both the vision based localization, and mapping is explained in detail. At the end of this chapter, the mathematical model which is developed in order to not only calculate the real movement of the robot by the optical flow vectors acquired from the images taken by the camera of the robot, but also draw a map of the environment in which the robot is moving using the ultrasonic sensor data concerning the distance of the obstacles is explained. In the third chapter, the hardware of the robot platform used during this thesis is mentioned, and the software instruments are introduced. In the last chapter of the thesis, some tests that are applied to examine the validity of the mapping algorithm and the results of these tests are interpreted.

1. GİRİŞ

Robotik, fiziksel aktivite ve karar verme gibi uygulamalarla bir görevi yürüterek insanların yerini alabilecek makinelerle ilgili çalışmaları içeren bilim dalıdır [1]. Günümüzde uzay çalışmaları, endüstriyel üretim ve kontrol sistemleri, tıbbi (medikal) uygulamalar, bilgisayar ve insan arasındaki etkileşim sistemleri, insan sağlığını ve güvenliğini tehdit edici ortamlar gibi daha birçok alanda robotlardan yaygın bir şekilde faydalanılmaktadır. Uygulamada kullanılan robotik platform ne türden olursa olsun, elektronik, bilgisayar, makine ve kontrol mühendisliği alanlarında çalışma yapan araştırmacıları bir araya getiren disiplinler arası bir çalışmayı gerektirmektedir.

Birçok robotik uygulamada, robottan kendisine verilen bir görevi yerine getirirken, bir noktadan başka bir nokta gitmesi, diğer bir deyişle ortam içinde hareket etmesi beklenebilir. Böyle uygulamalarda kullanılan robotlar mobil robot olarak adlandırılırlar. Çoğu mobil robot uygulamalarında robotlar içerisinde çalıştığı ve görev bölgesi olarak tanımlanmış ortamlarda hareketini engelleyebilecek nesneler ile ya da ne yönde gideceği konusunda karar vermesi gereken alternatif yol ayrımları ile karşılaşabilirler. Mobil robotların görev bölgesi içerisindeki engellere çarpmaksızın, güvenli bir şekilde ve en uygun yolları takip ederek görevini yerine getirebilmesi için içinde bulunduğu ortamı tanıyabilmesi gerekmektedir. Görev bölgesi içerisinde ortamı verilen belirli bir süre içinde tanıyan ya da öğrenen, daha sonra uygun hareket yollarını tayin edebilen ve bu yollarda kendi kendine ilerleyebilen robotlar otonom mobil robotlar olarak isimlendirilirler.

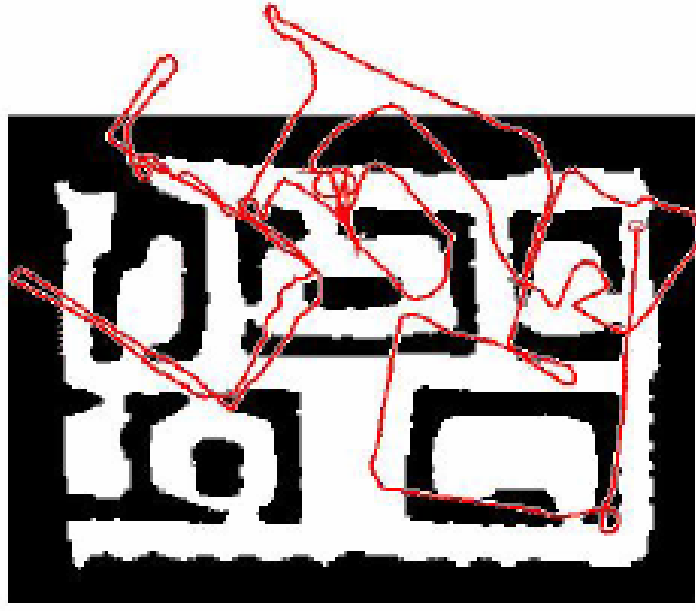
Otonom mobil robot uygulamalarında, kullanılan robotlar ne amaçla kullanılırsa kullanılsın, etrafındaki fiziksel çevreyi algılayabilmesi ve hareketlerini etrafındaki bu fiziksel çevreye uygun olarak yapması gerekir. Bundan dolayı, mobil bir robotun herhangi bir anda bulunduğu konumu bilmesi (lokalizasyon) ve yakın civarındaki nesneleri ve bu nesnelerin kendi konumuna göre olan konumlarını algılayarak görev yaptığı ortamın haritasını çıkarması (haritalama) problemi önem kazanmaktadır. Böylece, gidebileceği konumların ve ortamın haritasının tanıtıldığı bir robot, ortam

içerisinde karşısına çıkabilecek engellere çarpmaksızın planlı bir şekilde hareket edebilme yeteneğine sahip olabilecektir.

Lokalizasyon ve Haritalama, bir robotun bulunduğu fiziksel ortamı tanıyabilmesi için gerekli olan önemli kabiliyetlerdir. Lokalizasyon robotun her hangi bir anda konumunu belirleyebilme yeteneğidir. Haritalama ise robotun konum bilgisini de kullanarak etrafında algılayabildiği engellerin veya nesnelerin konumlarını ve şekillerini belirleyerek bir harita oluşturulmasıdır. Robotun bulunduğu ortam içerisindeki hareketi sırasında algıladığı her yeni cisim, robotun konum değiştirdiği anlamına gelir. Ancak, mükemmel bir lokalizasyon için robotun daha önceden algıladığı cisimleri, o ana kadar hiç algılamadığı cisimlerden ayırt edebilmesi gerekir. Bu ayır edebilme yeteneği robotun bulunduğu konum bilgisinin doğruluğuna bağlıdır. Konum bilgisinin doğruluğu ise, araç üzerine yerleştirilen sensorların hassasiyetine ve hata oranına bağlıdır. Lokalizasyon sırasında oluşacak hatalar, oluşturulan haritanın da doğruluğunu etkileyecektir. Robotun konum değiştirmesiyle, yeni konumunun belirlenmesinde daha önceki harita bilgisinden faydalanılacağından, konum belirlemedeki hata birikerek artacaktır. Bunun sonucunda, zaman içerisinde robotun belirlediği harita gerçek haritadan çok farklı olacaktır. Şekil 1.1 araç tekerlerinin hareketini algılayan kodlayıcıdaki (İng. encoder) kaçıklıktan kaynaklanan hataların gerçek bir uygulamada elde edilen haritalama sonuçlarını ne derecede etkileyebileceğini göstermektedir.

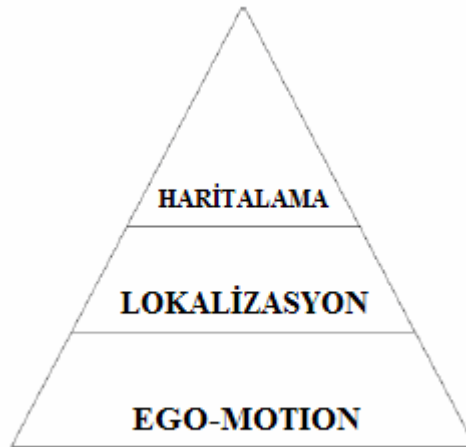
1.1. Robot Navigasyonu

Navigasyon kelime anlamı olarak yönlendirme anlamına gelir. Birçok mobil robot uygulamasında, robotun kendi kendini yönlendirmesi yani otonom hareket edebilmesi gerekir. Navigasyon yapamayan bir robot fiziksel dünyaya uygun hareket işlemini gerçekleştiremez. Nasıl ki insanlar görsel ve işitsel olarak algılarıyla otomobil içerisinde araca uygun hareketi veriyorlar ise, robotlar da sahip oldukları yardımcı donanımlar yardımıyla kendi hareketlerini yapmak durumundadırlar. Bir robot için navigasyon işlemleri üç parçadan oluşur: haritalama, lokalizasyon ve kendi hareketini algılama (İng. egomotion). Şekil 1.2, bir robot için bu işlemleri ve kendi aralarındaki hiyerarşiyi göstermektedir. Bir robot kendi hareketini anlamlandıramadan lokalizasyon yapamaz, lokalizasyon yapmadan da ortamın haritasını çıkaramaz [2].



Şekil 1.1 : Lokalizasyon sırasında oluşan hatanın birikmesi sonucu elde edilen haritadaki kayma [3]

Egomotion bir ortam içerisinde hareket eden robotun üzerinde bulunan sensorlardan gelen verilere göre kendi hareketini anlamlandırabilmesidir. Örneğin üzerinde kamera bulunan bir robot hareketsiz dururken ortam içerisinde bulunan diğer nesneler yer değiştiriyor ise robot bu hareketleri kendi hareketi gibi algılayabilir. Bu da robotun yanlış bir lokalizasyon ve dolayısıyla haritalama yapmasına sebep olabilir. Bundan dolayı robotun içinde bulunduğu çevrede kendi hareketini veya hareket miktarını doğru bir şekilde bulması önemlidir. Bu çalışmada robotun içinde



Şekil 1.2 : Navigasyon üçgeni ve önem dereceleri [2]

bulunduğu ortamdaki nesnelerin hareketsiz olduğu varsayılarak lokalizasyon ve haritalama uygulaması yapılmıştır. Robotun içinde bulunduğu ortamdaki hareketli nesneleri algılayabilmesi ve kendi hareketini diğer nesnelerin hareketlerinden ayırt edebilmesi ayrı bir çalışma konusu olarak ele alınmaktadır.

1.2. Motivasyon ve Araştırma Konusu

Mobil bir robotun otonom olarak navigasyonu için gerekli lokalizasyon işlemleri sırasında robot üzerine yerleştirilmiş algılayıcılardan elde edilen mesafe ölçen ve engelleri algılayan ultrasonik algılayıcılar, lazer algılayıcılar ve diğer optik algılayıcılardan veya kameralardan yararlanır. Robotun hareketi ve yönü ise yine robot üzerine yerleştirilmiş olan eyleyiciler sayesinde kontrollü bir şekilde yapılır. Henüz içerisinde çalışacağı ortam hakkında bir bilgiye sahip olmayan bir robot için, ortam içerisinde rastgele dolaşırken önüne çıkacak engellere çarpmaması da gerekir. Bunun için algılayıcılardan elde edilen ölçüm sonuçlarının hızlı bir şekilde işlenerek sayısal verilere dönüştürülmesi ve bu verilerin değerlendirmesi gerekir. Bu anlık ölçümlerin değerlendirilmesi bir dizi basit aritmetik hesaplamalar ve karşılaştırmalar yardımı ile gerçekleşir. Bu işlemlerin sonucunda elde edilen konum değiştirme işlemleri sonucunda robot yeni bir konuma yerleşerek bir dizi ölçümler yapar. Bu yeni ölçümlerle haritalama için gerekli değerlendirmelerin yapılması ve elde edilen sonuçların bir bilgi depolama (bellek) ortamında saklanması gerekir. Bu işlemlerin hızlı bir şekilde yapılabilmesi için robot üzerinde programlanmış bir mikroişlemci ile hesaplanan verilerin saklanacağı bir veri depolama biriminin olması gerekir. Robot üzerine yerleştirilen bu donanımın pratik olarak uygulanabilirliği, kullanılacak robot platformunun boyutları ile sınırlanır.

Boyut sınırlaması olmayan mobil robot uygulamalarında robot üzerine yerleştirilen bir merkezi işlem birimi sayesinde ölçüm sonuçlarını değerlendirilebilir. Robot üzerine yerleştirilecek bir bilgisayar, algılayıcılardan gelecek işaretleri ve kamera görüntüleri gibi yoğun bilgiyi işleyerek, lokalizasyon ve haritalama yapabilir ve bu bilgileri robot üzerinde saklayabilir. Böyle bir robot, dış ortamdan tamamen bağımsız bir şekilde otonom olarak çalışabilir.

Robot boyutları küçülmeye başladıkça, robot üzerine yerleştirilecek işlemci ve diğer bilgisayar donanımı, robot ağırlığını arttıracak, robotun hareket kabiliyetini engelleyecektir. Çoğu zaman bu donanımı robotun üzerine yerleştirmek bile

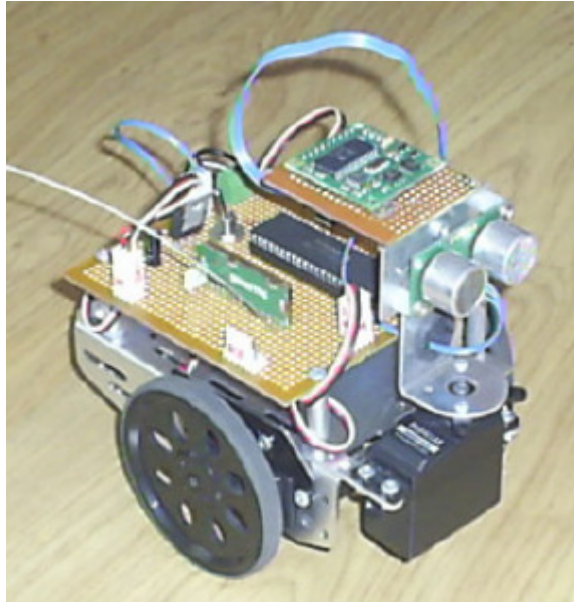
mümkün olamayacaktır. Bu tür sınırlamaların olduğu mini robot uygulamalarında robot üzerine mümkün olduğunca basit bir işlemci kullanılarak robotun hareket kontrolü için gerekli basit komutları icra edilebilir. İşlemci gücü düşük olduğundan algılayıcılardan elde edilen ölçümler robot üzerine yerleştirilmiş bir Radyo Frekansı (RF) Alıcı/Verici yardımıyla robot dışında bulunan ve güçlü bir bilgisayar ünitesinden oluşan merkezi işlem birimine sayısal olarak iletilir. Böylece lokalizasyon ve haritalama için gerekli sayısal bilgi robot dışında işlenerek değerlendirilir, depolanır ve robotun güvenli bir biçimde çalışma bölgesinde hareket edebilmesi için gerekli komutlar robota iletilir. Fiziksel olarak erişimi zor bölgelerde kullanılacak küçük boyutlu robotlar sayesinde arama ve görüntüleme faaliyetlerinin yürütülmesi bu tür teknolojiler sayesinde mümkün olabilecektir.

Bu tez çalışmasında, daha önceden geliştirilen küçük boyutlu bir robot platformu için lokalizasyon ve haritalama algoritmalarının geliştirilmesi ve bu algoritmaların performansının belirlenmesi amaçlanmıştır. Bu tezin kapsamı, hedef olarak ortaya konan bir ortamın haritasını çıkarma işlemi için yukarıda sözü edilen problemlerin (lokalizasyon, haritalama) görüntü işleme teknikleri kullanılarak piyasada rahatlıkla bulunabilecek düşük maliyetli bir kamerayla doğru bir şekilde tespit edilmesi ve parçaların en uygun şekilde birleştirilmesi bu tezin temel problemini oluşturmaktadır.

Aşağıda bahsedilen çalışmalar birçok konudan etkilenecek şekilde ortaya konulmuştur. Ancak temel çıkış noktası Haziran 2007’de İ.T.Ü Fen Bilimleri Enstitüsü Mekatronik Mühendisliği öğrencilerinden Atilla [4] tarafından inşa edilen tekerlekli otonom bir robot ve bu robotla yapılmaya çalışılan haritalama işlemlerini kapsayan tez çalışmasıdır. İlgili çalışmada Şekil 1.3’te üzere bulunduğu ortamdaki engelleri algılayabilen ve üzerinde bulunan RF modem ile kullanıcıya kodlayıcı bilgisini ve açısal konumunu bildiren bir robotun tasarım aşamaları verilmiştir. Çalışmanın sonraki bölümlerinde günümüzde kullanılan haritalama tekniklerinden bahsedilmiş ve M. Atilla tarafından yapılan haritalama işlemleri anlatılmıştır.

Robot üzerinde, tekerlerin hareketini algılaması için optik kodlayıcı (İng. optical encoder) ve açısal konumunu vermesi için elektronik pusula bulunmaktadır. Optik kodlayıcı ve elektronik pusula bilgisiyle aracın kinematik denklemleri çıkarılmış ve kullanıcı tarafındaki bilgisayar üzerinde çalışan yazılım arayüzünde konum bilgisi verilmiştir. Kodlayıcı bilgisi ile hesaplanan hareket modeline “encoder odometry” denilmektedir. Eğer sensorlar hatasız bilgi verebilirse mükemmel bir lokalizasyon

bilgisi alınabilir. Ancak ilgili platformda kullanılan sensorlar özellikle elektronik pusula hassas olmayan bilgi vermekte veya çevresinde bulunan elektronik donanımlardan çabuk etkilenmektedir. Bu tür problemlerden kaynaklanan hatalardan dolayı yanlış lokalizasyon bilgisi alınmaktadır. Bu nedenlerden dolayı mevcut platform yapısını bozmadan minimum sayıda sensor kullanarak lokalizasyon yapan ve ortamın haritasını çıkaran bir robot platformu oluşturmak ve buna uygun algoritma geliştirmek bu tezin temel motivasyonu olmuştur. Bu amaçla ileriki bölümlerde anlatıldığı üzere araç üzerine bir kamera yerleştirilmiş, görüntü işleme teknikleriyle bir görsel odometri sistemi oluşturularak aracın yer değiştirme miktarının ve açısal dönme miktarlarının hesaplanması amaç edinilmiştir.



Şekil 1.3 : Tekerlekli otonom robot [4]

1.3. Geçmişte Yapılan Çalışmalar

Mobil robotlar ile görüntü işleme tabanlı haritalama ve lokalizasyon çalışmalarının başlama noktası, robot üzerine monte edilmiş kamera ile çekilmiş görüntülerin karakteristiklerinin bulunmasıdır. Kamera ile alınan görüntülerden elde edilen temel karakteristik bilgi görüntü üzerindeki özniteliklerin (İng. feature) belirlenmesidir. Küçük zaman dilimlerinde alınan her bir görüntü üzerindeki özniteliklerin belirlenmesi bu tür çalışmalarda kullanılan genel bir yaklaşımdır. Görüntüler üzerindeki özniteliklerin eşleştirilmesi ve bu özniteliklerin görüntüler üzerindeki konum değişikliklerinden robotun hareket miktarının bulunması çalışmaların bir diğer ayağını oluşturmaktadır.

