

**FPGA-TABANLI, STERYO GÖRMEYE SAHİP BİR ROBOTTA ÜÇ
BOYUTTA EŞ ZAMANLI KONUMLAMA VE HARİTALAMA**

MUHAMMET FATİH AYDOĞDU

YÜKSEK LİSANS TEZİ

ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2010

Fen Bilimleri Enstitü onayı

Prof. Dr. Ünver KAYNAK

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Prof. Dr. Mehmet Önder EFE

Anabilim Dalı Başkanı

Muhammet Fatih AYDOĞDU tarafından hazırlanan FPGA-TABANLI, STERYO GÖRMEYE SAHİP BİR ROBOTTA ÜÇ BOYUTTA EŞ ZAMANLI KONUMLAMA VE HARİTALAMA adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Coşku KASNAKOĞLU

Tez Danışmanı

Yrd. Doç. Dr. M.Fatih DEMİRCİ

Tez Danışmanı

Tez Jüri Üyeleri

Başkan: Yrd. Doç. Dr. Ali Cafer GÜRBÜZ

Üye : Doç. Dr. Bülent TAVLI

Üye : Doç. Dr. Hamza KURT

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Muhammet Fatih AYDOĞDU

Üniversitesi : TOBB Ekonomi ve Teknoloji Üniversitesi
Enstitüsü : Fen Bilimleri
Anabilim Dalı : Elektrik ve Elektronik Mühendisliği
Tez Danışmanları : Yrd. Doç. Dr. Coşku KASNAKOĞLU
: Yrd. Doç. Dr. M. Fatih DEMİRCİ
Tez Türü ve Tarihi : Yüksek Lisans – Temmuz 2010

Muhammet Fatih AYDOĞDU

**FPGA-TABANLI, STERYO GÖRMEYE SAHİP BİR ROBOTTA ÜÇ
BOYUTTA EŞ ZAMANLI KONUMLAMA VE HARİTALAMA**

ÖZET

Konum ve harita bilgisi mobil bir robotun başarımında etkili olan önemli faktörlerdendir. Bu çalışmada, stereo kameraya sahip mobil bir robotun gerçek zamanda ve eş zamanlı olarak konumlama ve haritalama yapmasını sağlayacak genişletilmiş kalman filtresi için FPGA üzerinde paralel bir mimari oluşturulmaya çalışılmıştır. Sistemde, görüntülerde bulunan köşeler özellik noktaları olarak kabul edilmiştir. Stereo kameralardan alınan görüntülerdeki köşeler eşleştirme ve 3 boyutta konum tespiti aşamalarından geçirilerek köşelerin 3 boyutta robota göre konumu hesaplanmaktadır. Köşelerin 3 boyuttaki konumları genişletilmiş kalman filtresine ölçüm güncellemesinde kullanılmak üzere aktarılmaktadır. Robotta bulunan eyleyicilerdeki konum değişimi ise filtrenin tahmin aşamasında kullanılmaktadır.

Çalışma Xilinx'e ait Virtex-5 XC5VLX50 çekirdeği üzerinde gerçekleştirilmiştir. Görüntüleri algılamak için Omni Vision'ın OV7720 dijital kameralar kullanılmıştır. Görüntüleri depolamak için Micron' a ait MT4HTF3264HY-53E DDR2 SODIMM bellek kullanılmıştır.

Anahtar Kelimeler: Eş zamanlı konumlama ve haritalama, Genişletilmiş kalman filtresi, Stereo görme, Plessey köşe tespiti algoritması

University : TOBB Economics and Technology University
Institute : Institute of Natural and Applied Sciences
Science Programme : Electrical and Electronics Engineering
Supervisor : Asst. Prof. Dr. Coşku KASNAKOĞLU
Asst. Prof. Dr. M. Fatih DEMİRCİ
Degree Awarded and Date : M.Sc. – July 2010

Muhammet Fatih AYDOĞDU

**SIMULTANEOUS LOCALIZATION AND MAPPING ON AN FPGA BASED
ROBOT WITH STEREO VISION**

ABSTRACT

Location and map information is an important factor affecting the success of a mobile robot. In this study, it is aimed to construct an FPGA based architecture for Extended Kalman Filter assuring simultaneous localization and mapping on a mobile robot with stereo cameras. In the system, corners in the images are regarded as feature points. The positions of the corners with respect to the robot are computed by applying stereo matching and location measurement in 3D. The 3D positions of the corners are fed into the Extended Kalman Filter to be used in the measurement update. The location change in the actuators of the robot is used in motion update.

The study is realized on a Virtex-5 XC5VLX50 FPGA core from Xilinx. OmniVision OV7720 digital cameras are used in order to sense the images. MT4HTF3264HY-53E DDR2 SODIMM is used to store the images.

Keywords: Simultaneous localization and mapping, Extended Kalman Filter, Stereo Vision, Plessey Corner Detection Algorithm

TEŐEKKÖR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocalarım Yrd. Doç. Dr. Coőku KASNAKOęLU'na ve Yrd. Doç. Dr. M. Fatih DEMİRCİ'ye teőekkörü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ	ix
ŞEKİLLERİN LİSTESİ	x
KISALTMALAR	xii
SEMBOL LİSTESİ	xiii
1. GİRİŞ	1
1.1. Eş Zamanlı Konumlama ve Haritalama	1
1.2. Önceki Çalışmalar	3
1.3. Çalışmanın Amacı ve Geliştirme Ortamı	3
2. ÖZELLİK NOKTALARININ ÇIKARIMI	6
2.1. Köşe Tespit Algoritmaları	6
2.2. Plessey(Harris) Köşe Tespit Algoritması	7
2.3. Test Aşaması	14
2.4. Algoritmaya Yönelik Boru Hattı Yapısının Oluşturulması	15
2.5. Köşe Tespit Modüllerinin Performans Sonuçları	22
3. STERYO EŞLEME VE 3 BOYUTTA KONUM ÇIKARIMI	23
3.1. Stereo Eşleme Algoritmaları	23
3.2. Kullanılan Eşleme Algoritması	23