Özniteliklerin belirlenmesi konusu üzerinde birçok çalışma yapılmış olmakla beraber hala çalışılan bir konu olması bakımından önemlidir. Günümüzde köşe belirleme (İng. corner detection) olarak da anılmaktadır. Köşe iki sınırın kesişimi olarak tanımlanmaktadır [5]. Köşe belirleme ile ilgili ilk çalışmaları 1977 yılında Moravec [6] yapmıştır. Bu çalışmasında Moravec görüntü üzerindeki her noktanın (İng. pixel) komşuları ile aralarındaki yoğunluk değişimine bakarak ilgili noktanın köşe olup olmadığı sorusuna cevap aramıştır. Günümüzde en yoğun olarak kullanılan köşe belirleme yöntemini 1988 yılında Harris ve Stephens [7] yaptıkları çalışma ile ortaya koymuşlardır. Literatürde Harris Corner Detector olarak geçen bu yöntem ileriki bölümlerde detaylı bir şekilde incelenmiştir. Öznitelik belirleme konusu ile ilgili olarak yukarıda bahsedilenler dışında birçok çalışma yapılmış ve yeni yöntemler ortaya konulmuştur [8, 9-11].

Belirlenen özniteliklerin her bir görüntü için belirlenmesi ve eşleştirilmesi problemi korelasyon (İng. correlation) olarak adlandırılır. Özellikle hareket takibi gibi konularda ve bir çok robotik uygulamada yaygın bir şekilde kullanılır. Ancak temeli işaret işleme yöntemlerine dayanır. Görüntü işlemede ise bir görüntü üzerindeki her bir özneliğin diğer görüntü üzerindeki özniteliklerle istatistiksel olarak karşılaştırılması ilkesine dayanır.

Görüntüler üzerinde özniteliklerin belirlenmesi ve eşleştirilmesinden sonra eşleştirilen özniteliklerin her bir görüntü çiftindeki yer değiştirme miktarı fiziksel ortamda belli bir yer değiştirmeye denk gelmektedir. Ne kadarlık bir yer değiştirme olduğu sorusundan yola çıkarak görsel odometri algoritması oluşturulur. Görüntü üzerinde bulunan öznitelikler ile nesnelerin tanınması konusu üzerine çalışmalar yaptıktan sonra Lowe [12], Se ve Little [13-14] ile birlikte görüntüler üzerinde bulunan özniteliklerin eşleştirme işleminden sonra ortaya çıkan yer değiştirmelerinden yola çıkarak mobil robotlar için lokalizasyon ve haritalama algoritması geliştirmişler ve robotun pozisyonunu belirlemede başarılı olmuşlardır.

Görsel odometri, otonom olarak hareket eden robot üzerinde bulunan kameradan alınan görüntüler üzerinde optik akış vektörlerinin bulunması yöntemiyle de oluşturulabilir. Bu tarz uygulamalardan biri Dornhege ve Kleiner [15] tarafından yapılan haritalama çalışmasıdır. Bu çalışmada kullanılan robot üzerinde bulunan dört adet paletle ortam içerisinde hareket etmekte ve basamak gibi engelleri aşabilmektedir. Dornhege ve Kleiner çalışmalarında düşük maliyetli bir kamera ve

IMU (İng. Inertial Navigation Unit) ile robotun pozisyonunu tahmini için çözüm yolları aramışlardır. IMU robotun yön bilgisini vermekte ve kamera ile alınan görüntüler üzerindeki öznitelikler “KLT feature tracker” [16] yöntemiyle robotun ileri ve geri yöndeki hareket miktarını vermektedir. Çalışmanın sonuçlar bölümünde ortaya konan değerler ve harita kıyaslamaları göz önüne alındığında ucuz bir kamera sistemiyle mobil bir robot üzerinden ortamın gerçeğe yakın bir haritasının elde edilebilmesinin mümkün olduğu gözükmemektedir.

Optik akış vektörlerinden faydalanarak bir görsel odometri sisteminin oluşturulması problemi Campbell, Sukthankar ve Nourbakhsh [17] tarafından incelenmiştir. Bu çalışmada amaç üzerinde ucuz bir kamera olan ve piyasada kolayca bulunan donanımlarla görsel bir odometri sistemi oluşturmaktır. Robotun üzerinde bulunan kameranın robotun içinde bulunduğu ortama doğrudan baktığı durumlarda robotun açısal ve doğrusal hareketini algılamak için 3-boyutlu ortamın 2-boyutlu izdüşümlerinden hareketi algılamak gerekmektedir. Bu tür uygulamalarda alınan sonuçlar tatmin edici olmadığı ve uygulaması zor olduğu için kameranın robotun hareket ettiği yüzey üzerinden elde edilen optik akış vektörlerinin analizinden daha verimli sonuçlar alınabileceği vurgulanmıştır. Campbell, Sukthankar ve Nourbakhsh [18] bir sonraki adım olarak optik akış vektörleriyle elde edilen lokalizasyon bilgisinin doğruluğunu mobil bir robot platformuyla birlikte ofis halısı, çim, asfalt ve buz gibi zorlu yüzeyler üzerinde test etmişlerdir. Aynı araştırmacılar Pahwa [19] ile birlikte daha kararlı sonuçlar veren ve robotun hareket ettiği yüzey üzerinde bulunan nesnelerin algılanması için bir mobil platform oluşturmuşlardır. Bu çalışmada kameranın yüzeye bakış açısı, yerden yüksekliği gibi bilgiler ışığında robotun gerçekte ne kadar yer değiştirdiğini bulmuşlardır. Bu çalışmalar sadece ucuz bir kamera sistemiyle ve bazı temel dönüşümler ile robotun lokalizasyon bilgisinin alınmasında bu teze ilham kaynağı olmuştur.

Yukarıda anlatılan çalışmalara benzer olarak diğer bir çalışma DiVerdi ve Hollerer [20] tarafından yapılmıştır. Bu çalışmada kullanılan hareketli platform mobil bir robot yerine insan olarak seçilmiştir. Bir sırt çantasına bağlanan kamera ve sırt çantasını taşıyan kişinin omuzlarına sabitlenen IMU ile hareket eden kişiye ait lokalizasyon bilgisi alınmaktadır. Sırt çantasına bağlı olan kamera yere bakmakta ve yer yüzeyinden alınan görüntüler üzerindeki öznitelikler takip edilmektedir. Özniteliklerin yer değiştirme miktarı ve IMU’dan alınan açısal bilgiler ışığında

gerçeğe yakın bir lokalizasyon bilgisi alınmıştır. Ayrıca Cheng [21, 22] önderliğinde Mars yüzeyine gönderilen mobil robotlar için oluşturulan görsel odometri sistemlerinde robotun hareket ettiği yüzeyden alınan optik akış vektörleri kullanılmıştır. Bu vektörler daha karışık filtreleme işlemleri ile gezegen yüzeyleri hakkında bilgiler vermektedir.

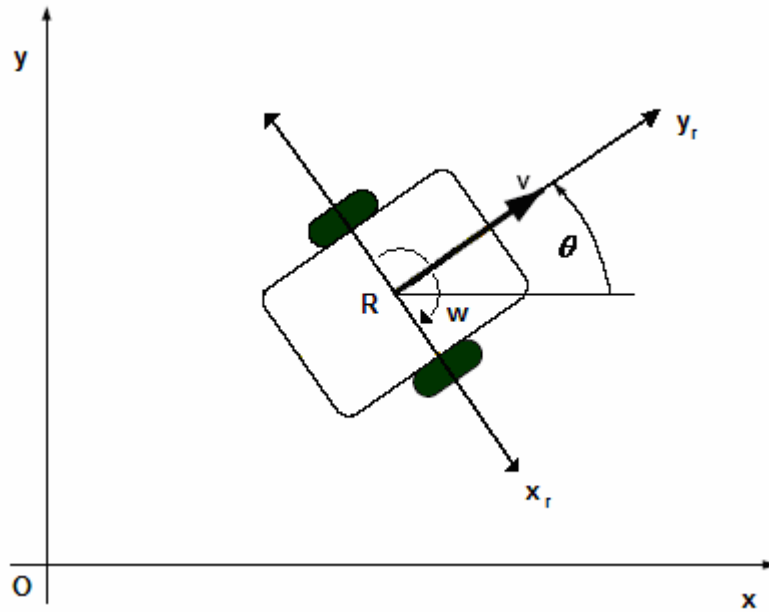
2. GÖRÜNTÜ İŞLEME TABANLI HARİTALAMA

Bu bölümde tez çalışmasında kullanılan robotun hareket modeli, robotun konum bilgisi için oluşturulan görüntü işleme tabanlı lokalizasyon algoritması ve bulunduğu ortamın haritasını çıkarmak için oluşturulan haritalama yapısı anlatılacaktır.

2.1. Hareket Modeli

Tekerlekli mobil robot, tekerleri olan ve otonom hareket eden bir araçtır [23]. Her araçta olduğu gibi mobil robotlar da belli bir kinematik modele göre hareket eder. Kinematik hareketi belirleyen ise araç üzerinde bulunan donanımlardır. Bu bölümde tez çalışmasında kullanılan tekerlekli mobil robotun hareket modeli açıklanacaktır.

Tekerlekli otonom robotun hareket denklemleri Şekil 2.1’de görülen koordinat eksenlerine göre düzenlenmiştir. Şekilde görüldüğü üzere O , robotunda içinde bulunduğu global koordinat düzlemini, R ise robotun gövde eksenine göre oluşturulmuş koordinat düzlemini göstermektedir.



Şekil 2.1 : Global hareket ekseni ve robotun kendi ekseni

$O(x,y)$ hareketin sabit eksenini ve $R(x,y)$ robot üzerindeki hareketli eksenini göstermektedir. Bu çalışmada oluşturulan kinematik denklemler tekerleklerin hareketinde kaçıklık veya kayıp olmadığı varsayılarak oluşturulmuştur. Ayrıca tekerlerin yüzeye noktasal temas ettiği düşünülmüştür. Robotun ön ve arkasından geçen eksen, R 'nin y eksenini, yanlarından geçen eksen ise R 'nin x eksenini oluşturmaktadır. Robotun her konum değişikliği aynı zamanda $R(x,y)$ koordinat düzleminin konum ve açısal olarak yer değiştirdiği anlamına gelmektedir. $O(x,y)$ koordinat düzlemi ise sabittir ve robotun asıl konumu bu düzlem üzerinde bulunduğu noktalardır. Robotun anlık konum değişikliği (2.1)'de gösterildiği şekildedir.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} \quad (2.1)$$

Burada :

$\dot{x}_r$ robotun yataydaki anlık konum değişimini,

$\dot{y}_r$ robotun dikeydeki anlık konum değişimini,

$\dot{\theta}$ robotun yönelim açısındaki anlık değişimi,

v_r robotun ileri yöndeki hızını,

w_r ise robotun yöneliminin açısal hızını göstermektedir. Robotun global eksene göre anlık konumu, bir önceki konumu ile anlık konum değişikliğinin toplamı şeklinde ifade edilir. Sürekli zamanda global eksene göre hareket:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \int_{t-1}^t \begin{bmatrix} v_r \cdot \cos \theta_{t-1} \\ v_r \cdot \sin \theta_{t-1} \\ w_r \end{bmatrix} dt \quad (2.2)$$

şeklinde ifade edilir. Aynı model ayırık zamanda ise şu şekilde ifade edilmektedir:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \Delta T \cdot \begin{bmatrix} v_r \cdot \cos \theta_{k-1} \\ v_r \cdot \sin \theta_{k-1} \\ w_r \end{bmatrix} \quad (2.3)$$

ΔT , k . konum ve $(k-1)$. konum arasındaki zaman dilimini gösterir.

Bu denklemden anlaşılacağı üzere bir robotun anlık konum değişimini, dolayısıyla global eksene göre konumunu belirlemek için ileri yöndeki hızı ve eksenel dönüş hızını ölçmek yeterli olacaktır. Ancak bu çalışmada kullanılan tekerlekli mobil robot üzerinde ileri yöndeki hızını veya eksenel dönüş hızını doğrudan ölçecek bir donanım bulunmamaktadır. Bu durumda robotun belli zaman dilimlerindeki ileri yöndeki ve eksenel dönme yönündeki yer değiştirmelerini bulmak ve (2.4)'teki denklem uyarınca konumlarını farklı yöntemlerle hesaplamak gerekmektedir.

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} \quad (2.4)$$

Yukarıda anlatıldığı üzere robotun konumunu hesaplamak için gerekli kinematik model çıkarılmıştır. Bundan sonraki çalışmalar görüntü işleme yöntemiyle Δx , Δy ve $\Delta \theta$ değişim miktarlarını algılayabilme hedefi doğrultusunda ilerletilmiştir.

2.2. Görsel Odometri

Odometri, tekerlekli araçların seyri sırasında pozisyonlarının belirlenmesi ya da yer değiştirmelerinin hesaplanması anlamına gelmektedir. Robotik uygulamalarda ise tekerlekli robotun başlangıç pozisyonuna göre konumunun tahmin edilmesi şeklinde tanımlanmaktadır. Odometri, klasik yaklaşımla tekerleklerin dönüş hareketinden gelen verilerin kullanılmasıdır. Bu veriler tekerleklerle entegre edilmiş kodlayıcı gibi donanımlardan alınır. Ancak odometri bilgisinin kesinliği için tekerleklerin dönüş hareketini algılayan donanımların kalibre edilmiş olması, hassas olması ve hata seviyelerinin düşük olması gerekir [24]. Günümüzde ise robotların odometri bilgileri klasik yöntemler dışında farklı donanımlar veya hesaplama teknikleri ile belirlenir. Görsel odometri bunlardan biridir.

Görsel odometri basit bir tanımla robotun pozisyonunun veya konum değişikliğinin görüntüler üzerinden belirlenmesidir. Görüntüler üzerinden robotun hareketinin incelenmesi için görüntülerdeki bilginin işlenmesi gerekmektedir. Bunun için farklı teknikler kullanılabilir. Ancak ortak hedef, robotun yer değiştirme miktarının mümkün olan en üst seviyede hesaplanmasıdır. Robotun anlık yer değiştirme miktarları toplamı da belli bir referansa göre konumunu verecektir. Görsel odometri yönteminin avantajları ise şunlardır:

- Görsel Odometri ile bir görüntü kaynağından hareket algılanabilir.
- Hareketi algılayabilmek için ortamı önceden bilmeye gerek yoktur.
- Haritalama işlemi çok hızlı bir şekilde uygulanabilir.
- Elde edilen veriler GPS, INS, kodlayıcı veya diğer sensorlardan gelen bilgilerle kolayca birleştirilebilir.
- Bu algoritma otonom hava araçları, kara araçları ve elde tutulan video kaynaklarından gelen görüntülere de uygulanabilir.

Görsel odometri bir görüntü kaynağından alınan görüntüler üzerinde seçilen nokta veya bölgelerin diğer görüntüler üzerindeki yer değiştirmelerini ve görüntü üzerindeki bu yer değiştirmelerin robotun fiziksel ortamda ne kadar bir yer değiştirmesine karşılık geldiğini farklı yöntemlerle bulma esasına dayanır. Görüntüler üzerinde seçilen noktaların veya bölgelerin belirlenmesi öznitelik belirleme olarak tanımlanmaktadır. Özniteliklerin hangi kriterlere göre belirleneceği birçok çalışmanın temelini oluşturmuştur. Çünkü görüntü üzerindeki özniteliklerin belli bir temele dayandırılmış ve takip edilebilir olması gerekmektedir.

Robotun hareket miktarını ve hareket yönünü belirlemek için her bir görüntü üzerinde özniteliklerin belirlenmesi ve belirlenen bu özniteliklerin görüntüler arasında eşleştirilmesi gerekmektedir. Eşleştirme işleminden sonra özniteliklerin görüntü üzerindeki yer değiştirmeleri optik akış yöntemiyle bulunan yer değiştirmeler ve bazı hesaplamalarla karşılaştırılır. Özniteliklerin yer değiştirme miktarı, eşleştirme ve optik akış işlemlerinde aynı sonucu veriyorsa bu noktalar robotun hareket miktarını belirlemek için kullanılır. Bu noktadan sonra robotun gerçekte ne kadar yer değiştirdiği kamera açısı, yüksekliği gibi bazı bilgiler ışığında hesaplanır. Son olarak robot üzerinde bulunan sensor ile algılanan engellerin robota göre konumları ve robotun hareketi kullanıcı tarafındaki bilgisayar üzerinde gerçek zamanlı olarak haritaya aktarılır. Görüntü işleme tabanlı haritalama sisteminde uygulanan adımlar aşağıda sıralandığı ve Şekil 2.2’de gösterildiği şekilde sıralanabilir:

Her bir görüntü çifti için;

- 1. görüntüyü al.
- Öznitelikleri belirle ve sakla.