3.3. Eşleşme Algoritmasının Benzetim Sonuçları	30
3.4. 3 Boyutta Konum Çıkarımı	32
3.5. 3 Boyutta Konum Çıkarımının Benzetim Sonuçları	37
4. EŞ ZAMANLI KONUMLAMA VE HARİTA ÇIKARIMI	39
4.1. Genişletilmiş Kalman Filtresi	39
4.2. Hareket Güncellemesi	40
4.3. Ölçüm Güncellemesi	42
4.4. Algoritma	44
4.5. Harita Organizasyonu	54
4.6. EZKH Benzetim Sonuçları	55
5. SONUÇLAR	58
6. KAYNAKLAR	59
ÖZGEÇMİŞ	61

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Gradyan çarpımlarının bulunması	12
Çizelge 2.2. KKOÖ'nün hesaplanması	13
Çizelge 2.3. Kullanılan donanım miktarı	22
Çizelge 3.1. Ölçümleme sonuçları	34

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.3. Moravec algoritmasındaki kaymalar(solda) ve (1,1) yönündeki kayma(sağda)	8
Şekil 2.4. Özdeğerlerin büyüklüğüne göre pikselin sınıflandırılması.	11
Şekil 2.5. 5x5lik bir maske için değişkesi 1,2 olan, ayrık, 2 boyutlu Gaussian dağılımının katsayılarının tamsayıya dönüştürülmüş şekli	14
Şekil 2.6. Farklı özdeğerler için tespit edilen köşeler. ($\sigma_M=1.2$, $\sigma_N=1.2$ sol üst), ($\sigma_M=1.2$, $\sigma_N=1.8$ sağ üst), ($\sigma_M=1.8$, $\sigma_N=1.2$ sol alt), ($\sigma_M=1.8$, $\sigma_N=1.8$ sağ alt)	15
Şekil 2.7. Birinci aşama için oluşturulan modülün K_x 'i hesaplayan parçası	17
Şekil 2.8. Birinci aşama için oluşturulan modülün ikinci parçası	18
Şekil 2.9. İkinci aşama için oluşturulan modülün A değerini hesaplayan parçası	20
Şekil 2.10. İkinci aşama için oluşturulan modülün KKOÖ'nü hesaplayan parçası	21
Şekil 3.1. Eş adaylarının bulunabileceği alan (G1 üstte, G2 altta)	25
Şekil 3.2. 242 ve 256cm uzaklıktan alınan görüntülerde tespit edilen köşeler ve eşleştirme sonucu oluşturular fark haritası. (sol üst 242cm sol kamera, sol orta 242cm sağ kamera, sol alt 242cm fark haritası, sağ üst 256cm sol kamera, sağ orta 256cm sağ kamera, sağ alt 256cm fark haritası)	31
Şekil 3.3. Piksellerdeki ölçekleme ve çarpıklık hatalarından ve lens bozunmalarından kaynaklanan bozunmalar	32
Şekil 3.4. Kalibrasyon görüntülerinin 3 boyuttaki konumları (eksenlerin birimi mm)	33
Şekil 3.5. Üçgenleyerek yeniden kurma yöntemi	35
Şekil 3.6. 3 Boyutta robota göre konumları çıkarılan köşe noktaları (242cm soldaki görüntüler, 256cm sağdaki görüntüler)	38

Şekil 4.1. Robotun konumunun tanımlanması	40
Şekil 4.2. Robotun konum deęişiminin kestirilmesi	41
Şekil 4.3. Robotun konum deęişiminin kestirilmesi	41
Şekil 4.4. Robotun konum deęişiminin kestirilmesi	43
Şekil 4.5. Haritada en fazla bir köşenin bulunabileceęi hacim	55
Şekil 4.6. Robot kapıya 256cm(üstte) ve 250cm(altta) uzaklıktayken simülasyon ortamındaki robotun konumu ve haritasındaki köşelerin konumu ve belirsizlikleri	56
Şekil 4.7. Robot kapıya 246cm(üstte) ve 242cm(altta) uzaklıktayken simülasyon ortamındaki robotun konumu ve haritasındaki köşelerin konumu ve belirsizlikleri	57

KISALTMALAR

Kısaltmalar	Açıklama
GEÖU	Genişletilmiş eğrilik ölçekli uzay
GKF	Genişletilmiş Kalman filtresi
EÖU	Eğrilik ölçekli uzay
EZKH	Eş zamanlı konumlama ve haritalama
G1	1. görüntü
G2	2. görüntü
KF	Kalman filtresi
KKOÖ	Köşe ve kenar olma ölçüsü
KOÖ	Köşe olma ölçüsü
SV	Saat vuruşu

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
ζp	çarpıklık parametresi
f_x	odak uzaklığının yatay piksel büyüklüğü cinsinden değeri
f_y	odak uzaklığının dikey piksel büyüklüğü cinsinden değeri
F	farklılık
G	gradyan
Gçt	gradyan çarpımları toplamı
I	görüntü
k	haritadaki köşe
M maskesi	gradyan çarpımlarının toplamını bul fonksiyonundaki Gaussian
r	robotun parametreleri
R	kameralar arasındaki yön farkı
T	kameralar arasındaki mesafe
u	konum değişimi
w	aktif pencere
z	ölçüm
Σ	covaryans matrisi
μ	durum vektörü
Θ	iki köşenin aynı uzaktaki eş adayları arasındaki maksimum fark