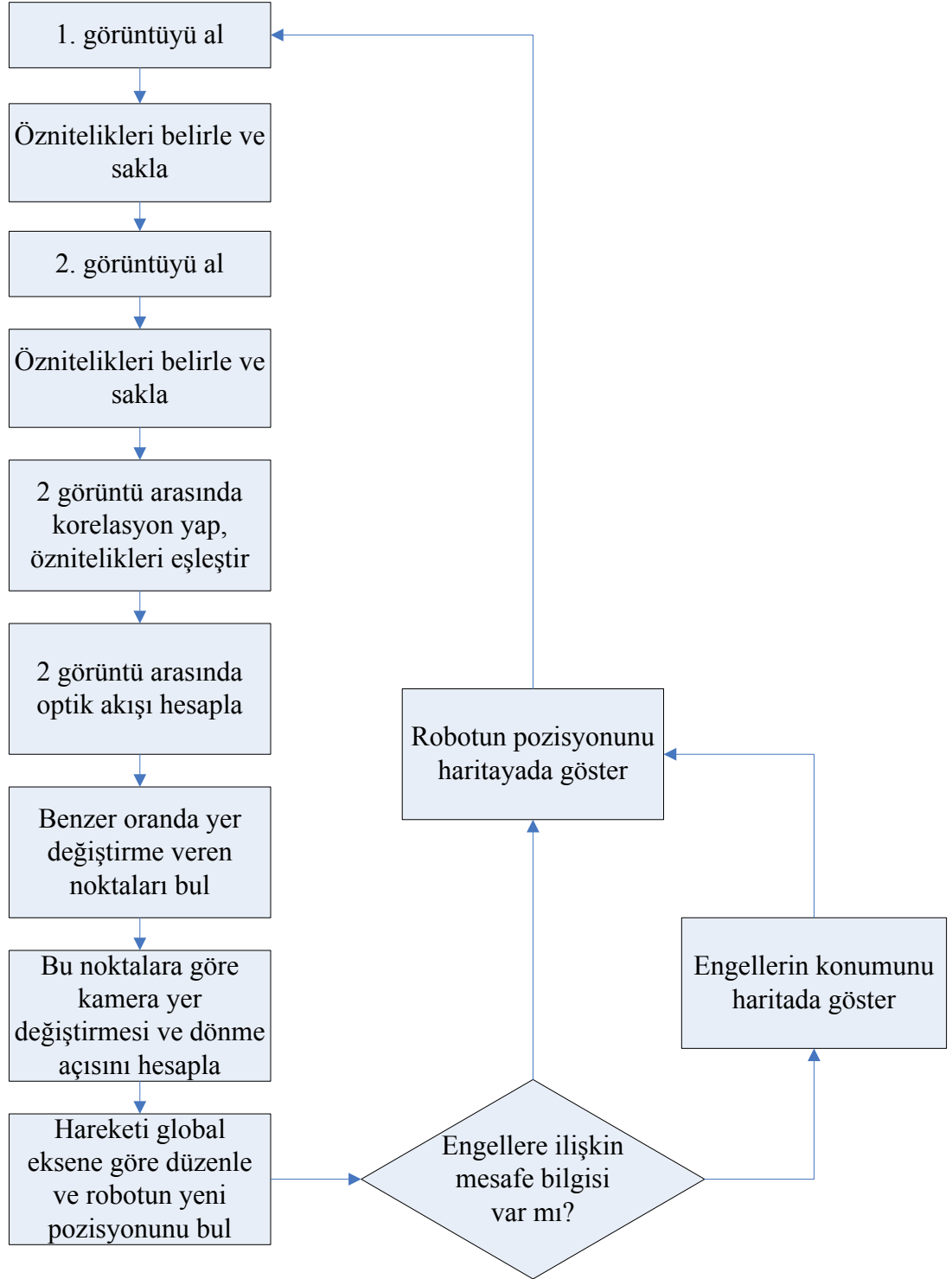
- 2. görüntüyü al.
- Öznitelikleri belirle ve sakla.
- 2 görüntü arasındaki korelasyon yap ve öznitelikleri ilişkilendir.
- 2 görüntü arasındaki optik akışı hesapla.
- Aynı miktar yer değiştirme veren noktaları bul.
- Kamera yer değiştirmesini hesapla.
- Dönme açısını hesapla.
- Hareketi global eksene göre düzenle.
- Yeni pozisyonu bul.
- Sensor verisi var mı kontrol et.
- Araçtan gelen sensor bilgilerini ve robot pozisyonunu haritaya uygula.

Bundan sonraki başlıklarda yukarıda sıralanan işlemler detaylı bir şekilde anlatılacaktır.

2.2.1. Öznitelik Belirleme

Öznitelik belirleme (İng. feature extraction) farklı görüntü özelliklerini belirleme ya da görüntü üzerindeki nesneleri tanımlama işlemidir. Bir görüntü çekildiği ortam ile ilgili birçok özelliği içerebilir. Ancak belirlenen özellikler her görüntüde aynı olmayabilir. Bazı uygulamalarda görüntüsü alınan nesnelerin kaba hatları veya açısal konumları yeterli olabilirken başka uygulamalarda nesnelere ait birçok özellik istenebilir. Örneğin yürüyen bir bant üzerinden geçen nesnelere ait belirlenen özellikler sınırlı iken uydudan çekilmiş görüntülerdeki ayrıntılar daha fazla önem kazanmaktadır.

Dijital bir görüntü bazen üç kanallı ve on milyon piksellik bir bilgiyi içerebilir. Bu nedenle görüntü üzerindeki öznitelikleri belirlerken hesaplama sürelerini minimize etmek için mümkün olduğu derecede bilgiyi süzmek gerekmektedir. Bu tarz yöntemlerden biri üç kanallı olarak alınan renkli bir görüntüyü gri ölçekli (İng. grey scale) denilen tek kanallı (siyah-beyaz) bir görüntüye dönüştürmektir. Ancak üzerinde çalışılacak olan görüntünün renkli olması söz konusuysa veya fazla detay isteniyorsa bu durumda farklı bir çalışma yapılması zorunludur. Bu gibi sebeplerden dolayı görüntü üzerindeki değişimleri en kısa sürede ve en iyi şekilde algılamak için üzerinde görüntüyü niteleyebilecek nokta veya bölgelerin belirlenmesi gerekmektedir. Buna öznitelik belirleme denir.



Şekil 2.2 : Tezde kullanılan görsel odometri algoritmasının akış şeması

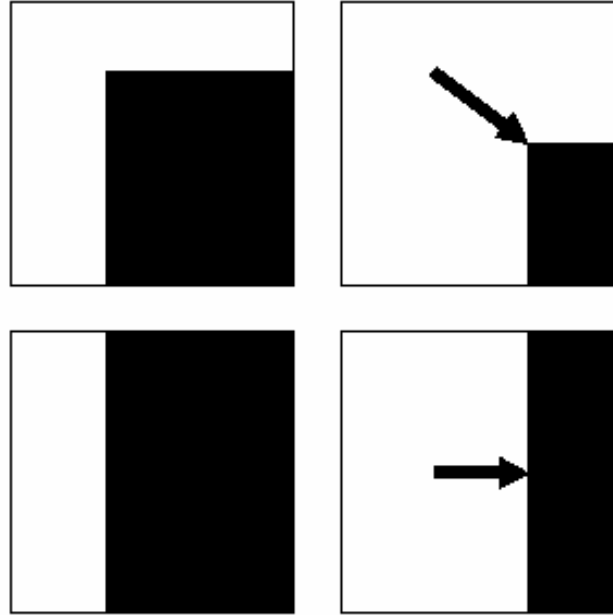
Öznitelik belirleme üç çeşittir:

- Spektral öznitelikler:
Renk, ton, spectral indeks oranı vs.
- Geometrik öznitelikler:
Kenarlar, hatlar, köşeler, yuvarlaklar vs.

- Yapısal öznitelikler:
Desenler, homojenlik vs.

Robotik görüntü işlemede ve daha birçok uygulamada geometrik öznitelikler yaygın bir şekilde kullanılmaktadır. Renk veya ton bilgisinin kullanımının ışık veya farklı sebeplerden dolayı sınırlı olması görüntü üzerindeki nesnelere ait kenar, köşe veya noktasal öznitelikler gibi geometrik özniteliklerin belirlenmesi daha etkili sonuçlar ortaya koymaktadır.

Bu çalışmada öznitelik kavramı yerine köşe veya köşelerin belirlenmesi şeklinde bahsedilecektir. Çünkü köşelerin belirlenmesi çok yaygın bir şekilde kullanılmakta ve özellikle görüntüler üzerindeki nesnelerin belirlenmesi ve tanımlanması konusunda büyük avantajlar sağlamaktadır. Zira bir nesnenin köşe noktalarını bilmek demek aynı zamanda o nesnenin kabaca şeklini veya yapısını bilmek demektir. Ayrıca farklı görüntülerin eşleştirilmesi konusunda da ileriki bölümlerde görüleceği üzere büyük avantajlar sağlamaktadır.



Şekil 2.3 : Görüntü üzerinde köşe ve kenarlar

Köşe, görüntü üzerinde yoğunluk fonksiyonu $f(x,y)$ 'nin x ve y yönlerindeki değişimin büyük olduğu noktalardır. Bu df/dx ve df/dy türevlerinin büyük olduğu anlamına gelir. Diğer taraftan kenar, görüntü üzerinde x veya y yönlerinden birinde $f(x,y)$ 'nin değişimi büyük iken diğer yönde değişimin az olduğu bölgeleri niteler. Şekil 2.3'te görüleceği üzere sol üstteki resimde görüntünün x ve y yönlerinde

hareket edildiği takdirde sağ üstte okla belirtilen noktada yoğunluk değişimleri fazladır. Bu bölge köşe olarak tanımlanır. Sol alttaki resimde ise x yönünde hareket edildiği takdirde sağ altta okla gösterildiği üzere bir kenar algılanmıştır. Çünkü y yönünde herhangi bir noktada yoğunluk değişimi olmamıştır.

Köşe belirleme ile ilgili ilk çalışmaları 1977 yılında Moravec [6] yapmıştır. Yukarıda bahsedilen ilkeye dayanan çalışma görüntüdeki nesnelerin dönmesi, boyut değişimi, ışık miktarı ve gürültüden olumsuz yönde etkilenme gibi konularda eksik kalmıştır. 1988 yılında Harris ve Stephans [7] hareket eden bir kameradan alınan görüntülerin eşleştirilmesi amacıyla Moravec'in yaklaşımını geliştirmişler ve günümüzde çok popüler olan ve Harris Operatörü diye anılan çalışmayı yapmışlardır. Harris operatörü, Moravec operatöründen oto-korelasyon fonksiyonunun hesaplama yönüyle farklılık gösterir. Lokal oto-korelasyon fonksiyonu görüntü üzerinde küçük adımlarla her bir piksel için bir lokal yapı matrisi (İng. local structure matrix) oluşturur:

$$M = \begin{pmatrix} A & C \\ C & B \end{pmatrix} \quad (\text{oto-korelasyon matrisi}) \quad (2.5)$$

A , B ve C görüntü düzlemi üzerinde farklı yönlerde hesaplanan türevlerdir.

$$A = \left(\frac{\partial I}{\partial x} \right)^2 \otimes w \quad (2.6)$$

$$B = \left(\frac{\partial I}{\partial y} \right)^2 \otimes w \quad (2.7)$$

$$C = \left(\frac{\partial I}{\partial x} \right) \cdot \left(\frac{\partial I}{\partial y} \right) \otimes w \quad (2.8)$$

I , herhangi bir (x, y) noktasının yoğunluğu ve w ise yumuşatma maskesidir. w genellikle Gauss filtresidir. Yumuşatma maskesi görüntü üzerinden gürültünün alınmasını sağlar. Türevler ise belirli bir alanda belli bir büyüklükte maske ile eğimin hesaplanması işidir. Prewitt ve Sobel operatörleri dijital görüntü işlemede sıklıkla kullanılan yöntemlerdir. Prewitt operatörü uygulama açısından Sobel'e göre daha

basit ancak gürültünün önlenmesi açısından daha az etkiye sahiptir. Gürültünün görüntüden çıkarılması başa çıkılması gereken önemli bir problemdir.

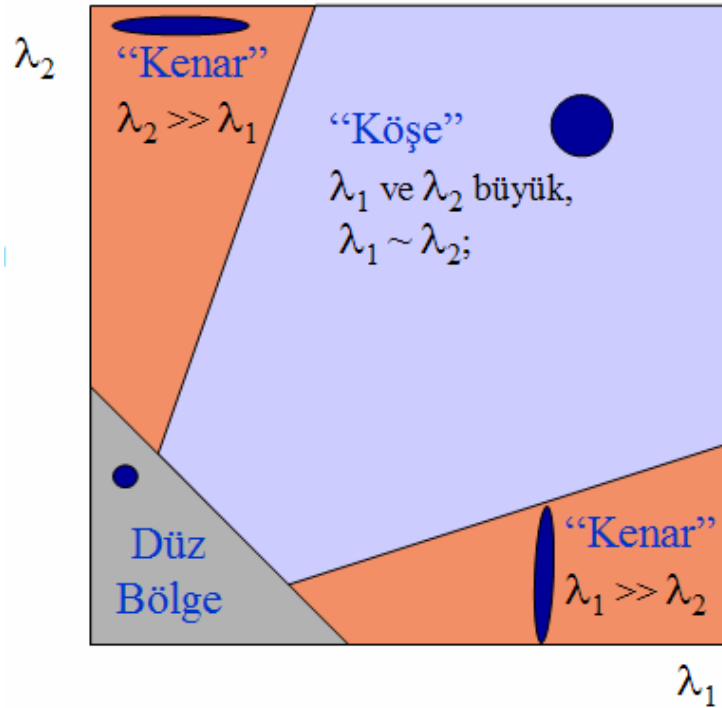
Bir sonraki aşama üzerinde işlem yapılan noktanın köşe olup olmadığının istatistiksel olarak belirlenmesi için gerekli olan değerin hesaplanmasını gerektirir.

$$c(x, y) = \det(M) - k \cdot (\text{trace}(M))^2 \quad (\text{kazanç değeri}) \quad (2.9)$$

$$\det(M) = \lambda_1 \cdot \lambda_2 = A \cdot B - C^2 \quad (2.10)$$

$$\text{trace}(M) = \lambda_1 + \lambda_2 = A + B \quad (2.11)$$

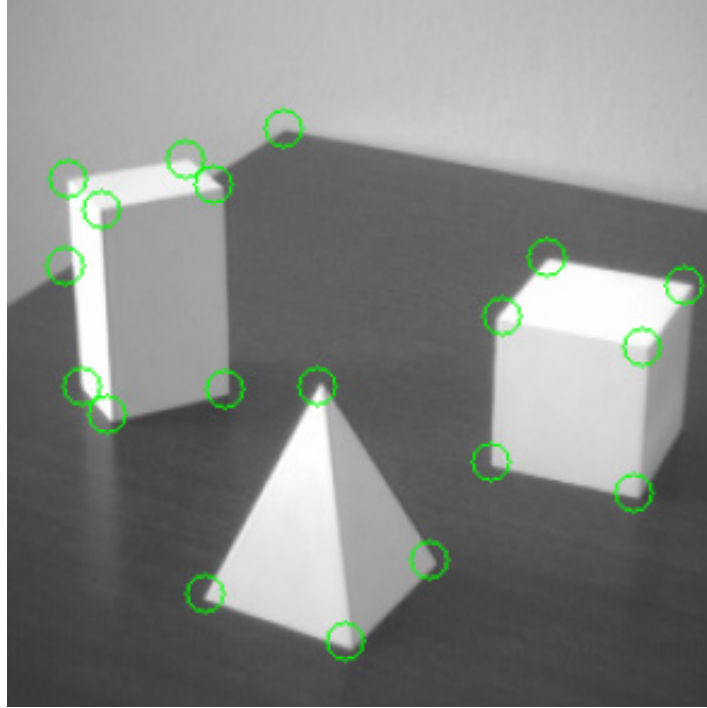
λ_1 ve λ_2 oto-korelasyon matrisinin özdeğerleri olup köşe noktalarının doğru belirlenmesini sağlar. k bir sabit katsayı olup 0 ile 0.25 arasında bir değer alır. k kullanılan operatörün köşe belirlemedeki hassasiyetini de gösterir. k 'nın küçük değeri olması operatörün köşeleri daha hassas algıladığını gösterir. Deneysel olarak 0.04 olarak seçilse de uygulamadan uygulamaya farklılık gösterebilir. $c(x, y)$ kazanç değeri gereksiz köşeleri elimine etmek için belli bir eşik (İng. threshold) değerinden büyük seçilmelidir [9,10]. Bu eşik değeri ne kadar büyükse o derece keskin köşeler elde edilir.



Şekil 2.4 : Özdeğerlerin anlamlandırılması

Yukarıda bahsedilen özdeğerler görüntü üzerinde çalışılan noktanın kenar veya köşe olup olmadığı konusunda fikir verir. Şekil 2.4'te de görüleceği üzere özdeğerlerin her ikisi de küçük değerli ise oto-korelasyon matrisi düz ve incelenen görüntü kısmında yoğunluk değişimi çok azdır denir ve kenar veya köşe olarak algılanmaz. Eğer özdeğerlerin ikisi de büyük değerli ise görüntünün o kısmı için oto-korelasyon fonksiyonu keskin bir değişim yaşamıştır. Bu bölge köşe olarak tanımlanır. Özdeğerlerden herhangi biri büyük ve diğeri küçükse bir yönde yoğunluk değişimi fazla diğere yönde azdır. Bu bölge kenar olarak adlandırılır.

Yukarıda bahsedilenler ışığında tezle ilgili çalışmalara geçilmeden önce Şekil 2.5 ve Şekil 2.6'da görülen basit, orta seviye olarak nitelendirilebilecek görüntüler üzerinde köşe belirlmeleri yapılmış ve Harris operatörünün performansı incelenmiştir.



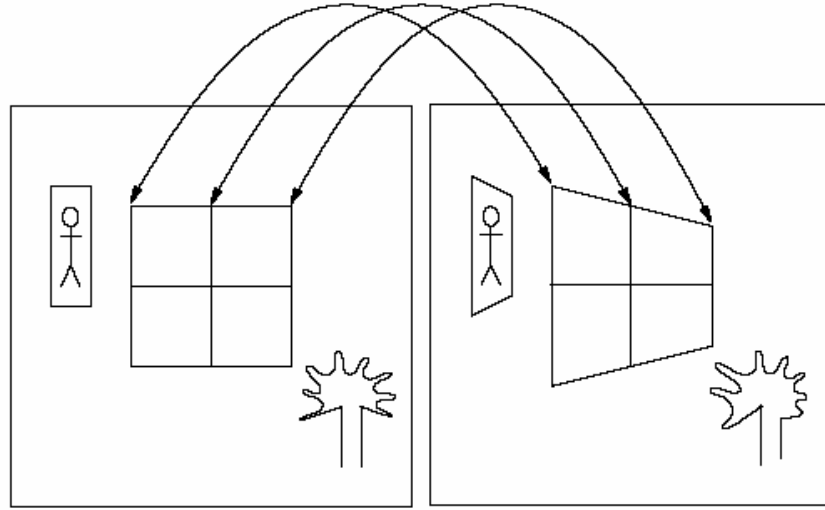
Şekil 2.5 : Prizma, piramit ve küpten oluşan bir görüntüdeki köşe noktalarının bulunması

2.2.2. Öznitelik Eşleştirme

Üzerinde çalışılan görüntüde bulunan her bir öznitelik Şekil 2.7'de görüldüğü gibi bir sonraki görüntüde aranır. Buna öznitelik eşleştirme denir. Bu çalışmada öznitelikler daha önce anlatıldığı üzere köşe noktalarıdır. Dolayısıyla her bir görüntü çiftinde, ilk görüntüde bulunan köşelerin bir sonraki görüntüde bulunan köşelerle eşleştirilmesi problemi bu bölümde ele alınacaktır.