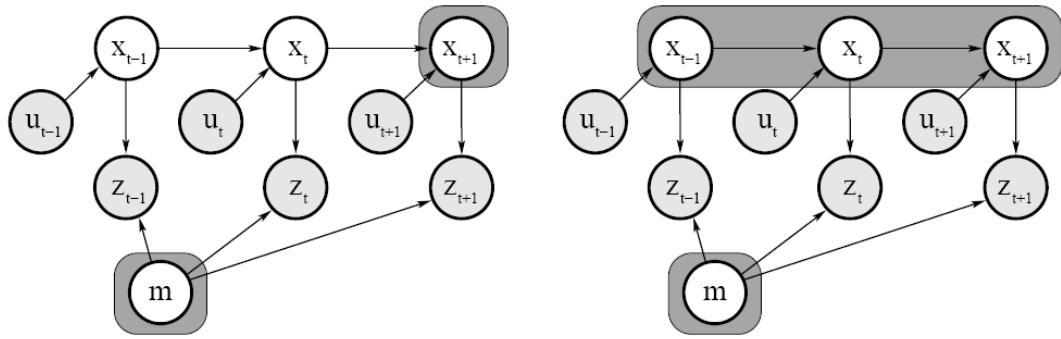
1. GİRİŞ

1.1. Eş Zamanlı Konumlama ve Haritalama

Robotlar, görevlerini gerçekleştirebilmek için buldukları ortam ve bu ortamdaki konumları hakkında bilgi sahibi olmalıdırlar. Hatasız algılayıcılara sahip bir robot ortamdaki konumunu hatasız bir şekilde bildiği zaman algılayıcılarıyla ortam hakkında bilgi toplayıp ortamın haritasını oluşturabilir. Benzer şekilde ortamın hatasız haritasına sahip olan bir robot da ortamdaki konumunu hatasız algılayıcılarıyla tespit edebilir. Yani robotlar ortamın haritasını çıkarmak için konum bilgisine; konum bilgisini çıkarmak içinde ortamın harita bilgisine ihtiyaç duyarlar. Bu bilgilerden birinin tam olarak tespit edilebildiği sistemler çoğu zaman diğer bilgiyi az hatayla tahmin edebilir. Örneğin otomasyon sistemlerindeki montaj robotlarında yüksek çözünürlüklü enkoderler robotun konum bilgisini çok küçük hatalarla sağlarlar. Böylece robot gerektiğinde algılayıcılarıyla ortamın haritasını çıkarabilme yetisine de sahip olmuş olur. Benzer şekilde müze güvenliğinden sorumlu mobil bir robota müzenin doğru harita bilgisi aktarılırsa robot algılayıcılarıyla müzedeki konumunu tespit edebilir. Ancak mobil robotlar için çoğu zaman hatasız konum ve harita bilgisine erişmek imkânlı olmaz. Mobil bir robot, ortama sabitlenmediği için konumunu tespit ederken ve içinde bulunduğu ortamın haritasını çıkarırken algılayıcılarından gelen bilgiye ve eyleycilerine verdiği hareket komutlarının hatasız olarak hareket eylemlerine dönüştüğüne güvenmek zorundadır. Ancak hem algılayıcıların sağladıkları bilgilerin hatalı olma hem de eyleycilere aktarılan komutların yanlış eylemlere dönüşme ihtimalleri vardır. Bu belirsizlikler mobil robotlarda konum ve harita çıkarımını içi içe geçmiş ve eş zamanlı olarak çözülmesi gereken olasılıksal bir probleme dönüştürür. Eş Zamanlı Haritalama ve Konumlama¹ (EZKH) olarak adlandırılan bu problem herhangi bir mobil robotun çözmesi gereken en temel problemdir. EZKH tam EZKH ve çevrimiçi EZKH olarak

¹ Türkçe literatürde bu problemin “Eş Zamanlı Konum Belirleme ve Haritalama” ve “Eş Zamanlı Konum Belirleme ve Harita Çıkarımı” olarak adlandırılmış örneklerine rastladım. Bu çalışmada ise problemi tam anlamıyla kısa bir şekilde anlattığı ve problemin İngilizce karşılığı olan “Simultaneous Localization and Mapping” ile daha fazla bağdaştığı için problemi Eş Zamanlı Konumlama ve Haritalama(EZKH) olarak adlandırmayı uygun gördüm.

ikiye ayrılabilir. Tam EZKH’da belirli bir süre hareket eden robotun hareketi boyunca bulunduğu konumlar ile harita tespit edilirken çevrimiçi EZKH’da ise robotun son konumu ve harita tespit edilir. Alt indis zamanı, m robotun içinde bulunduğu haritayı, u robotun hareketini, x robotun konumunu ve z robotun yaptığı gözlemi göstermek üzere, Şekil-1.1.’de tam EZKH ve çevrimiçi EZKH grafiksel olarak gösterilmektedir. Şekillere göre robot her t anındaki konumu x_t ’ye u_t hareketiyle gelip burada z_t gözlemini yapmakta ve yaptığı bu gözlem ile sahip olduğu harita bilgisini karşılaştırmaktadır. Robot solda gösterilen çevrimiçi EZKH’da koyu renkle belirtilen son konumu x_{t+1} ile harita bilgisi m ’yi bulurken sağda gösterilen tam EZKH’da ise robot harita bilgisi m ile geçtiği tüm x_{t-1} , x_t ve x_{t+1} konumlarını tespit etmektedir.