Şekil 2.6 : Bir koridorun çekilmiş görüntüsü üzerinde bulunan köşe noktaları



Şekil 2.7 : Benzer iki görüntü üzerindeki özniteliklerin eşleştirilmesi

Köşe noktalarının farklı görüntüler arasında eşleştirilmesi için Normalize Çapraz Korelasyon (İng. Normalized Cross Correlation - NCC) yöntemi yaygın bir şekilde kullanılmaktadır. Bu yöntem, denklem 2.7’den de anlaşılağı üzere ilk görüntü üzerinde bulunan özniteliklerin içerisinde bulunduğı bir W görüntü (İng. window)

penceresinin diğer görüntü üzerinde bulunan öz niteliklerin içerisinde bulunduğu görüntü penceresi ile karşılaştırması ilkesine dayanır.

$$NCC(I_1, I_2) = \frac{\sum_w (I_1(x) - \bar{I}_1) \cdot (I_2(x) - \bar{I}_2)}{\sqrt{\sum_w (I_1(x) - \bar{I}_1)^2 \cdot \sum_w (I_2(x) - \bar{I}_2)^2}} \quad (2.12)$$

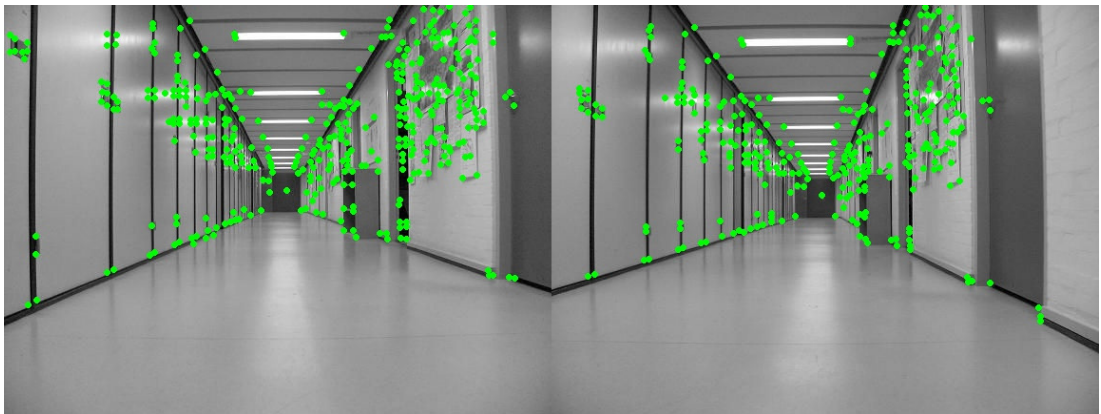
$$\bar{I}_1 = \frac{1}{N} \cdot \sum_w I_1(x) \quad (2.13)$$

$$\bar{I}_2 = \frac{1}{N} \cdot \sum_w I_2(x) \quad (2.14)$$

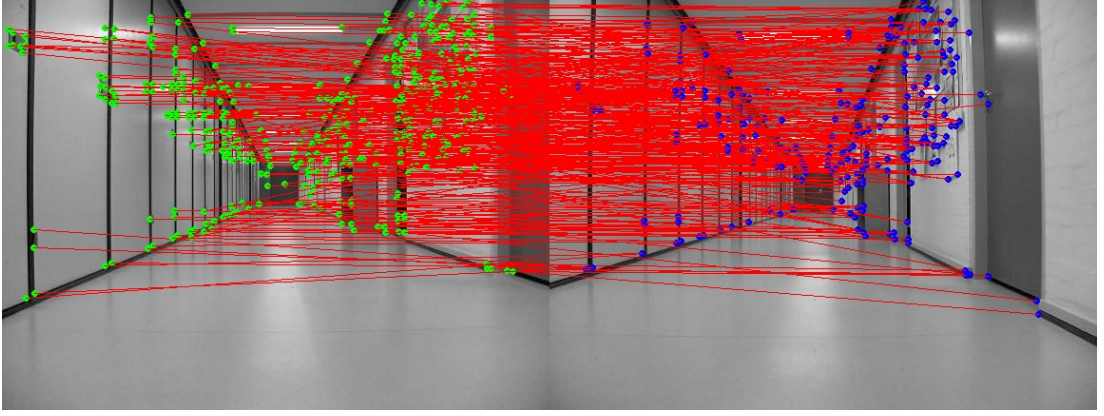
Köşe noktalarının eşleştirilmesi şu işlemler sonucu gerçekleşir:

- Her bir köşe noktası için görüntü penceresi oluşturulur.
- Bu pencere ile diğer görüntüdeki köşeler için oluşturulan görüntü pencereleri arasındaki NCC değerini hesaplanır.
- Görüntüdeki her bir köşe noktası için hesaplanan NCC değerlerinden en büyük olanı alınır ve en büyük NCC değerini veren köşeler eşleştirilir.

Yukarıda anlatılanlar ışığında yapılan çalışmalar farklı görüntüler üzerinde test edilmiştir. Bu çalışmalardan biri Şekil 2.8’de görülebilir. Şekil 2.8’de bir koridorun birbirine yakın 2 farklı noktadan alınmış görüntüleri üzerinde köşe noktaları belirlenmiştir. Köşelerin eşleştirilmesi işlemi ise Şekil 2.9’da görülebilir.

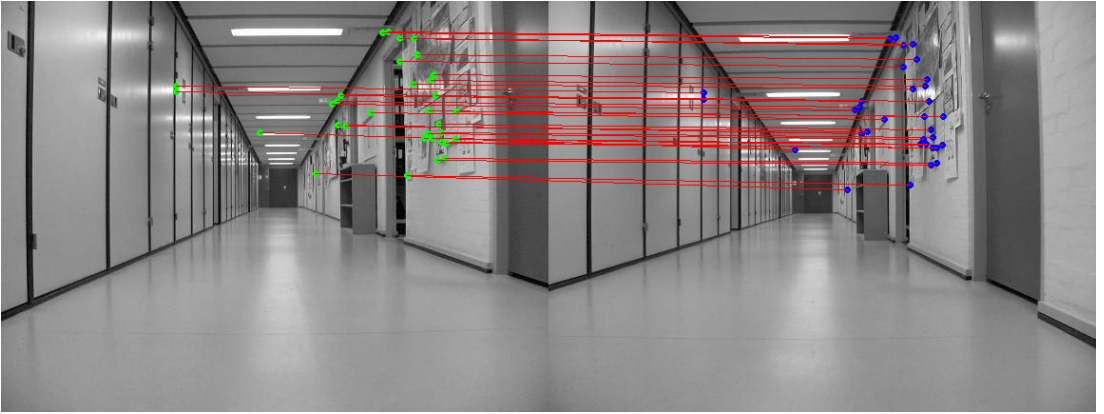


Şekil 2.8 : İki farklı görüntü üzerinde bulunan köşe noktaları



Şekil 2.9 : Görüntüler üzerindeki köşelerin eşleştirilmesi

Uygulamalarda bazı durumlarda bulunan tüm köşe noktaları değil yüksek eşleştirme oranına sahip köşe noktaları alınabilir. Yukarıda köşeleri eşleştirilmiş görüntüde yanlış eşleştirmeler atılmış ve en yüksek eşleştirme değerine sahip noktalar Şekil 2.10’da gösterilmiştir.



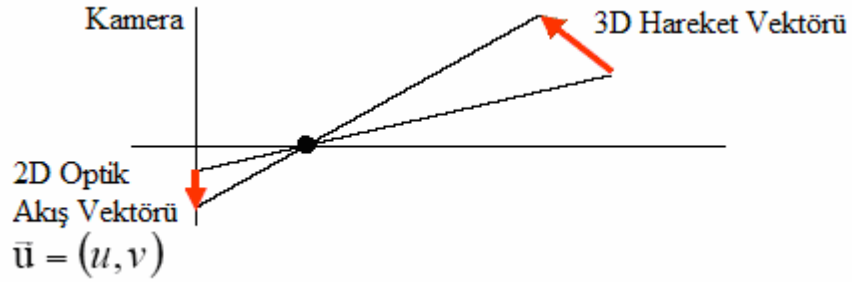
Şekil 2.10 : Yüksek eşleşme oranına sahip köşe noktaları

Görüntüler üzerinde bulunan özniteliklerin eşleştirilmesi (Şekil 2.9) örnek görüntülerde ve kameradan gerçek zamanlı olarak alınan görüntüler üzerine uygulandığında programın bir görüntü çiftini işleme zamanı beklenenden uzun sürdüğü için optik akış yöntemi bu çalışmada tek başına kullanılmıştır.

2.2.3. Optik Akış

Optik Akış terim olarak görüntü düzlemindeki her bir noktanın hareketinin incelenmesi anlamına gelir [25]. Nesne tanıma, bilgisayarlı görü ve görüntü işleme başlıca uygulama alanlarıdır. Özellikle hareket algılama uygulamalarında sıklıkla kullanılır. Optik akış, robotik görüntüleme içerisinde ise bir görüntü içerisindeki nesnelerin hareketlerini inceler. Bir robot üzerinde bulunan kamera vasıtası ile alınan

video ya da belli bir sıra ile çekilmiş görüntülerle robotun bulunduğu ortam içindeki nesnelerin hızlı bir şekilde eşleşmeleri ve bu eşleşmelerle nesnelerin görüntü düzlemi üzerindeki hareketlerini veya yer değiştirmelerini vektörel olarak ifade etmek için kullanılır [25]. Şekil 2.11’de 3-boyutlu bir uzayda hareket eden bir noktanın görüntü düzlemindeki hareketi gösterilmiştir.



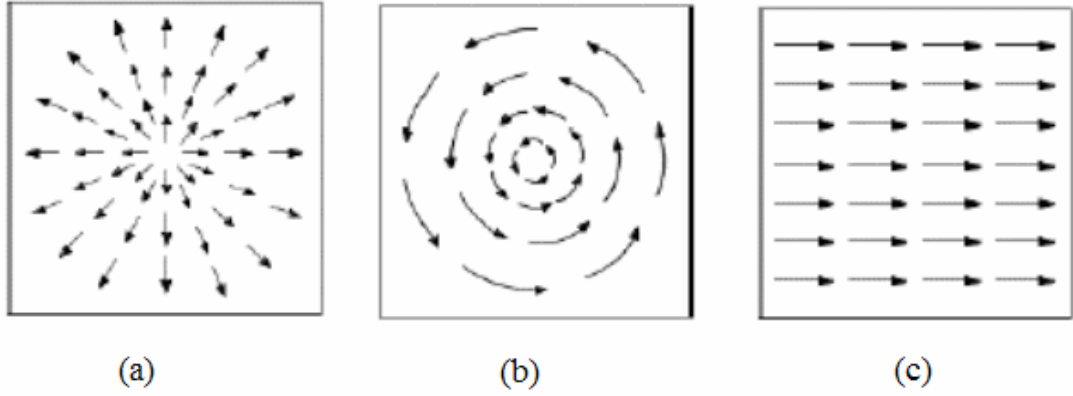
Şekil 2.11 : 3-boyutlu uzaydaki bir hareket vektörünün kamera eksenleri üzerinde 2-boyutlu olarak gösterimi

Robotun 3-boyutlu uzayda hareket ettiği bir ortamdan alınan görüntüler üzerindeki optik akış vektörleri düzgün bir şekilde incelendiğinde robotun bulunduğu ortam ve robotun kendi hareketi ile ilgili olarak aşağıdaki şu sorulara cevaplar kolaylıkla bulunabilir [26].

- Kaç tane hareket eden cisim var?
- Hangi yönlere hareket ediyorlar?
- Düzgün mü yoksa belirli bir fonksiyona göre mi hareket ediyorlar?
- Ne kadar hızlılar?

Yukarıda sorulan sorulara örnek olması açısından Şekil 2.12’de görüntü düzlemi üzerinde bulunan noktaların hareketleri incelendiğinde robotun kendi hareket yönü için cevap bulunabilir. Sol tarafta bulunan görüntü üzerindeki noktaların bir kaynaktan her yöne doğru dağılıyor gibi gözükmesi aslında kameranın ileri doğru hareket ettiğini ve uzaktaki nesnelerin kameraya daha da yaklaştığını göstermektedir. Ortadaki görüntü üzerinde bulunan noktalar ise saatin tersi yönünde dairesel bir hareket yapıyor gözükmektedir. Bu aslında ortamdaki nesnelerin sabit olduğu farz edilirse kameranın saat yönüne dairesel bir dönüş yaptığı anlamına gelir. Kamera saat yönüne doğru bir dönüş hareketi yapsa dahi ortamdaki nesnelerin kamera göre bağıl hızları düşünüldüğünde görüntü üzerindeki noktaların neden saatin ters yönüne

doğru döndükleri sorusuna cevap bulunabilir. Aynı şekilde, sağda bulunan görüntü üzerindeki noktalar sağ tarafa doğru lineer bir hareket yapıyor gözükmektedirler. Ancak, yine nesnelerin kameraya göre bağlı hızları düşünüldüğünde kameranın sola doğru lineer bir hareket yaptığı anlaşılmaktadır.



Şekil 2.12 : Kameranın hareketi referans alınarak optik akış vektörlerinin hareket yönlerinin anlamlandırılması: a) ileri yönde hareket, b) kendi etrafında dönme hareketi, c) sola doğru hareket

Optik akış görüntü üzerindeki her bir nokta için $\vec{u} = (u, v)$ şeklinde hız vektörlerini bulmayı amaçlamaktadır. Bu hız vektörleriyle:

- Noktaların görüntü üzerinde ne kadar hızlı hareket ettiği,
- Noktaların hangi yöne doğru hareket ettiği

bulunur. Bu hesaplamaları yaparken kullanılan yöntemlerde alınan sıralı görüntüler arasında renk ve ton değişiminin çok yoğun olmadığı kabul edilir. Günümüzde en yaygın olarak Lucas ve Kanade [16] tarafından temelleri atılan Lucas-Kanade Optik Akış algoritması kullanılmaktadır.

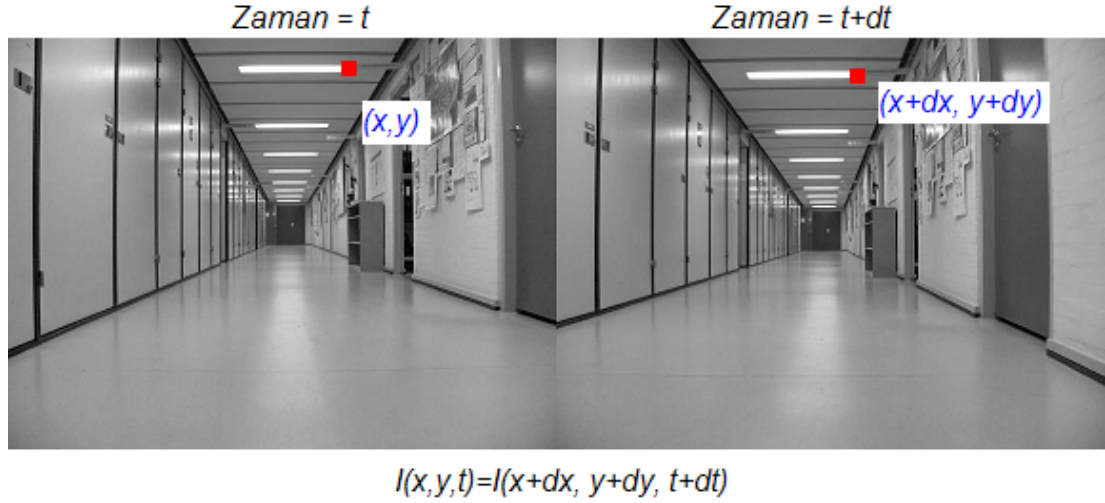
Görüntüler üzerinde yoğunluğun eşit olduğu kabul edilirse, görüntü üzerindeki bir noktanın yoğunluk fonksiyonu Şekil 2.13'te gözüktüğü üzere:

$$I(x, y, t) = I(x + \partial x, y + \partial y, t + \partial t) \quad (2.15)$$

şeklinde olur. Yer değiştirmenin küçük olduğu kabul edilerek birinci dereceden Taylor açılımı yapılırsa:

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} \cdot dx + \frac{\partial I}{\partial y} \cdot dy + \frac{\partial I}{\partial t} \cdot dt \quad (2.16)$$

şeklinde ifade edilir. Bu ifadede yüksek mertebeden terimler ihmal edilir. Basit bir gösterimle yukarıdaki denklemden:



Şekil 2.13 : Küçük zaman aralığında alınan iki görüntüden ilki üzerindeki bir noktanın diğer görüntü üzerindeki konumu ve yer değiştirme miktarı

$$\frac{\partial I}{\partial x} \cdot dx + \frac{\partial I}{\partial y} \cdot dy + \frac{\partial I}{\partial t} \cdot dt = 0 \quad (2.17)$$

sonucu ortaya çıkar. Burada $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ yoğunluğun x ve y yönlerindeki değişimini,

$\frac{\partial I}{\partial t}$ ise yoğunluğun zamanla değişimini ifade eder. Sembolik gösterimle aynı denklem:

$$I_x \cdot dx + I_y \cdot dy + I_t \cdot dt = 0 \quad (2.18)$$

şeklinde gösterilir. Denklemin her iki tarafı da dt ile bölünürse:

$$I_x \cdot \frac{dx}{dt} + I_y \cdot \frac{dy}{dt} + I_t = 0 \quad (2.19)$$

denklemini elde edilir. Genel gösterimle aynı denklem:

$$I_x \cdot u + I_y \cdot v = -I_t \quad (2.20)$$

şeklinde ifade edilir. Bu ifadede u ($u = \frac{dx}{dt}$) ve v ($v = \frac{dy}{dt}$) görüntü üzerindeki noktaların sırayla x ve y yönündeki hız değişimini ifade etmektedir. 2.20 denklemi daha genel bir gösterimle:

$$\nabla I \cdot \vec{V} = -I_t \quad (2.21)$$

şeklinde ifade edilir. Ancak 2.21 denkleminde görüleceği üzere bir denklem ve iki bilinmeyen vardır. Dolayısıyla denklem bu haliyle çözülemez ve bu duruma açıklık problemi (İng. Aperture Problem) denir. Lucas-Kanade Optik Akış algoritması bu problemi ortadan kaldırmak için şu yaklaşımı ortaya koyar:

x ve y yönündeki akışların $m \times m$ ($m > 1$) boyutunda küçük bir pencere içerisinde olduğu kabul edilir ve bu pencere üzerindeki noktalar 1'den n 'e kadar ($n = m^2$) numaralandırılırsa aşağıda gösterilen bir dizi denklemler bulunur.

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \cdot & \cdot \\ \cdot & \cdot \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \cdot \\ \cdot \\ -I_{tn} \end{bmatrix} \quad (2.22)$$

Daha genel bir ifadeyle yukarıdaki denklemler:

$$A \vec{V} = -b \quad (2.23)$$

şeklinde gösterilir. Burada $\|A\vec{V} - b\|^2$ farkını minimuma indirmek için en küçük kareler yöntemi uygulanır. Bu amaçla:

$$A^T (A\vec{V}) = A^T (-b) \quad (2.24)$$

yada

$$\vec{V} = (A^T A)^{-1} A^T (-b) \quad (2.25)$$

denklemiyle hız vektörü hesaplanır. Burada:

$$(A^T . A) = \begin{bmatrix} \sum I_x^2 & \sum I_x . I_y \\ \sum I_x . I_y & \sum I_y^2 \end{bmatrix} \quad (2.26)$$

ifadesinin özdeğerleri (x, y) noktasının davranışı hakkında fikirler verir. Matrisinin tersinin alınabilmesi için özdeğerlerinin sıfırdan farklı olması gerekmektedir. Bu matristeki özdeğerlerin yapısı ise görüntü üzerindeki noktalar hakkında şu bilgileri verir.

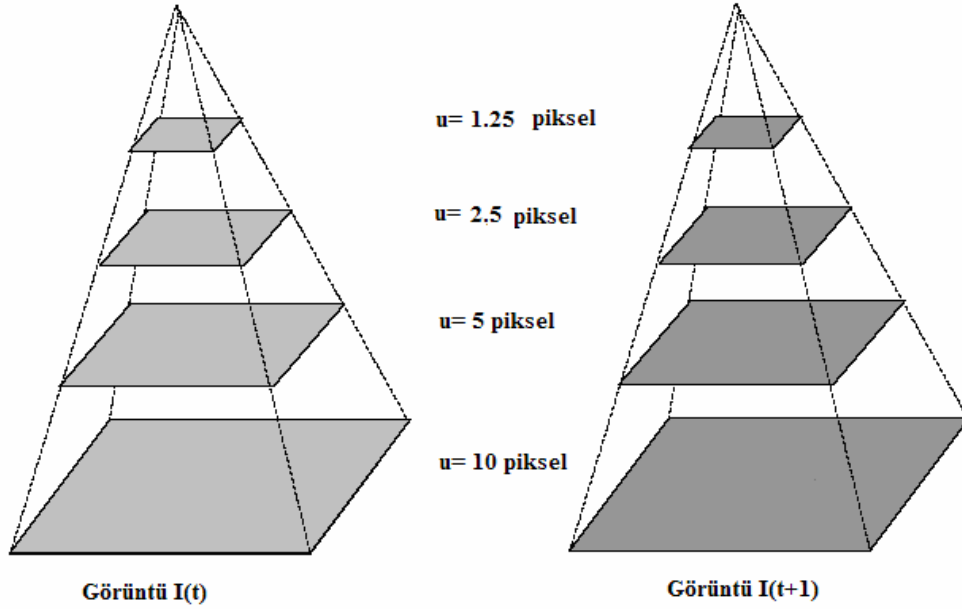
- Özdeğerlerden biri sıfır ise görüntü üzerindeki nokta bir kenar ifade eder.
- Özdeğerlerin her ikisi de düşük değerli ise nokta görüntü üzerinde homojen bir bölgededir.
- Özdeğerler büyükse nokta bir köşe ifade eder ve her yönde yoğunluğun değişimi büyük olur.

Lucas-Kanade Optik akış algoritmasının performansını artırmak için Bouget [27] tarafından Piramidal Lucas-Kanade Optik Akış Algoritması geliştirilmiştir. Bu algorithmada görüntü üzerinde seçilen özniteliklerin hareketinin boyutuyla orantılı olarak işlem penceresi boyutu değişir. Şekil 2.14'te örneği görüleceği üzere hareketi incelenen özniteliğin hareketi çözünürlüğü değiştirilmiş görüntüler üzerinde iteratif bir yaklaşımla hesaplanır. Daha büyük hareketleri daha hassas algılamak için görüntü büyüklüğü piramit şeklinde bir değişim göstermektedir [28].

Özetle görüntü işleme teknikleriyle yapılan işlemler şu şekildedir: Kameradan alınan görüntülerden ilki üzerinde en iyi köşe noktaları belirlenir, belirlenen bu köşe noktalarının koordinatları tutularak diğer görüntü üzerinde belirlenen bu koordinatlar etrafındaki bir pencere içerisinde köşelerin hareketi incelenir. Hareketin incelendiği pencere büyüklüğü küçükten başlayarak en az hatayı veren noktayı bulana kadar piramidal olarak değişir. Bu işlemler bir döngü içerisinde tekrarlanarak yapılır [29]. Sonuç olarak ilk görüntü üzerinde bulunan köşelerin diğer görüntü üzerindeki yer değiştirme miktarları robotun gerçek hareketini bulmak üzere aşağıda anlatıldığı şekilde kullanılmıştır.

Yukarıda anlatılan denklemlerle görüntü üzerindeki $\vec{V} = [u \ v]^T$ anlık hız vektörleri bir takım matematiksel yaklaşımlarla kameranın hareketini verebilir. Bu çalışmada görüntü üzerinde bulunan özniteliklerin optik akış yöntemiyle elde edilen hız vektörlerinin yer düzlemine izdüşümü ve bu izdüşümler sonucu oluşan noktaların her

bir görüntü çifti arasında yer değişiminin ortalama değerleriyle kameranın, dolayısıyla robotun, hareketi modellenmiştir. Bu amaçla Şekil 2.15'te görüldüğü üzere robot platform üzerine bir kamera monte edilerek robotun hareket ettiği yüzeyden aldığı görüntüler üzerinde bulunan özniteliklerin yönlerindeki değişim ve yer değiştirme miktarları değerlendirilmiş ve bu değerlendirmelerle gerçek ortama göre robotun hareket yönü ve miktarı elde edilmiştir.



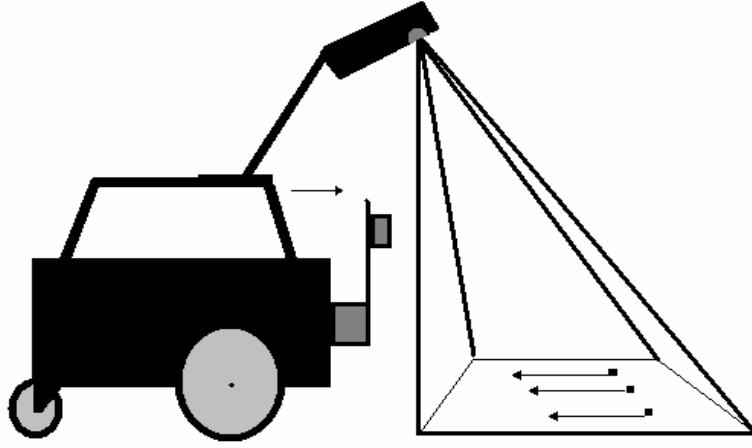
Şekil 2.14 : Optik akışı incelenen bir noktanın görüntüler üzerinde arandığı pencerelerin büyüklüğünün piramit şeklinde gösterimleri [28]

Robotun hareketinin görüntüler üzerinden elde edilebilmesi için bazı ön kabuller gerekmektedir. Bunlar:

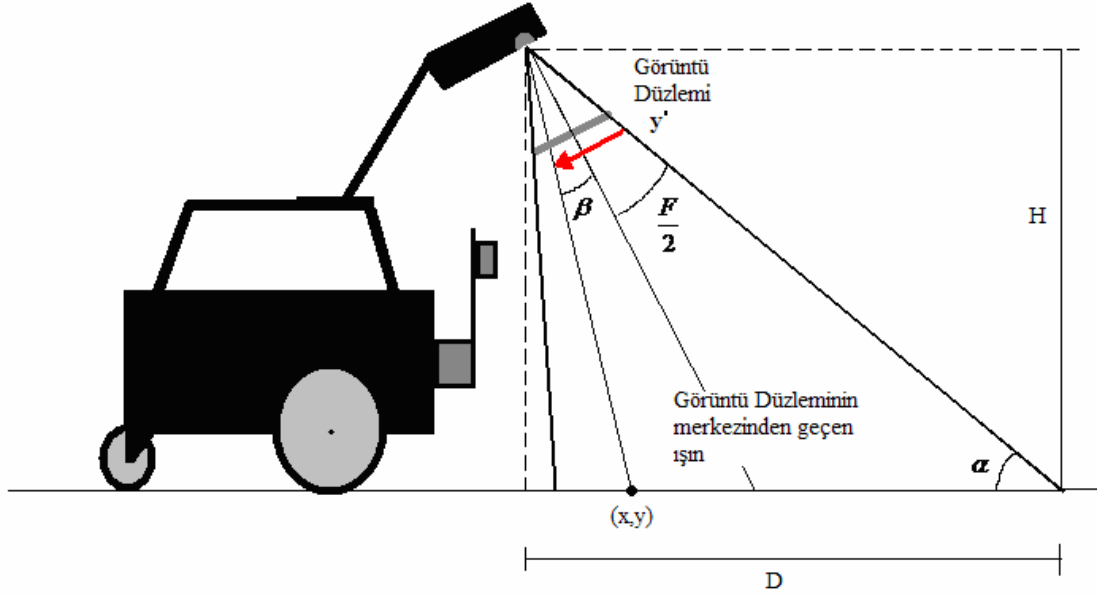
- Robotun önünde ultrasonik sensorun algılayamadığı yapıda ve hareketini engelleyecek nesnelerin olmaması,
- Robotun çok parlak veya pürüzsüz yüzeylerde hareket etmemesi,
- Robotun hareketi sırasında kameranın önünden hareketli nesnelerin geçmemesi

şeklinde sıralanabilir.

Şekil 2.16'da robot platform üzerinde bulunan kameranın görüntü düzleminin robotun hareket ettiği yüzey üzerine iz düşümü görülmektedir. Bu şekilde gösterilen açısal değerler ve mesafeleri kullanarak basit trigonometrik denklem hesaplamalarıyla robotun hareket miktarı bulunabilir.



Şekil 2.15 : Hedef robot platformu



Şekil 2.16 : Görüntü koordinatlarından yer düzlemine göre dikey koordinatın hesaplanmasını sağlayan dönüşüm

Görüntü düzlemi üzerinde (x', y') noktasında bulunan bir özneliğin yer düzlemindeki koordinatları (x, y) olarak kabul edilirse x , takip edilen özneliğin yataydaki koordinatını, y ise dikeydeki koordinatını verir. Kameranin lensinin yerden yüksekliği H ve kameranın yer düzleminde görüntü düzlemine aktarabildiği dikey mesafe D el ile ölçülür. Buradan:

$$\tan(\alpha) = \frac{H}{D} \quad (2.27)$$

açısal değeri bulunur. Kamera lensinden görüntü düzleminin merkezine kadar olan mesafe h , görüntü düzlemi merkezinden geçen ışın baz alınarak:

$$h = \frac{Y}{2} \cdot \frac{1}{\tan(\frac{F}{2})} \quad (2.28)$$

şeklinde bulunur. Bu ifadede Y görüntü düzleminde dikey piksel sayısıdır. Örneğin 640x480 piksellik bir kamera görüntüsünde bu değer 480 pikseldir. F ise dikey mesafede kameranın görme açısıdır. β açısal değeri ise:

$$\tan(\beta) = \frac{y' - \frac{Y}{2}}{h} \quad (2.29)$$

denkleminde,

$$\tan(\beta) = \frac{y' - \frac{Y}{2}}{\frac{Y}{2} \cdot \frac{1}{\tan(\frac{F}{2})}} = \frac{(2 \cdot y' - Y)}{Y} \cdot \tan(\frac{F}{2}) \quad (2.30)$$

şeklinde hesaplanır. Denklemden kullanılan y' değeri ise hareketi incelenen öznenin görüntü düzleminde dikey koordinatını simgelemektedir. Hareketi incelenen öznenin görüntü düzleminde karşılık geldiği noktanın dikey mesafedeki uzaklığı y :

$$y = \frac{H}{\tan(\alpha + \beta + \frac{F}{2})} \text{ (cm)} \quad (2.31)$$

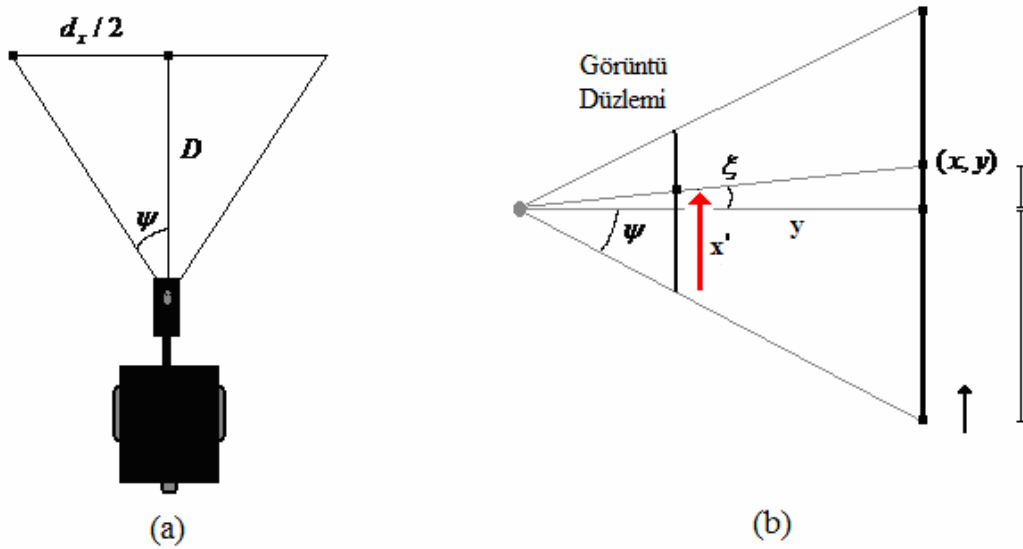
şeklinde hesaplanır. Şekil 2.17'den yola çıkarak kameranın yatay görme açısı $2 \cdot \psi$ ve Z yataydaki toplam piksel sayısı olmak üzere:

$$\tan \psi = \frac{d_x}{2 \cdot D} \quad (2.32)$$

$$\tan \xi = \frac{x' - \frac{Z}{2}}{\frac{Z}{2} \cdot \frac{1}{\tan \psi}} = \frac{2 \cdot x' - Z}{Z} \cdot \tan \psi \quad (2.33)$$

$$x = y \cdot \tan \psi + y \cdot \tan \xi = y \cdot \tan \psi \cdot \frac{2 \cdot x'}{Z} \quad (2.34)$$

şeklinde x noktası hesaplanır.



Şekil 2.17 : Robotun x yönünde aldığı mesafeyi hesaplamak için kullanılan yaklaşım: a) yataydaki görme açısı, b) görüntü düzlemi ve gerçek düzleme göre pozisyon

Robotun hareket miktarı hesaplanırken robot üzerinde bulunan kameradan alınan görüntü çiftlerinden ilki üzerindeki öznitelikleri ve bu özniteliklerin yer düzlemine izdüşümleri yukarıdaki denklemler uyarınca bulunur. Yer düzlemindeki noktaların robota olan mesafeleri (y değerleri) bulunduktan sonra aynı işlemler ikinci görüntü üzerine uygulanarak özniteliklerin robota olan mesafeleri bulunur. Bulunan bu mesafelerdeki fark yer değiştirme miktarlarını verir ve bu yer değişim miktarlarının ortalama değeri robotun hareketini verir. Görüntüler üzerinde bulunan ve eşleşen öznitelik sayısı n olarak ve her bir görüntü çifti başına işlemler için geçen zaman ortalama Δt olarak kabul edilirse:

$$v_y = \frac{\sum_{i=1}^n (y_{2,i} - y_{1,i})}{n} \cdot \frac{1}{\Delta t} \quad (2.35)$$

ve

$$v_x = \frac{\sum_{i=1}^n (x_{2,i} - x_{1,i})}{n} \cdot \frac{1}{\Delta t} \quad (2.36)$$

şeklinde ortalama hız vektörü bulunmuş olur. Aynı zamanda robotun açısal değişimi:

$$\phi = \tan^{-1} \left(\frac{v_x}{v_y} \right) \quad (2.37)$$

Denklemden hesaplanır. Aynı şekilde birim işlem zamanı Δt boyunca alınan yol L bulunur.

$$L_y = \frac{\sum_{i=1}^n (y_{2,i} - y_{1,i})}{n} \quad (2.38)$$

$$L_x = \frac{\sum_{i=1}^n (x_{2,i} - x_{1,i})}{n} \quad (2.39)$$

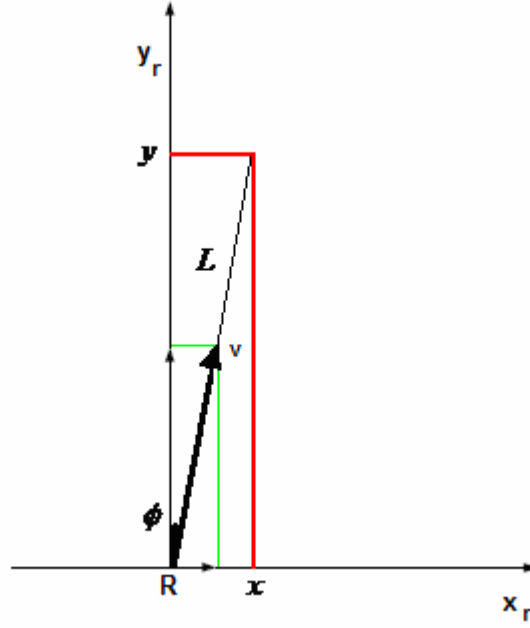
$$\vec{L} = \vec{L}_x + \vec{L}_y \quad (2.40)$$

2.2.4. Yer Değiştirme ve Dönme Hareketlerinin Hesaplanması

Robot, global eksene (sabit eksen) göre herhangi bir $[x_R \ y_R]^T$ pozisyonundan sabit bir v hızıyla geçtikten Δt kadar bir süre sonra, global düzlemin x – ekseninde Δx kadar, y – ekseninde Δy kadar yol alır. Şekil 2.18’de görüldüğü üzere bu süre içinde robot hareketli düzlemin $x_r - y_r$ düzleminde ϕ açısıyla L kadar yol alır. Hesaplamalar şu şekilde yapılır:

$$x = L \cdot \sin \phi = L_x \quad (2.41)$$

$$y = L \cdot \cos \phi = L_y \quad (2.42)$$



Şekil 2.18 : Robotun kendi hareket eksenlerine göre anlık hız ve yer değiştirme vektörleri

Burada x ve y hareketli eksen üzerindeki noktalardır. Hareketli eksen Şekil 2.19 üzerinde görüleceği üzere $[x_R \ y_R]^T$ noktalarına ötelenmiş olan global düzleme göre saat yönünde $(90 - \theta)$ derece kadar döndürülmüştür. Koordinat döndürme yöntemine göre robotun Δt kadar sürede aldığı yol ve açısal değişimi düzenlenirse:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \cos(90 - \theta) & \sin(90 - \theta) & 0 \\ -\sin(90 - \theta) & \cos(90 - \theta) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} \quad (2.43)$$

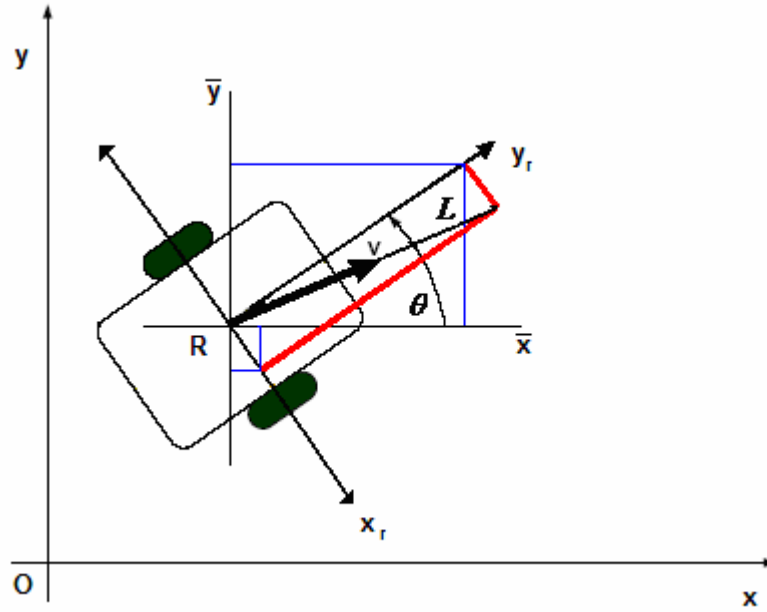
şeklinde hesaplanır.