Şekil 1.1. Çevrimiçi EZKH(solda) ve tam EZKH(sağda) [1]

Çevrimiçi EZKH ile gerçek zamanlı olarak çözülebilen bir problem iken tam EZKH çoğu zaman yalnızca çevrimdışı çözülebilen bir problemdir. Tam EZKH problemine grafik EZKH algoritmaları ile çözüm aranabilirken çevrimiçi EZKH problemi için ise Genişletilmiş Kalman Filtresi ve Kokusuz Kalman Filtresi gibi çözümler mevcuttur. Ayrık Genişletilmiş Bilgi Filtresi ile çevrimiçi çalışan ve Grafik EZKH algoritmasının avantajlarından yararlanan sistemler geliştirilebilir.

1.2. Önceki Çalışmalar

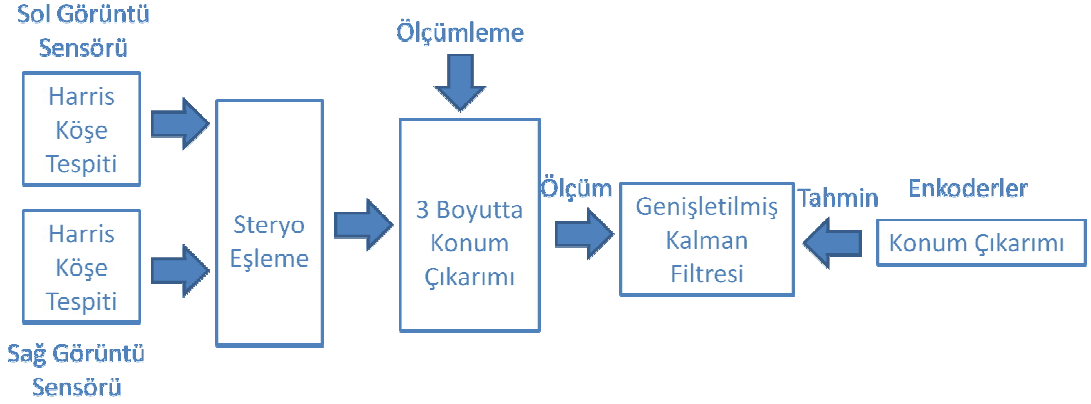
Daha önce belirtildiği gibi EZKH çoğu mobil robot uygulamasında çözülmesi gereken bir sorun olduğu için bu probleme dış ortamda, iç ortamda, havada ve sualtındaki uygulamalarda çözüm aranmıştır. Se, Barfoot, Jasiobedzki tarafından uzay keşiflerinde kullanılmak üzere oluşturulan sistem [2] dış ortam uygulamalarına örnek verilebilir. Sonar sinyallerini kullanarak EZKH gerçekleştiren sistem [3] ise su altı uygulamalarına örnektir.

Literatürde gerçek zamanda çalışmaya uygunluğu ve daha az bellek gereksinimi olması sebebiyle çevrimiçi EZKH ile ilgili çalışma sayısı daha fazladır. Çevrimiçi EZKH konusunda ise teorik yapılarını karmaşıklığının aksine kolay uygulanabilirlikleriyle Kalman Filtresi tabanlı algoritmalar ön plana çıkmaktadır. Olasılıksal yaklaşımlar gerektiren çoğu problemin çözümünde Kalman filtresi tabanlı algoritmalar kullanılmaktadır. Genişletilmiş Kalman Filtresi(GKF) ise sadece lineer durumlarda çalışan Kalman Filtresinin(KF)'nin aksine lineer olmayan durumlarda da çalışabildiğinden çözebildiği problem kümesi daha büyüktür. Olasılıksal tabanlı bu algoritmalar sabit mimariye sahip işlemciler üzerinde koşturuldukları gibi uygulamaya yönelik geliştirilmiş olan devreler üzerinde ve FPGA'ler üzerinde de uygulanabilmektedir [4,5]. Ayrıca sabit mimarinin hızlı programlanabilirliği avantajını hem FPGA'lerin paralelleştirilebilen yapısının avantajlarını kullanan sistemler vardır [2].

Görüntü işleme uygulamalarında işlenmesi gereken bilginin büyüklüğü sebebiyle çoğu zaman FPGA'lerden yararlanılmaktadır [6]. Mobil robotlar için geliştirilmiş FPGA üzerinde çalışan GKF uygulamaları da geliştirilmektedir [7,8].

1.3. Çalışmanın Amacı ve Geliştirme Ortamı

Bu çalışmada stereo kameraya sahip, FPGA tabanlı bir robotta GKF kullanarak EZKH yapan bir donanımın sentezlenmesi amaçlanmıştır. Oluşturulması planlanan yapının genel yapısı şekil 1.1.'de görülebilir.



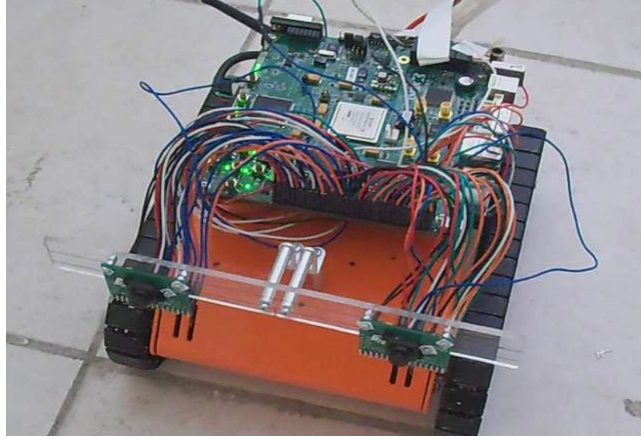
Şekil 1.2. Oluşturulması planlanan sistem

Sistem tamamlandığında robotun sahip olduğu stereo kameralardan elde edilen görüntülerde Harris köşe tespit algoritmasıyla köşe tespiti yapılacak, tespit edilen köşeler arasında stereo eşleme yapıldıktan sonra stereo sistemin ölçümleme verilerinin yardımıyla köşelerin 3 boyuttaki konumları tespit edilecektir. Tespit edilen köşelerin konum bilgileri GKF'nin ölçüm aşamasına aktarılırken filtrenin tahmin aşamasında ise robotun enkoderlerinden gelen konum değişimi bilgisi kullanılacaktır.