$\theta > 90$ durumunda saatin ters yönünde $(\theta - 90)$ derece dönmüş olur ve bu dönüşler için aşağıdaki denklem kullanılır.

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \cos(\theta - 90) & -\sin(\theta - 90) & 0 \\ \sin(\theta - 90) & \cos(\theta - 90) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} \quad (2.44)$$

Ayrıca robot mevcut pozisyonunda sağa doğru hamle yapıyorsa ϕ açısı sıfırdan büyük, sola doğru hamle yapıyorsa ϕ açısı sıfırdan küçük alınır. Bundan dolayı sağa

doğru hamle θ açısının küçülmesine, sola doğru hamle ise θ açısının büyümesine yol açar. Bu durumda yukarıdaki denklemler yerine aşağıdaki denklemler kullanılır:



Şekil 2.19 : Robotun global düzleme ve kendi düzlemine göre yer değiştirmesi

$$x = L \cdot \sin(-\phi) = -L \cdot \sin \phi \quad (2.45)$$

$$y = L \cdot \cos(-\phi) = L \cdot \cos \phi \quad (2.46)$$

Denklem aşağıdaki şekliyle düzenlenir.

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \cos(90 - \theta) & \sin(90 - \theta) & 0 \\ -\sin(90 - \theta) & \cos(90 - \theta) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ -\phi \end{bmatrix} \quad (2.47)$$

2.3. Engel Algılama

Robotlar içinde bulundukları ortamı üzerlerinde bulunan sensorlardan gelen bilgilere göre algılar. Bundan dolayı robotların içinde bulundukları ortamın yapısını doğru algılamaları için üzerlerinde bulunan sensorların hassas ve ölçüm kapasitelerinin yüksek olması gerekmektedir. Ayrıca algılanan engel veya cisimleri harita üzerinde doğru konumlandırmak için sensorlardan gelen bilgilerin hareket modeline uygun bir şekilde işlenmesi gerekmektedir. Bu bölümde tez çalışmasında engelleri algılamak

için kullanılan sensor yapısı ve harita üzerinde engellerin konumlandırılması anlatılacaktır.

Robot, ön tarafına monte edilmiş olan Devantech SRF05 ultrasonic mesafe ölçüm sensoruyla etrafındaki cisimleri algılayabilir. Sensor servo motorun şaftına monte edildiğinden sensor sağa veya sola 90 dereceye kadar dönme yeteneğine sahiptir. Dolayısıyla robot ön tarafını 180 derece tarayarak engelleri algılayabilir. Sensorun hassasiyeti oldukça yüksektir. Sensor 1.5 metrelik bir mesafe içerisinde maksimum %1.33 hata göstermekte, mesafe kısaltıkça daha hassas sonuçlar vermektedir [4].

Şekil 2.20’de robotun engel tarama modunda iken anlık pozisyonuna göre tarayabileceği alan ve herhangi bir engel algıladığında bu engelin global eksene göre izdüşümleri Şekil 2.21’de gösterilmiştir. Robot, 180 derecelik bir alanı tarayabilme kapasitesine sahiptir. Robotun global eksene (sabit eksen) göre herhangi bir $[x_R \ y_R]^T$ pozisyonunda kendi koordinat düzlemine göre α açısıyla d mesafesinde bulunan bir engelin global eksene göre pozisyonu aşağıda gösterilen denklemlere göre hesaplanır. Denklemlerden anlaşılacağı üzere robotun global eksen üzerinde bir (x_R, y_R) konumundayken algıladığı ve hareketli eksen üzerinde (x, y) noktasında bulunan engelin sabit eksenindeki konumu koordinat dönüşümü yapılarak bulunur. Şekil 2.21’den görüleceği üzere kendi y -ekseni yatay eksene göre θ açısı ile duran robotun hareketli eksenini $(90 - \theta)$ kadar saat yönünde dönmüştür. Saat yönünde ki dönüşüme göre şu denklem kullanılır:

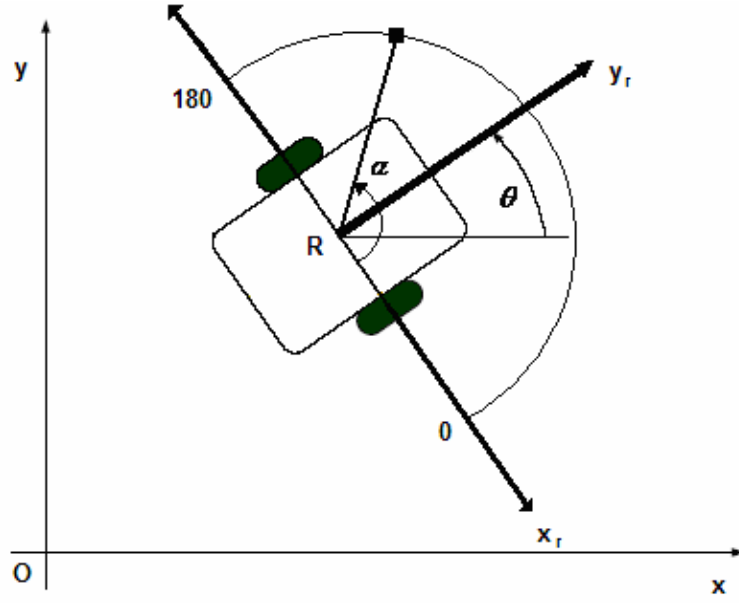
$$\begin{bmatrix} x_o \\ y_o \end{bmatrix} = \begin{bmatrix} x_R \\ y_R \end{bmatrix} + \begin{bmatrix} \cos(90 - \theta) & \sin(90 - \theta) \\ -\sin(90 - \theta) & \cos(90 - \theta) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.48)$$

$\theta > 90$ durumunda saatin ters yönünde $(\theta - 90)$ derece dönmüş olur ve bu dönüşler için şu denklem kullanılır:

$$\begin{bmatrix} x_o \\ y_o \end{bmatrix} = \begin{bmatrix} x_R \\ y_R \end{bmatrix} + \begin{bmatrix} \cos(\theta - 90) & -\sin(\theta - 90) \\ \sin(\theta - 90) & \cos(\theta - 90) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.49)$$

$$x = d \cdot \cos \alpha \quad (2.50)$$

$$y = d \cdot \sin \alpha \quad (2.51)$$

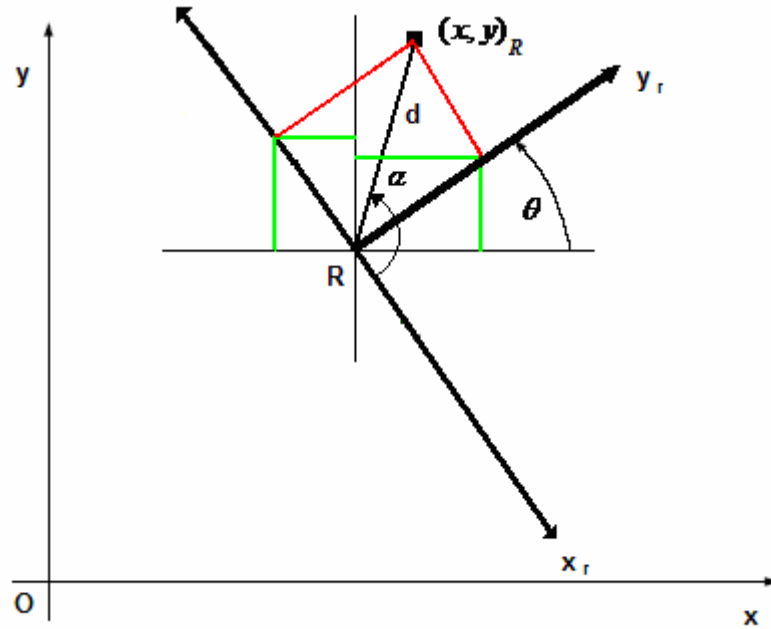


Şekil 2.20 : Algılanan engelin robotun kendi hareket eksenine göre konumu

Şekilde görülen örnekte global eksene göre engelin konumu hesaplanırsa:

$$x_o = x_R + [-d \cdot \cos(180 - \alpha) \cdot \cos(90 - \theta) + d \cdot \sin(180 - \alpha) \cdot \sin(90 - \theta)] \quad (2.52)$$

$$y_o = y_R + [d \cdot \cos(180 - \alpha) \cdot \sin(90 - \theta) + d \cdot \sin(180 - \alpha) \cdot \cos(90 - \theta)] \quad (2.53)$$



Şekil 2.21 : Bir engelin lokal ve global eksen takımında gösterilmesi

Sonuç olarak yukarıdaki eşitlikler düzenlenirse aşağıda görülen denklemler elde edilir ve oluşturulan matematiksel yapının doğruluğu ortaya konur.

$$x_o = x_R + [d \cdot \cos \alpha \cdot \cos(90 - \theta) + d \cdot \sin \alpha \cdot \sin(90 - \theta)] \quad (2.54)$$

$$y_o = y_R + [-d \cdot \cos \alpha \cdot \sin(90 - \theta) + d \cdot \sin \alpha \cdot \cos(90 - \theta)] \quad (2.55)$$

3. DONANIM ve YAZILIM

Otonom olarak hareket eden ve bulunduğu ortamın haritasını çıkaran tekerlekli bir otonom robotu inşa etmek ve uygun yazılımı üzerinden koşturmak için birçok görev parçasının başarı ile tamamlanması gerekmektedir. Ayrıca bu parçaların birbiri ile uyumlu çalışması sistemin kararlılığı için çok önemlidir. Bu bölümde mobil platformun Atilla [4] tarafından yapılan donanımın geliştirilmesi süreci, bu donanım üzerine eklenen parçalar ve kısaca görevleri ile tüm yazılım işlemleri için kullanılan yazılım araçları anlatılmıştır.

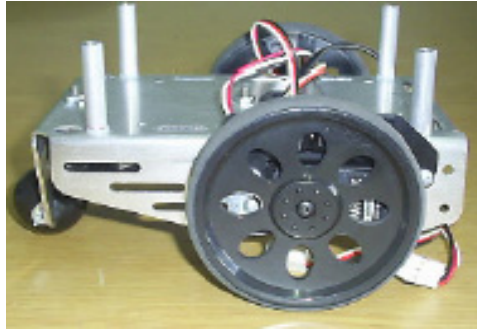
3.1. Donanım

Bir robotun inşası ve programlanması mekanik, elektronik ve bilgisayar bilimlerinin kombinasyonu ile birçok problemin çözümünü kapsar. Basit bir seviyede bir robot:

- Tekerlekli bir platform, kol veya çevresi ile etkileşim içerisinde bulunabilecek yapıları kapsar.
- Üzerindeki veya çevresinde bulunan ve etrafındaki cisim veya ortamları algılayan sensorları içerir.
- Sensor girdilerini işleyen ve anlık konumuna göre bu girdilere cevap veren sistemleri içerir.

Tezde kullanılan robotta öncelikle çevresiyle etkileşim kurabilmesi için 2 adet tekerlek monte edilmiştir. Bu tekerler devamlı bir hareket sağlayan servo motorlarla kontrol edilebilirler. Robotu dengede tutmak için kuyruk tekeri diye adlandırılan küre şeklinde bir teker arka tarafa eklenmiştir. Robot, ileri yönde hareket, geri yönde hareket, sola dönüş, sağa dönüş ve geriye dönüş manevralarını yapabilecek şekilde programlanmıştır. Tekerlere bu manevralarını sağlayan motorlar Parallax firması tarafından üretilen ve sürekli servo (İng. continuous servo) motor olarak adlandırılan eyleyicilerdir. Ayrıca robotun gövdesinin altına sisteme gerekli enerjiyi vermek üzere pil yuvası monte edilmiştir. Robotun tekerleri ve pil yuvasının bir araya getirilmiş hali Şekil 3.1’de görülmektedir.

Robot, ön tarafına monte edilmiş olan Devantech SRF05 ultrasonik mesafe ölçüm sensoruyla etrafındaki cisimleri algılayabilir. Montaj plakaları alüminyum malzemeden yapılmış olup, robotun ön tarafına monte edilmiş olan servo motorun şaftına bağlanmıştır. Montaj plakalarının fonksiyonu ultrasonik mesafe ölçeri taşımaktır. Plaka servo motorun şaftına monte edildiğinden sensor sağa ve sola dönme yeteneğine sahiptir. Atilla [4] tarafından yapılan çalışmada açısız konum bilgisini almak için plaka üzerine ayrıca bir elektronik pusula eklenmiştir. Ancak elektronik pusuladaki hassasiyet ve gürültüden etkilenme problemleri nedeniyle ve açısız konumun da kamera ile tespit edilecek olması sebebiyle bu tezde elektronik pusula kullanılmamıştır.

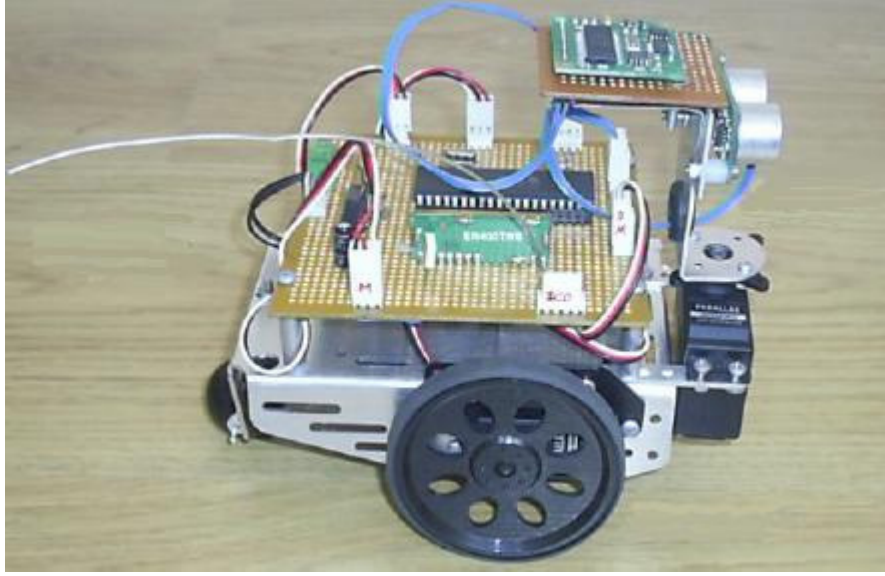


Şekil 3.1 : Robot gövdesi ve tekerlekler [4]

Ultrasonik sensorlar mobil robot uygulamalarında kullanılan en iyi sensorlardandır. Eğer robot engellerle dolu bir odada navigasyon yapıyor ise ultrasonik sensorlar kullanılarak oldukça başarılı sonuçlar alınabilir. Ayrıca robotun bulunduğu ortamın anlamlı bir haritasını çıkarması için sensorların farklı yönleri algılayabilmesi gerekir. Bunun için özellikle tarama yapan veya bu şekilde bir yapıya entegre edilmiş sensorların kullanılması gerekir. Bu tezde kullanılan robotta kullanılan sonar sensor servo motor şaftına bağlandığı için 180 derecelik bir tarama açısına sahiptir.

Günümüzde robotların en büyük üstünlüklerinden biri mikro kontrolörler (MCU) ile kontrol edilmeleridir. Mikro kontrolörler PC'lerdeki mikroişlemciler (CPU) gibidirler. Ancak mikro kontrolörler daha yavaştır ve CPU'lara göre daha az bellek adresi vardır. Buna rağmen gerçek zamanlı problemler için tasarlanmış, CPU'lara göre daha ucuz ve kullanımı daha kolaydır. MCU'lar çoğunlukla harici parçalara ihtiyaç duymadan çalışabilirler.

Tezde kullanılan robotun (Şekil 3.2) faaliyetlerini yönetmek üzere Microchip firmasının ürettiği yüksek performanslı kontrolörü olan PIC18F452 kullanılmıştır. Ayrıca masaüstü bilgisayar ile robot arasında haberleşmeyi sağlamak için sisteme bir RF haberleşme modülü entegre edilmesi gerekmektedir. Bu amaçla Devantech firmasının ürettiği RF04 USB telemetry modülü sisteme entegre edilmiştir. Bu modül bilgisayarın USB portundan enerji almakta ve harici güç kaynağına ihtiyaç duymamaktadır.



Şekil 3.2 : Robotun Atilla tarafından inşa edilen ve kullanılan son hali [4]

Robotun yer değiştirme miktarını ve dönme açılarını hesaplamak üzere düşük ücretli ve piyasada rahatça bulunabilecek A4 Tech web cam kullanılmıştır. Bu ürün bir çok bilgisayara USB portundan bağlanabilir. Ayrıca hafiftir ve düşük güç tüketimi vardır. Bu tezde yapılan çalışmalar Şekil 3.3’de görülen platform üzerinde yapılmıştır. Araç üzerinde ayrıca tekerlere bağlı kodlayıcı ve 9VDC güç veren bir pil bulunmaktadır. Tezde yapılan çalışmalar bu platformdan alınan verilerle 2-boyutlu uzayda yapılmıştır.

3.2. Yazılım

Tez çalışmasında görüntü işleme algoritmalarının geliştirilmesi, kullanıcı ara yüzünün tasarlanması, seri port üzerinden gelen verilerin okunması gibi genel yazılım süreçleri Microsoft firması tarafından geliştirilen .NET 2005 platformu içerisinde Visual C++.NET yazılımıyla gerçekleştirilmiştir. Kameradan gerçek zamanlı olarak görüntünün alınması, alınan görüntüler üzerindeki işlemlerin

yapılabilmesi için Intel firması tarafından oluşturulan ve açık kütüphane şeklinde ücretsiz olarak kullanılan bilgisayarlı görü kütüphanesi OpenCV (İng. Intel Open Source Computer Vision Library) [30] kullanılmıştır. OpenCV bir çok yazılım platformu tarafından desteklenen ve çalışılan platforma kolayca entegre edilebilen içerisinde 300 adet C fonksiyonu ve bir çok C++ sınıfı içeren açık bir kütüphanedir.



Şekil 3.3 : Tezde kullanılan robot ve kamera platformu

Bu çalışmada ayrıca yazılımların geliştirildiği bilgisayara RF Modülden gelen dataların RS-232 portu üzerinden okunabilmesi için sanal com-port sürücüsü (İng. virtual com-port driver), kameradan görüntülerin alınabilmesi için üretici firmanın sağladığı web cam sürücüsü yüklenmiştir. Ayrıca robot üzerinde işlemleri yöneten Microchip PIC18F452 mikro kontrolörünün programlanması için CCS PIC Compiler derleyicisi ve bu derleyicinin ürettiği verileri mikro kontrolöre yüklemek için MPLAB programı bilgisayara yüklenmiştir. Bir önceki bölümde anlatılan görsel

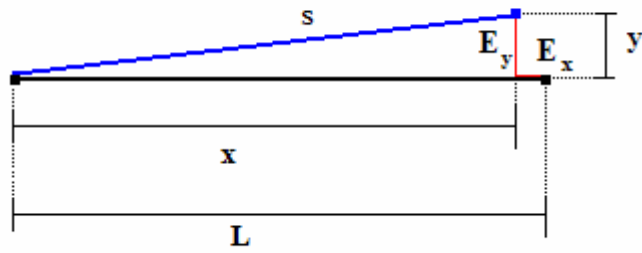
odometri alıřmaları iin oluřturulan denklemler ve algoritmalar yazılım ortamına aktarılmıř ve test edilmiřtir. Yapılan testler ve elde edilen sonular bir sonraki blmde ayrıntılı olarak incelenmiřtir.

4. DENEYSEL SONUÇLAR ve GENEL DEĞERLENDİRME

Bu bölümde önceki bölümlerde anlatılan ve robot üzerinde uygulanan algoritma ve yaklaşımların deneysel yöntemleri ve bu deneylerden elde edilen sonuçlar yorumlanmıştır.

4.1. Doğrusal Yol Testi

Yazılım ve donanımı tamamlanan robot platformda ilk olarak robotun doğrusal yolda aldığı belli bir mesafe boyunca ölçtüğü değerler ve bu değerlerin hata oranları tespit edilmiştir. Bunun için robot kapalı bir oda içerisinde halı üzerinde L metre uzunluğunda bir yol boyunca hareket ettirilmiş ve Şekil 4.1’de görüldüğü gibi robotun ölçtüğü x ve y mesafeleri, açısal pozisyonu bulunmuştur. Dikkat edileceği üzere robotun y yönünde mesafesi E_y ölçüm hatasına eşittir. E_x robotun x eksenindeki ölçüm hatasını ve s ise robotun toplam aldığı yolun ölçüm sonucunu vermektedir.



Şekil 4.1 : x- ekseninde L mesafesi kadar gitmesi istenen robotun algıladığı mesafeler ve ölçüm hataları

Tablo 4.1’de yukarıda anlatılan amaca yönelik oluşturulan 1 metre uzunluğundaki test ortamında robotun x ve y yönünde alınan ölçüm sonuçlarından örnekler gözükmemektedir. Robot ideal şartlarda x ekseninde 100 cm yol alması gerekirken ortalama 96.65 cm yol almıştır. Aynı şekilde robot y ekseninde 0 cm yol

alması gerekirken ortalama 4.66 cm yol almıştır. Bu ölçüm sonuçlarından anlaşılacağı üzere robotun 0 derece olması gereken açısal konumu ortalama 2.76 derece şeklinde ölçülmüştür.

Tablo 4.1 : 1m uzunluğunda doğrusal bir hat boyunca robotun ölçtüğü mesafe örnekleri

Görsel Odometri ile ölçülen x mesafesi (cm)	Gerçekte istenen x mesafesi (cm)	Görsel Odometri ile ölçülen y mesafesi (cm)	Gerçekte istenen y mesafesi (cm)	Görsel Odometri ile ölçülen x-y mesafelerine göre robot açısı (derece)	Gerçekte istenen robot açısı (derece)
94.75	100	4.41	0	2.66	0
95.83	100	5.45	0	3.26	0
94.81	100	4.31	0	2.60	0
93.67	100	4.30	0	2.63	0
104.2	100	4.82	0	2.65	0
Görsel Odometri ile ölçülen ortalama x mesafesi : 96.65	Gerçekte istenen ortalama x mesafesi: 100	Görsel Odometri ile ölçülen ortalama y mesafesi: 4.66	Gerçekte istenen ortalama y mesafesi: 0	Görsel Odometri ile ölçülen x-y mesafelerine göre ortalama robot açısı: 2.76	Gerçekte istenen ortalama robot açısı: 0

Şekil 4.1’de yapısı ve Tablo 4.1’de ölçüm sonuçları gösterilen robotun hareketiyle ilgili olarak istenen değerler göz önüne alındığında hatalar aşağıda gösterildiği şekliyle bulunur.

Robotun x eksenindeki ölçtüğü yolun alması istenen toplam yola göre yüzde olarak ölçüm hatası:

$$E_x = \frac{|x_{istenen} - \bar{x}_{ölçülen}|}{L} \times 100 = \frac{|100 - 96.65|}{100} \times 100 = \%3.35 \quad (4.1)$$

Robotun y eksenindeki ölçtüğü alması istenen toplam yola göre yüzde olarak ölçüm hatası:

$$E_y = \frac{|y_{istenen} - \bar{y}_{ölçülen}|}{L} \times 100 = \frac{|0 - 4.66|}{100} \times 100 = \%4.66 \quad (4.2)$$

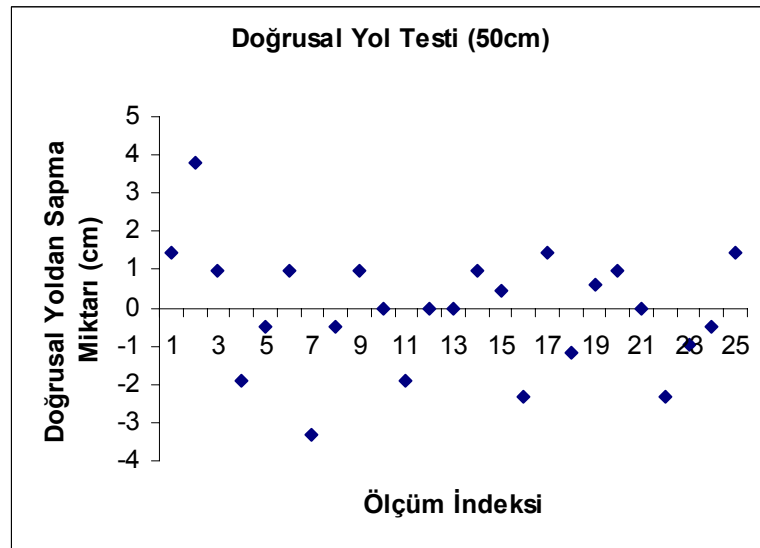
Robotun ölçtüğü toplam yolun alması istenen toplam yola göre yüzde olarak ölçüm hatası:

$$E = \frac{L_{istenen} - L_{ölçülen}}{L_{istenen}} \times 100 \quad (4.3)$$

$$s = \sqrt{x^2 + y^2} = L_{ölçülen} \quad (4.4)$$

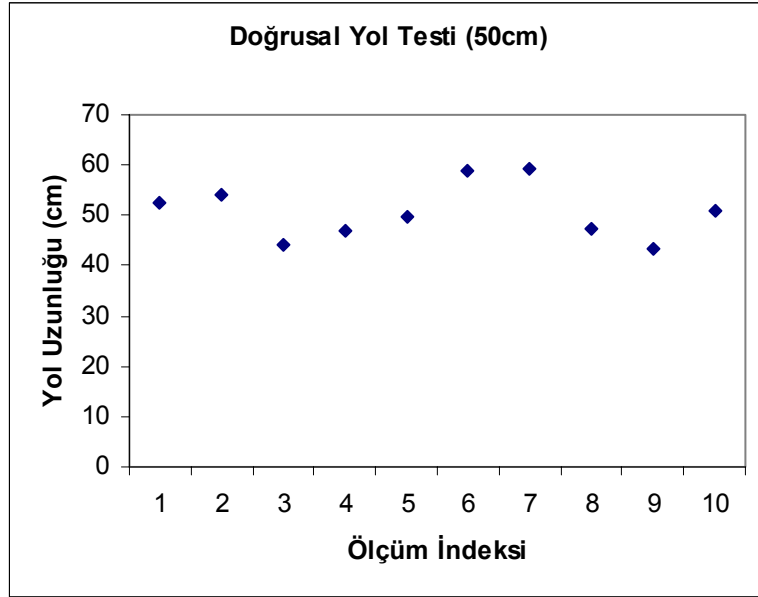
$$E = \frac{100 - \sqrt{96.65^2 + 4.66^2}}{100} \times 100 = \%3.24 \quad (4.5)$$

Yukarıda gösterildiği üzere robotun her 1m hareket ettiğinde robot 3.24 cm ve 2.76 derece pozisyon hatası vermektedir. Yukarıdaki tabloda belirtilen ölçüm sonuçlarına robot üzerinde bulunan servo motorların yüzde yüz kalibre edilmemiş olması ve aralarındaki hız farkından kaynaklanan belli yönlerde sapma gibi etkilerde dahil olduğu için görsel odometri algoritmasının doğrusal bir yoldaki performansı daha hassas ölçebilmek için el ile robot doğrusal bir hat üzerinde hareket ettirilerek bir dizi ölçümler yapılmıştır. Bu amaçla 50 cm uzunluğunda bir hat üzerinde robot el ile itilerek görsel odometri algoritmasının ölçüm sonuçları kaydedilmiştir. Robot 25 kez ilerletilerek yoldan sapma miktarı Şekil 4.2'deki grafiğe yansıtılmıştır. Grafikten de anlaşılacağı üzere robot düzgün bir yolda ilerletilse bile alınan ölçümlerde sapmalar görülmektedir. Bu testlerde elde edilen değerlere göre doğrusal yoldan sapmaların standart sapması 1.56 cm'dir ve bu da algoritmanın performansının kabul edilebilirliğini göstermektedir.



Şekil 4.2 : Robotun el ile 50cm uzunluğunda doğrusal bir hat üzerinde ilerletilmesiyle ölçülen doğrusal yoldan sapma miktarları

Yukarıda bahsedilen amaca yönelik olarak görsel odometri algoritmasının doğrusal yola ilişkin verdiği ölçümler Şekil 4.3'te görülmektedir. Robot 10 kez 50 cm uzunluğunda yol üzerinde el ile hareket ettirilmiş ve yol uzunluğu ortalama 50.65 cm olarak ölçülmüştür. Hata ortalama %1.3 olarak ölçülmüştür.

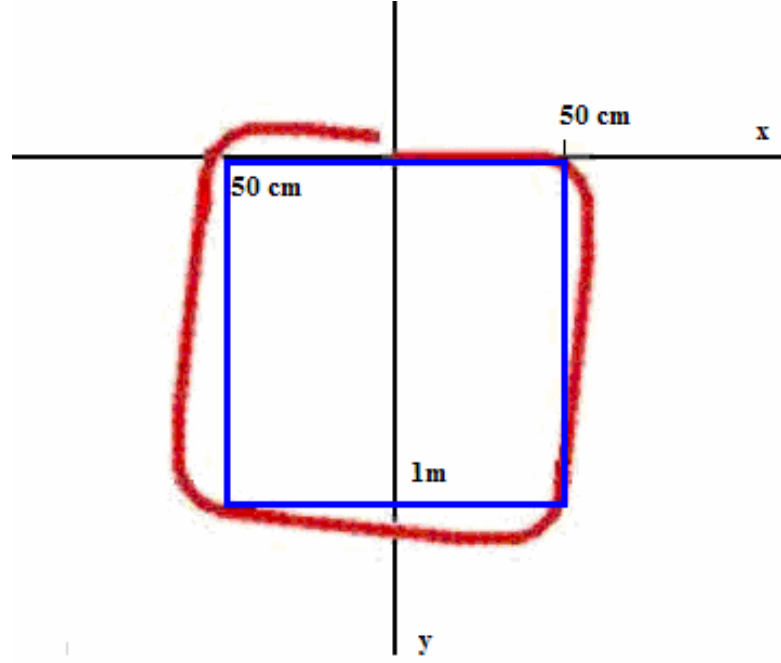


Şekil 4.3 : Robotun el ile 50cm uzunluğunda doğrusal bir hat üzerinde ilerletilmesiyle ölçülen doğrusal mesafeler

4.2. Lokalizasyon Testi

Bir önceki testte robotun doğrusal yolda ne kadar bir sapma yaptığı denenmişti. Bir sonraki testte ise robotun hem doğrusal hareketinden hem de dönüş hareketlerinden elde edilen lokalizasyon bilgilerinin doğruluğunu saptamak üzere uygun test ortamı yaratılmıştır. Bu amaçla kapalı bir alan yaratılmıştır. Robot bu kapalı alan içerisinde 1m kenar uzunluğuna sahip kare şeklinde bir yol izlemiş ve halı üzerindeki optik akış vektörleriyle elde edilen lokalizasyon bilgileri bilgisayar üzerine aktarılmıştır. Oluşturulan kapalı alan içerisinde robotun hareketinden elde edilen lokalizasyon bilgisi Şekil 4.4'te gösterildiği gibi elde edilmiştir. Şekilden de anlaşılabileceği üzere robot genel itibariyle izlediği yolun şeklini çıkarabilmeyi başarmıştır. Ancak dikkat edileceği üzere dönüş hareketlerinde oluşan hatalardan dolayı gerçek yoldan sapmalar gözlenmektedir. Lokalizasyondaki hataların genel kaynağı robotun dönmesi sırasında oluşan optik akış vektörlerinin yazılım üzerinde örnekleme zamanından kaynaklanmaktadır. Yapılan testler sonucu robotun sağa veya sola dönüş

hareketlerinden ortalama 5-10 derece arasında pozisyon hatası olduğu tespit edilmiştir. Robotun kendi dönüş hareketinden kaynaklanan pozisyon hatası da göz önüne alındığında kapalı bir eğri üzerindeki lokalizasyon bilgisi geçerli sayılmış ve robotun bulunduğu ortamını çıkarması için deneyler yapılmıştır.



Şekil 4.4 : 1m x 1m ölçülerinde kare şeklinde bir yol üzerinde robotun oluşturduğu harita



Şekil 4.5 : Robotun hareketini ve ortaya koyduğu haritayı test etmek için oluşturulan koridor

4.3. Haritalama Testi

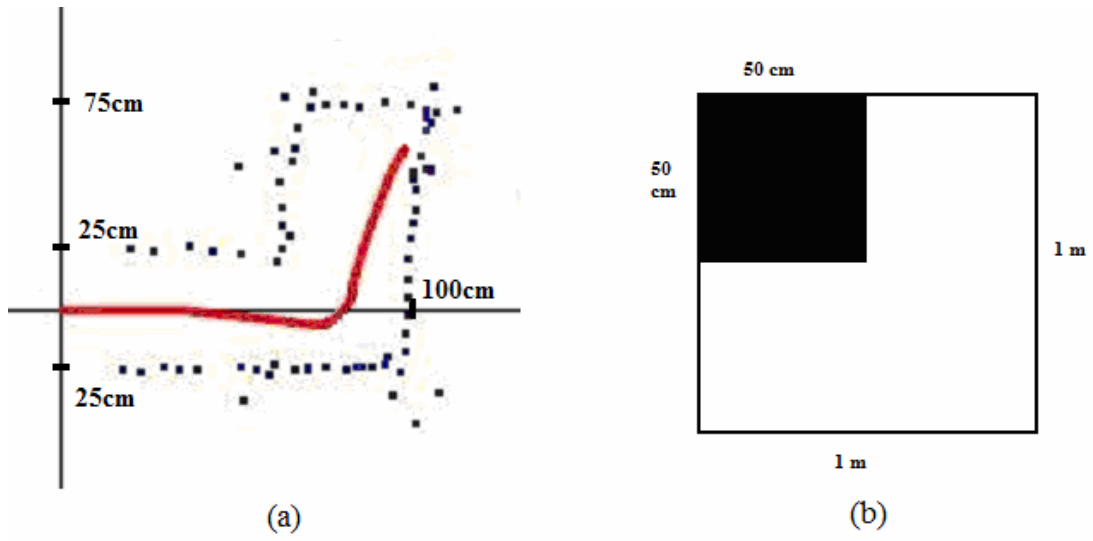
Robotun boyutları önceden belirlenen bir yol üzerindeki lokalizasyon bilgisi alındıktan sonra üzerinde bulunan ultrasonik sensor ile bilmediği bir ortamın haritasını çıkarması için Şekil 4.5'te görüleceği üzere halı üzerinde 1m kenar genişliğine sahip kare şeklinde bir kapalı alan yaratılmıştır. Bu kare şeklindeki alanın dörtte birlik bir alanı kapatılarak robotun bir koridor boyunca engellere çarpmadan ilerlemesi ve eş zamanlı olarak bu koridorun haritasını çıkarması istenmiştir.