Çalışma üzerinde Virtex-5 XC5VLX50 FPGA çekirdeği bulunan Xilinx'e ait ML501 kartı üzerinde geliştirilmektedir. Görüntüleri depolamak için yine kartın üzerinde bulunan Micron'a MT4HTF3264HY-53E DDR2 SODIMM bellekten yararlanılmaktadır. Görüntü sensörü olarak RGB555 formatında, 640x480 çözünürlüğünde görüntü verebilen OmniVision'a ait kameralardan yararlanılmıştır. Çalışmada kullanılan robotun görüntüsü şekil 1.2.'de verilmiştir.

Çalışmada benzetim ve uygulama aşamaları iç içe geçtiğinden ilk önce robotun dış dünyayla rahat bir iletişim kurması amaçlanmıştır. Geliştirilen görüntü işleme algoritmalarını hızlı bir şekilde test etmek için ML501 kartı üzerinde bulunan Chrontel CH7301C entegresi için sürücü oluşturulmuş ve bu entegre yardımıyla test aşamasında işlenen görüntüler ekrana yansıtılmıştır. Ayrıca FPGA'deki bilgilerin bilgisayara aktarılması için Digi firmasına ait XBee modülleri için de FPGA sürücüsü oluşturulmuş ve bilgiler ZigBee protokolü ile benzetim ortamına aktarılmıştır.

Sistemin üstüne kurulduğu robot sadece hareketli bir platform sağlayarak EZKH algoritmasının test edilebilmesini sağlamaktadır. Üzerine ML501 kartı sabitlenmiş test robotu şekil 2.1.'de görünmektedir.



Şekil 1.3. Test Robotu

Yüksek lisans tezi kapsamında oluşturulan bu sistemin anlatıldığı makalenin ikinci bölümünde köşe tespit algoritmalarından, Plessey köşe tespit algoritmasından ve Plessey köşe tespit algoritması için oluşturulan FPGA tabanlı mimari anlatılmaktadır. Makalenin üçüncü bölümünde stereo eşleme algoritmalarından ve Barnard'ın stereo eşleme algoritmasından bahsedildikten sonra ölçümleme bilgileri yardımıyla köşe noktalarının robota göre konumlarının üçgenleyerek yeniden kurma yöntemiyle çıkarılmasından bahsedilmektedir. Dördüncü bölümde enkoder bilgisi ve 3 boyutta robota göre konumları tespit edilmiş olan köşe noktaları yardımıyla oluşturulan GKF algoritmasının yapısından ve benzetim sonuçlarından bahsedilmektedir. Son olarak sonuçlar bölümü ve referanslarla makale sonlanmaktadır.

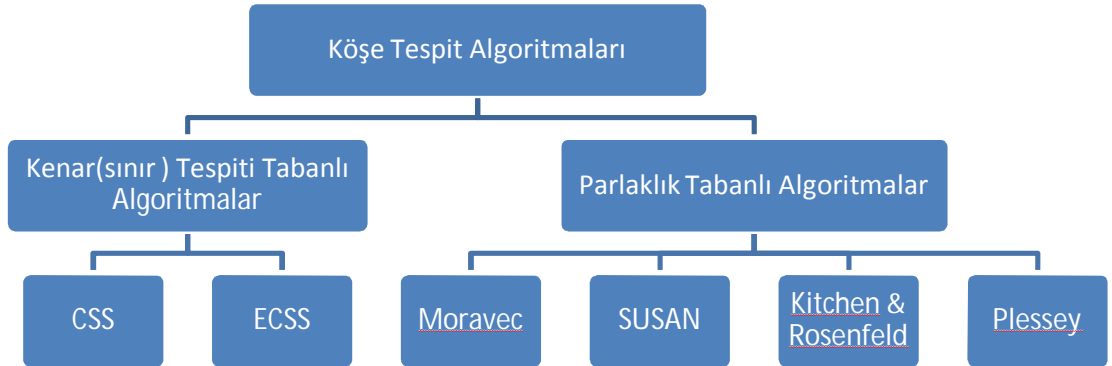
2. ÖZELLİK NOKTALARININ ÇIKARIMI

Mobil robotlar konumlarını güncellerken ve ortamın haritasını oluştururken çevrelerinde bulunan, algılayabildikleri özellik noktalarına ihtiyaç duyarlar. Basit bir şekilde iki veya daha fazla kenarın kesişim noktası olarak tanımlanabilen köşeler, ayırt edilebilir olmaları, çoğu ortamda bulunmaları ve daha karmaşık modellerin algılanmasında kullanılabilir olmaları sebebiyle görüntü işleme ve robotla görme algoritmalarında sıkça kullanılan özellikler arasındadırlar. Bu çalışmada özellik noktaları olarak robotun stereo kameralarıyla algılayacağı köşe noktalar kullanılmıştır.

2.1. Köşe Tespit Algoritmaları

Köşe tespit algoritmaları kenar(sınır) tespiti tabanlı algoritmalar ve parlaklık tabanlı algoritmalar olmak üzere şekil 2.1.'de görüldüğü gibi ikiye ayrılabilir [9].