Yapılan testler sonucu robot bu koridor boyunca engellere çarpmadan ilerlemiş ve Şekil 4.6'te görülen haritayı çıkarmıştır. Harita üzerinde (Şekil 4.6a) görülen kırmızı çizgi robotun lokalizasyon bilgisini, mavi kareler ise robot üzerinde 180 derece dönebilen sonar sensorun aldığı mesafe örneklerini göstermektedir. Aynı şekil üzerinde sağ tarafta bulunan görüntü (Şekil 4.6b) oluşturulan alanın idealdeki halini göstermektedir.

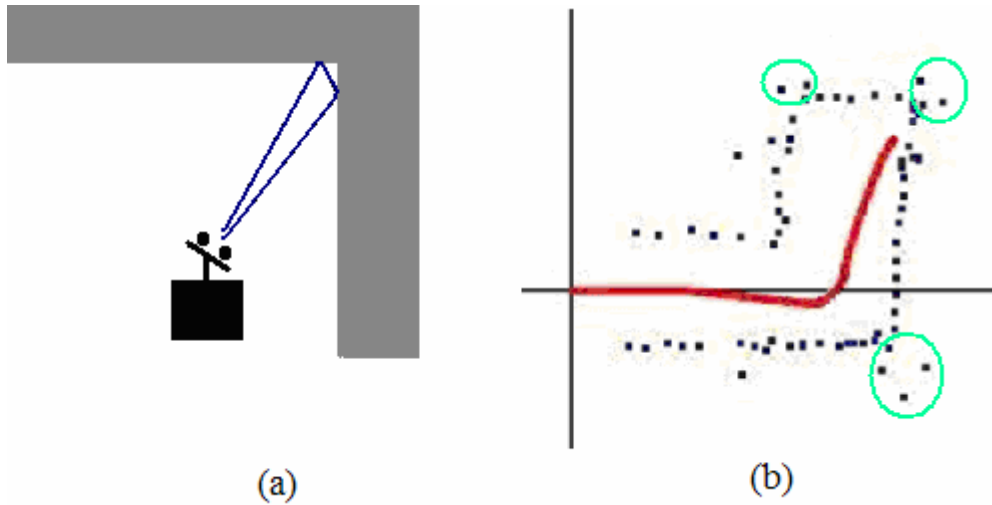
Robotun bulunduğu ortamda kendi pozisyonunu belirleyebildiği ve ortamın haritasını çıkarabildiği yapılan deneylerle gösterilmiştir. Ancak engellerin algılanması ve harita üzerine aktarılmasıyla ilgili olarak dikkat çeken bir nokta engellerin kesişim veya köşe noktalarında oluşan hatalardır. Bu tür hatalar engeller üzerindeki yansılardan kaynaklanır. Şekil 4.6'da gösterilen harita üzerinde oluşan hatalar Şekil 4.7'de sağ tarafta (Şekil 4.7b) yeşil daireler içerisinde gösterilmiş olmakla beraber engellerin mesafesinin nasıl yanlış algılandığı sol tarafta (Şekil 4.7a) sembolik olarak gösterilmiştir. Bu tür hatalar minimize edildiği takdirde daha kesin sonuçların elde edileceği aşikardır.

Yapılan deneyler içerisinde son olarak kapalı bir alan içerisinde rastgele konumlandırılmış cisimler arasında robotun hareketini ve ortaya koyduğu haritanın doğruluğunu sınamak adına 1m x 1m ölçülerinde bir test ortamı yaratılmıştır. Kapalı alanın ortasına ve bir köşesine kutular yerleştirilerek robotun bir köşeden başlayarak yaptığı hareket ve ortaya koyduğu harita ölçülmüştür. Ancak şunu belirtmekte yarar var ki robotun çalışması süresince bilgisayar fazla bellek kullanımı hatası verdiğinden bu alan içerisindeki testler parça-parça yapılmıştır. Dolayısıyla açısız hatalar belli bir noktada tekrar sıfırlandığından Şekil 4.8'de görülen ve ideale yakın olan harita elde edilmiştir. Eğer robot kesintisiz çalıştırılrsa ve bu şekilde bir harita

elde edilmiş olsaydı daha önce deneylerde olduğu gibi açısal hataların birikmesinden dolayı haritada belli bir yöne doğru kayma olacaktı.



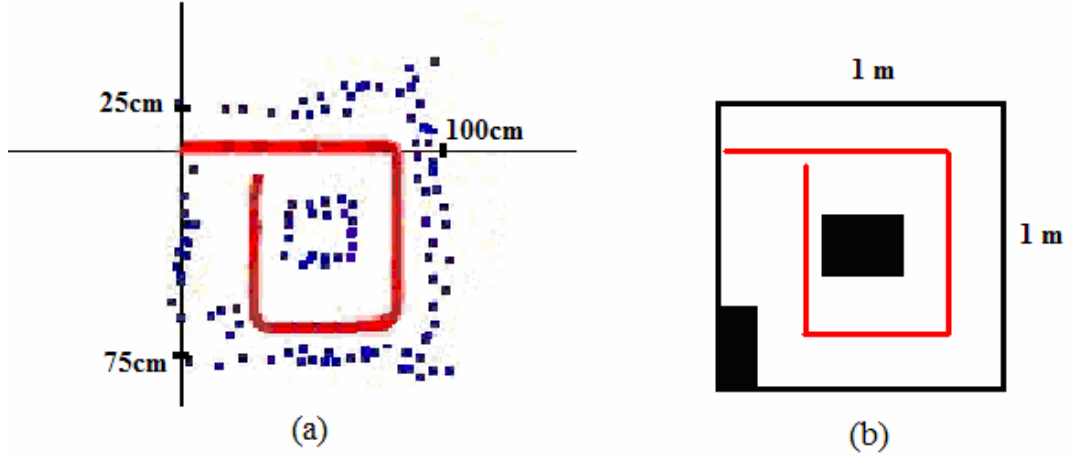
Şekil 4.6 : Robotun 1 m uzunluğunda ve L şeklinde oluşturulan koridor boyunca yaptığı harekete ilişkin: a) ortam haritası, b) ideal şekil



Şekil 4.7 : a) Ultrasonik sensordan çıkan dalgaların engellerin kesişim bölgelerinde yansıması ve b) koridor boyunca ilerleyen robotun köşe noktalarındaki yansılardan dolayı algıladığı hatalı noktalar

Şekil 4.8a'da görüldüğü üzere robotun hareketi ve oluşturduğu harita Şekil 4.8b'de bulunan ideal haritaya oldukça yakın bir sonuç vermiştir. Buradan anlaşılacağı üzere, bu tez kapsamında oluşturulan algoritma özellikle lineer hareketin algılanması ve engellerin konumlandırılması açısından oldukça iyi sonuçlar vermektedir. Üzerinde daha çok çalışılması gereken veya farklı bir bakış açısı isteyen konu robotun açısal

konumunu belirlemedir [31]. Gürültü ve sallantı robotun açısal konumunu etkilemektedir. Bundan dolayı dış etkilere daha dayanıklı ve daha kesin sonuçlar veren algoritma konusunda çalışmalar devam edecektir.



Şekil 4.8 : 1m x 1m alan içerisinde konumlandırılmış cisimler arasında: a) robotun izlediği yol ve oluşturduğu harita, b) idealde robotun izlemesi gereken yol ve ortamın haritası

4.4. Genel Değerlendirme

Bu çalışmada tekerlekli mobil robotlar için bir lokalizasyon ve haritalama algoritması geliştirilmesi amaç edinilmiş ve bu amaca yönelik çalışmalar yapılmıştır. Yapılan çalışmalar ve geliştirilen algoritmalar daha önceden inşa edilmiş, üzerinde bulunan algılayıcılar ve eyleyiciler sayesinde bulunduğu ortam içerisinde otonom olarak hareket edebilen küçük boyutlu tekerlekli mobil bir robot üzerinde test edilmiştir. Robot, üzerinde bulunan bir adet sonar sensor ve ucuz bir kamerayla içinde bulunduğu ortamın haritasını çıkarabilme kabiliyetine erişmiştir.

Lokalizasyon robotun anlık konumunu doğru bir şekilde bilmesi ve bulunduğu ortamın haritasını çıkarabilmesi açısından önemlidir. Bu çalışmada robotun anlık pozisyonunu bilmesi için görsel bir odometri algoritması geliştirilmiştir. Bunun için robot üzerinde bulunan kamera vasıtasıyla robotun bulunduğu yüzey üzerinden alınan görüntüler bilgisayar ortamında işlenmiştir. Kameradan alınan görüntüler üzerinde öncelikle görüntüye ilişkin karakteristik bilgiler içeren özniteliklerin belirlenmesi amaç edinilmiştir. Bu amaçla görüntü işlemede yaygın bir şekilde kullanılan köşe belirleme yöntemiyle görüntüler üzerinde temel noktalar belirlenmiştir. Daha sonra belirlenen köşe noktalarının optik akış yöntemiyle diğer

bir görüntü üzerindeki yer deęiřtirmeleri hesaplanmıřtır. Bu iřlem her bir görüntü çiftine uygulanmıřtır. Elde edilen optik akıř vektörleri ve bazı matematiksel yaklařımlarla robotun gerçekte hangi açıyla ne kadar yer deęiřtirdięi hesaplanmıřtır. Yer deęiřtirmeler ve açısal deęiřimler toplamı robotun pozisyon bilgisini vermiř, ultrasonik sensordan alınan engellere iliřkin mesafe bilgileriyle ortamın haritası çıkarılmıřtır. Yapılan testler sonucunda robotun doęrusal ve açısal yer deęiřtirmelerinde % 3-5 arasında hata ölçölmüřtür. Bu deęerler bu türden mobil robot uygulamalarında kabul edilebilir sınırlar içerisinde olup geliştirilen algoritmanın her boyuttan hareketli robot üzerine uygulanabileceęini göstermiřtir.

Robot platform üzerinde kullanılacak algılayıcı sayısı arttıkça ve iřlemleri yürüten iřlemci kapasitesi arttıkça daha doęru sonuçlar algılanacaęı açıktır. Ancak bu çalıřmada amaç edinildięi üzere bundan sonraki çalıřmalarda robot platformun boyutunu daha da küçölterek ve mümkün olan en az sayıda algılayıcı veya sensor kullanarak haritalama algoritmaları geliřtirmek hedef alınacaktır. Gelecekte çok küçük boyutta robotlar üzerine yine küçük boyutta kamera entegre ederek veya mikro sensorlar kullanarak robotun bulunduęu ortamın haritasını çıkarması üzerine çalıřmalar yapılacaktır.

Sonuç olarak görüntü ve görüntülerin iřlenmesi robotik haritalamada güçlü bir seçenek olarak karřımıza çıkmaktadır ve ilerleyen yıllarda daha da ilgi çekecek bir yöntemdir. Buna verilebilecek en basit kanıt görüntünün eř zamanlı olarak daha çok bilgi içerebilmesidir. Bir kamera ile birden fazla bilgi çıkarılabilmekte ve alınan bu bilgiler farklı yöntemler deęerlendirilebilmektedir. Dijital kameralar ve gömölü iřlemciler daha güçlü ve daha ucuz olduęu sürece robotik haritalamada kullanılanlar bařta olmak üzere robotlar üzerinde birçok sensorun önüne geçecek ve daha kesin haritalar elde edilecektir.

KAYNAKLAR

- [1] Ondokuz Mayıs Üniversitesi Robotik Sayfası, 2007. *Robotik*, http://www2.omu.edu.tr/akademik_birimler/muhendislik/elektr/Lego_sitesi4/Giris.htm
- [2] **Schaerer, S. S.**, 2006. Practical Visual Odometry For Small Embedded Systems, *Master Thesis*, The University of Manitoba, Manitoba.
- [3] **Thrun, S.**, 2002. Robotic Mapping: A Survey, *Technical Report*, **CMU-CS-02-111**, Carnegie Mellon University, Pittsburgh.
- [4] **Atilla, M.**, 2007. Building an Autonomous Wheeled Robot and Mapping, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [5] Wikipedia, 2008. *Corner Detection*, http://en.wikipedia.org/wiki/Corner_detection
- [6] **Moravec, H.**, 1980. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, *Robotics Institute Technical Report*, **CMU-RI-TR-3**, Carnegie-Mellon University, Pittsburgh.
- [7] **Harris, C. and Stephens, M.**, 1988. A combined corner and edge detector, *Proceedings of the 4th Alvey Vision Conference*, Manchester, UK, 1988, 147-151.
- [8] **Smith, S. M. and Brady, J. M.**, 1997. SUSAN - a new approach to low level image processing, *International Journal of Computer Vision*, **23**, 45-78.
- [9] **Shi, J. and Tomasi, C.**, 1994. Good Features to Track, *9th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994, 593-600.
- [10] **Tomasi, C. and Kanade, T.**, 1991. Detection and Tracking of Point Features, *Technical Report*, **CMU-CS-91-132**, Carnegie-Mellon University, Pittsburgh.
- [11] **Trajkovic, M. and Hedley, M.**, 1998. Fast corner detection, *Image and Vision Computing*, **16**, 75-87.

- [12] **Lowe, D.**, 1999. Object recognition from local scale- invariant features, *Proceedings of International Conference on Computer Vision*, 1999, 1150-1157.
- [13] **Se, S., Lowe, D. and Little, J.**, 2002. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, *The International Journal of Robotics Research*, **21**, 735–758.
- [14] **Se, S., Lowe, D. and Little, J.**, 2001. Vision-based mobile robot localization and mapping using scale-invariant features, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)*, Seoul, Korea, May 2001, 2051 – 2058.
- [15] **Dornhege, C. and Kleiner, A.**, 2006. Visual Odometry for Tracked Vehicles, *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, Gaithersburg, Maryland, USA, 2006.
- [16] **Lucas B. D. and Kanade T.**, 1981. An Iterative Image Registration Technique with an Application to Stereo Vision, *Proceedings of Imaging Understanding Workshop*, Washington, USA, 121-130.
- [17] **Campbell, J., Sukthankar, R., and Nourbakhsh, I.**, 2004. Visual odometry using commodity optical flow, *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI 2004)*, Menlo Park, California, July 2004, 1008–1009.
- [18] **Campbell, J., Sukthankar, R. and Nourbakhsh, I.**, 2004. Techniques for evaluating optical flow for visual odometry in extreme terrain, *International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai, Japan, September 2004, **4**, 3704 – 3711.
- [19] **Campbell, J., Sukthankar, R., Nourbakhsh, I. and Pahwa, A.**, 2005. A Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision, *Proceedings of the 2005 IEEE International Conference of Robotics and Automation (ICRA 2005)*, Barcelona, Spain , April 2005, 3421-3427.
- [20] **DiVerdi, S. and Hollerer, T.**, 2007. GroundCam: A Tracking Modality for Mobile Mixed Reality, *IEEE Virtual Reality Conference 2007*, Charlotte, North Carolina, USA, 75 – 82.
- [21] **Cheng, Y., Helmick, D. M., Clouse, D. S., Matthies, L. H. and Roumeliotis, S. I.**, 2004. Path following using visual odometry for a Mars rover in

high-slip environments, *Proceedings of 2004 IEEE Aerospace Conference*, Montana, USA, March 2004, **2**, 772-789.

- [22] **Cheng, Y., Maimone, M. and Matthies, L.**, 2005. Visual odometry on the Mars Exploration Rovers Systems, *IEEE International Conference Man and Cybernetics*, 2005, **1**, 903- 910.
- [23] **Mester, G.**, 2006. Motion Control of Wheeled Mobile Robots, *4th Serbian-Hungarian Joint Symposium on Intelligent Systems (SISY 2006)*, Subotica, Serbia and Montenegro.
- [24] Wikipedia, 2008. *Odometry*, <http://en.wikipedia.org/wiki/Odometry>
- [25] Wikipedia, 2008. *Optical Flow*, http://en.wikipedia.org/wiki/Optical_flow
- [26] Boğaziçi Üniversitesi Robot Sitesi, 2007. *Optik Akış*, http://robot.cmpe.boun.edu.tr/593/gorusimge/3_9_Optik_Ak_i_c_s.html
- [27] **Bouget, J. Y.**, 2000. Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm, Microprocessor Research Labs, Intel Corporation.
- [28] **Thrun, S.**, 2006. *Optical Flow*, Lecture Notes, Computer Science 223b, Stanford University.
- [29] **Stavens, D.**, 2007. *Lecture Notes on Optical Flow and OpenCV*, Guest Lecture, Computer Science 223b, Stanford University.
- [30] OPENCV, 2004. *Open Source Computer Vision Library*, <http://www.intel.com/technology/computing/opencv/index.htm>
- [31] **Krose, B. J. A., Dev, A. and Groen, F. C. A.**, 2000. Heading direction of a mobile robot from the optical flow, *Image and Vision Computing*, **18**, 415-424.

ÖZGEÇMİŞ

Berkan ÇAKIR, 1983 yılında Sivas'ta dünyaya geldi. İlk ve orta öğrenimini bu şehirde tamamladıktan sonra 2000 yılında İTÜ Elektronik ve Haberleşme Mühendisliği bölümünde lisans eğitime başladı. 2005 yılında Elektronik ve Haberleşme Mühendisi olarak mezun olduktan sonra aynı yıl İTÜ Fen Bilimleri Enstitüsü Mekatronik Mühendisliği bölümünde yüksek lisans eğitime devam etti. 2006-2007 öğretim yılı güz döneminde öğrenci değişim programıyla Danimarka Teknik Üniversitesi'nde bir dönem eğitim aldıktan sonra yurt içinde özel sektörde çalışma hayatını sürdürdü. Şuan özel sektörde Elektronik Mühendisi olarak görev yapmaktadır.