Eğrilik ölçekli uzay(EÖU) ve genişletilmiş eğrilik ölçekli uzay(GEÖU) şeklinde örnekleri olan sınır tespiti tabanlı algoritmalar; Moravec, SUSAN ve Plessey gibi örnekleri olan parlaklık tabanlı algoritmalarla göre köşe tespitinde daha başarılıdırlar [9]. Sınır tespiti tabanlı çalışan algoritmalarda, tüm görüntüde uygulanan Canny kenar tespiti algoritmasıyla elde edilen tüm kenarlar sınıflandırılır. Bu sınıflandırma sayesinde köşe tespiti yaparken oluşabilecek hatalar azaltılmış olur.



Şekil 2.1. Köşe tespit algoritmaları

Daha düşük başarı oranına sahip parlaklık tabanlı algoritmaların avantajı ise daha az işlem gücü gereksinimine ihtiyaç duymalarıdır.

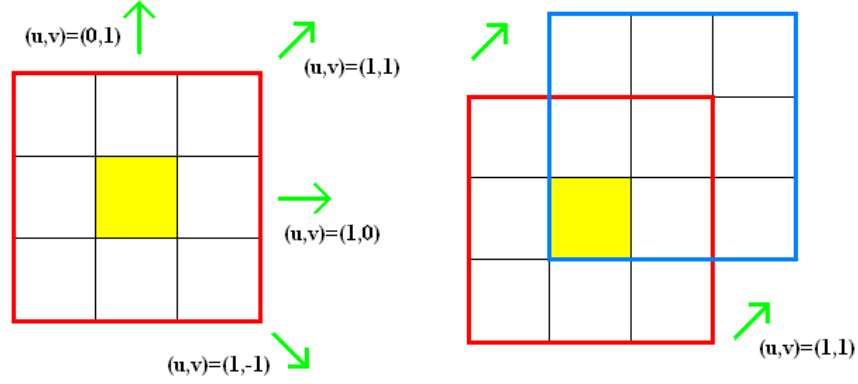
	Kenar Tespiti Tabanlı Algoritmalar	Parlaklık Tabanlı Algoritmalar
Başarı Oranı	Yüksek	Düşük
İşlem Gücü Gereksinimi	Yüksek	Düşük

Şekil 2.2. Köşe tespit algoritmalarının başarı oranı ile işlem gücü gereksiniminin kıyaslanması

Plessey(Harris) [10] köşe tespiti algoritması parlaklık tabanlı çalışan algoritmalar arasında başarımıyla öne çıkmaktadır [11-13]. Bu çalışmada FPGA üzerinde paralelleştirmeye uygun yapısı ve başarımından dolayı Plessey köşe tespiti algoritması kullanılmıştır.

2.2. Plessey(Harris) Köşe Tespit Algoritması

Plessey algoritması Moravec [14] algoritmasının eksiklerinin giderilmesiyle oluşturulmuştur. Moravec algoritmasında, her bir pikselin köşe olup olmadığına karar vermek için o pikseli merkez kabul eden belirli bir büyüklükteki alanın çevresine göre ne kadar farklı olduğuna bakılır. Şekil 2.3.'te sarı ile belirtilen pikselin 3x3 büyüklüğünde kırmızı ile işaretlenmiş komşuluğu gösterilmiştir. Bu alanın aralarında 45° fark olan 4 farklı yöne kaydırılması sonucu kapsanan alandaki piksellerin parlaklıkları ile kaydırılmadan önceki alandaki piksellerin parlaklıkları arasındaki fark hesaplanır.



Şekil 2.3. Moravec algoritmasındaki kaymalar(solda) ve (1,1) yönündeki kayma(sağda)

$$F_{u,v}(x,y) = \sum_{m,n} w(m,n) (I(x+m,y+n) - I(x+m+u,y+n+v))^2 \quad (2.1)$$

$$KOÖ = \min(F_{0,1}, F_{1,1}, F_{1,0}, F_{1,-1}) \quad (2.2)$$

I gri tonlamalı dijital görüntünün piksellerinin parlaklıklarını göstermek üzere, 2.1 eşitliğinde $F_{u,v}(x,y)$, (x,y) merkezli alanın piksellerinin parlaklık değerleri ile, alanın (u,v) yönünde kaydırılması sonucu ulaşılan alandaki piksellerin parlaklıkları arasındaki farkın toplamını göstermektedir. Denklemden m ve n ise alanın yatay ve dikeydeki büyüklüğüdür. Algoritmada parlaklık farklarının çarpılacağı katsayıları belirleyen $w(m,n)$ fonksiyonu alan içinde bir birim iken alan dışında sıfırdır. 2.2 eşitliğinde görüldüğü gibi her piksel için 4 farklı yönde hesaplanan parlaklık fark toplamlarından en küçüğü o piksel için köşe olma ölçüsü(KOÖ) olarak kabul edilir. KOÖ'leri deneysel olarak belirlenen bir eşik değerinin üzerindeki pikseller Moravec algoritmasına göre köşe özelliği göstermektedirler.

Harris [9] tarafından 1988'de, Moravec algoritmasındaki 3 eksiklik dikkate alarak köşe ve kenar tespiti için yeni bir algoritma geliştirilmiştir. Bu algoritma ile belirtilen eksikliklerden birincisi, Moravec algoritmasının sadece 45°'lik kaymalardaki

değişimleri dikkate aldığı için köşe tespitinin yönden bağımsız sonuçlar vermediğidir. Bunun için Harris algoritmasında tüm yönlerdeki değişimlere bakılması gerektiği belirtilmiştir. 4 farklı yöndeki (u,v) kaymaları yerine, kaymalar tüm küçük (Δu , Δv) yönlerini kapsayacak şekilde, 2.3 eşitliğindeki gibi geliştirilmiştir.

$$F_{\Delta u, \Delta v}(x, y) = \sum_{m,n} w(m,n)(I(x+m, y+n) - I(x+m+\Delta u, y+n+\Delta v))^2 \quad (2.3)$$

I_x ve I_y sırasıyla x ve y yönündeki kısmi türevleri ifade etmek üzere, 2.4 eşitliğindeki gibi Taylor açılımı kullanılarak

$$I(x+m+\Delta u, y+n+\Delta v) \approx I(x+m, y+n) + I_x(x+m, y+n)\Delta x + I_y(x+m, y+n)\Delta y \quad (2.4)$$

2.3 eşitliği 2.5'deki haline dönüştürülür.

$$F_{\Delta u, \Delta v}(x, y) = \sum_{m,n} w(m,n)(I_x(x+m, y+n)\Delta x + I_y(x+m, y+n)\Delta y)^2 \quad (2.5)$$

Harris'in Moravec algoritmasında gösterdiği ikinci eksiklik ise algoritmada kullanılan $w(m,n)$ fonksiyonunun yalnızca iki değer alması sebebiyle algoritmanın gürültüye karşı dayanıksız olmasıdır. KOÖ hesaplanan pikselin komşuluğu içerisinde kalan tüm piksellerin KOÖ'nü merkeze yakın veya uzakta olmasına bakılmaksızın aynı seviyede etkileme gücüne sahip olmalarından dolayı oluşturacakları gürültüler $w(m,n)$ fonksiyonunun katsayılarının Gaussian dağılımına uygun olarak seçilmesiyle 2.7 eşitliğinde gösterildiği gibi engellenmiştir.

$$w(m, n) = \exp \frac{-(m^2+n^2)}{2\sigma^2} \quad (2.6)$$

$$F_{\Delta u, \Delta v}(x, y) = \sum_{m, n} \exp \frac{-(m^2+n^2)}{2\sigma^2} (I_x(x+m, y+n)\Delta x + I_y(x+m, y+n)\Delta y)^2 \quad (2.7)$$

Sadeleştirmelerle eşitlik 2.7'ye ulaşılır.

$$F_{\Delta u, \Delta v}(x, y) = \sum_w (I_x \Delta x + I_y \Delta y)^2 \quad (2.8)$$

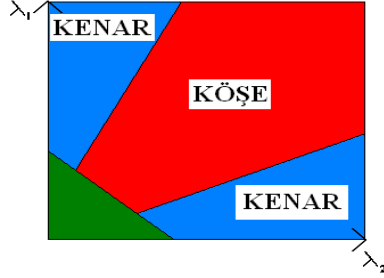
2.8 eşitliği matris formuna dönüştürülürse 2.9 eşitliğine ulaşılır.

$$F_{\Delta u, \Delta v}(x, y) = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} M(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2.9)$$

Burada 2x2 boyutlarında olan M matrisi, eşitlik 2.10'da gösterildiği gibi komşuluk sınırları içerisinde kalan piksellerin $G_{\zeta_{xx}}$, $G_{\zeta_{yy}}$ ve $G_{\zeta_{xy}}$ gradyan çarpımlarıyla oluşturulmuştur.

$$M(x, y) = \begin{bmatrix} \sum_w I_x^2 & \sum_w I_x I_y \\ \sum_w I_x I_y & \sum_w I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_w G_{\zeta_{xx}} & \sum_w G_{\zeta_{xy}} \\ \sum_w G_{\zeta_{xy}} & \sum_w G_{\zeta_{yy}} \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (2.10)$$

Harris tarafından diagonal olan M matrisinin özdeğerlerinin yerel oto korelasyon fonksiyonunun kavislenmesi ile orantılı olduğunu belirtilmiştir. Şekil 2.4.'te görüldüğü gibi matrisin özdeğerlerinden (λ_1, λ_2) ikisinin de belirli bir eşik değerinin üzerinde olmaları o pikselin köşe olduğuna, bir özdeğerin diğerine göre çok büyük olmasının ise pikselin kenar olduğuna işaret etmektedir. Özdeğerlerden ikisi de belirli bir büyüklüğün altında ise o pikselin köşe veya kenar özelliği göstermemektedir. Böylece Moravec algoritmasındaki 3.sorun olan kenar ve köşe noktaların yakın KOÖ'lerine sahip olması sorunu çözülmüştür.



Şekil 2.4. Özdeğerlerin büyüklüğüne göre pikselin sınıflandırılması.

M matrisinin özdeğerlerinin hesaplanmasının fazla zaman almasından daha kolay hesaplanabilen bir köşe veya kenar olma ölçüsü(KKOÖ) önerilmiştir.

$$KKOÖ = \det(M) - k.Tr(M)^2 = AB - C^2 - k(A + B)^2 \quad (2.11)$$

Eşitlikteki k değiştirilebilen bir sabit olmak üzere; elde edilen KKOÖ pozitif ve belirli bir büyüklükte ise piksel köşeye, negatif ve belirli bir büyüklükte ise kenara karşılık gelmektedir.

Montesinos [15] tarafından Harris'in gri tonlamalı görüntüler için geliştirdiği yöntem renkli görüntülerde kullanılmıştır. Renkli görüntülerde M matrisi 2.12 eşitliğindeki gibi oluşturulmaktadır.

$$M(x, y) = \begin{bmatrix} \sum_w K_x^2 + Y_x^2 + M_x^2 & \sum_w K_x K_y + Y_x Y_y + M_x M_y \\ \sum_w K_x K_y + Y_x Y_y + M_x M_y & \sum_w K_y^2 + Y_y^2 + M_y^2 \end{bmatrix} \quad (2.12)$$

Renkli görüntülerde M matrisi oluşturulurken, gri tonlamalı görüntülerde hesaplanan $G_{\zeta_{xx}}$, $G_{\zeta_{yy}}$ ve $G_{\zeta_{xy}}$ gradyan çarpımları yerine 3 renk bandındaki gradyan çarpımlarıyla oluşturulan $G_{\zeta_{xx}}$, $G_{\zeta_{yy}}$ ve $G_{\zeta_{xy}}$ gradyan çarpım toplamları oluşturulmuş olur.

$$M(x, y) = \begin{bmatrix} \sum_w G_{\zeta t_{xx}} & \sum_w G_{\zeta t_{xy}} \\ \sum_w G_{\zeta t_{xy}} & \sum_w G_{\zeta t_{yy}} \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (2.13)$$

Bu eşitlikte K_x , Y_x ve M_x sırasıyla pikselin kırmızı, yeşil ve mavi bantlardaki parlaklıklarının yatay gradyanlarını belirtirken K_y , Y_y ve M_y de bantlardaki parlaklık değerlerinin dikey gradyanları belirtmektedir. Renkli görüntülerdeki pikseller için de M matrisleri oluşturulduktan sonra gri tonlamalı görüntülerde kullanılan yöntemle KKOÖ hesaplanır ve pikselin köşe olup olmadığına karar verilir.

Çizelge 2.1. Gradyan çarpımlarının bulunması

```

1:   Gradyan_Çarpımlarının_Toplamını_Bul( Ik, Iy, Im, M)
2:   m ← büyüklük(M)+1;
3:   toplam ← M(1,4)+ M(1,5)+ M(2,4)+ M(2,5)+ M(3,4)+
           M(3,5)+ M(4,4)+ M(4,5)+ M(5,4)+ M(5,5);
4:   Gk,x ← 0; Gk,y ← 0; Gy,x ← 0; Gy,y ← 0; Gm,x ← 0; Gm,y ← 0;
5:   for a ← 1 to büyüklük(M)
6:     for b ← ((m/2)+1) büyüklük(M)
7:       Gk,x ← ( M(a,b)*(Ik(a,b)- Ik(a,m-b) ) + Gk,x;
8:       Gk,y ← ( M(b,a)*(Ik(b,a)- Ik(m-b,a) ) + Gk,y;
9:       Gy,x ← ( M(a,b)*(Iy(a,b)- Iy(a,m-b) ) + Gy,x;
10:      Gy,y ← ( M(b,a)*(Iy(b,a)- Iy(m-b,a) ) + Gy,y;
11:      Gm,x ← ( M(a,b)*(Im(a,b)- Im(a,m-b) ) + Gm,x;
12:      Gm,y ← ( M(b,a)*(Im(b,a)- Im(m-b,a) ) + Gm,y;
13:     endfor
14:   endfor
15:   Kx ← Gk,x/ toplam;
16:   Ky ← Gk,y/ toplam;
17:   Yx ← Gy,x/ toplam;
18:   Yy ← Gy,y/ toplam;
19:   Mx ← Gm,x/ toplam;
20:   My ← Gm,y/ toplam;
21:   Gçtxx ← Kx * Kx + Yx * Yx + Mx * Mx;
22:   Gçtyy ← Ky * Ky + Yy * Yy + My * My;
23:   Gçtxy ← Kx * Ky + Yx * Yy + Mx * My;
24:   return (Gçtxx, Gçtyy, Gçtxy);
25: endfunction

```

Algoritmalar incelendiğinde bir pikselin KKOÖ'nün bulunabilmesi için o pikselin belirlenen büyüklükteki komşuluğunda bulunan tüm piksellerin gradyan çarpımlarına ihtiyaç duyulduğu görülmektedir. Dolayısıyla köşe tespiti için tasarlanan boru hattı yapısının birbirinden bağımsız ancak sıralı olarak çalışacak 2 ayrı donanım modülünden oluşmasına karar verilmiştir. Buna göre algoritma iki fonksiyona ayrılmış ve bu iki parça için ayrı ayrı donanım modülleri tasarlanmıştır.

Algoritma çizelge 2.1.'de gösterilen ilk fonksiyon bir piksel için $G_{\text{çt}_{xx}}$, $G_{\text{çt}_{yy}}$ ve $G_{\text{çt}_{xy}}$ gradyan çarpım toplamlarını hesaplar. Tabloda fonksiyona girdi olarak giren I_k , I_y ve I_m söz konusu pikselin komşuluğunda bulunan piksellerin sırasıyla kırmızı, yeşil ve mavi bantlardaki parlaklık değerlerini göstermektedir. Fonksiyonun diğer girdisi olan M ise gradyan hesaplamasında kullanılan katsayıları içeren maskedir. Algoritmanın gürültüden olumsuz etkilenmesini azaltmak için bu maskenin katsayıları Gaussian dağılımında seçilmiştir.

KKOÖ'nü hesaplayan ikinci fonksiyon ise çizelge 2.2.'de gösterilmiştir. Bu fonksiyon KKOÖ hesaplanan pikselin komşuluğundaki her bir pikselin $G_{\text{çt}_{xx}}$, $G_{\text{çt}_{yy}}$ ve $G_{\text{çt}_{xy}}$ gradyan çarpım toplamlarını içeren $G_{\text{çt}_{xx}}$, $G_{\text{çt}_{yy}}$ ve $G_{\text{çt}_{xy}}$ girdilerini alır. Ayrıca katsayıları Gaussian dağılımında olan N maskesi ve algoritmada kullanılan k sabiti de fonksiyona girdi olarak girmektedir.

Çizelge 2.2. KKOÖ'nün hesaplanması

```

1: Kenar_Köşe_Olma_Ölçüsünü_Çıkar( $G_{\text{çt}_{xx}}$ ,  $G_{\text{çt}_{yy}}$ ,  $G_{\text{çt}_{xy}}$ ,  $N$ ,  $k$ )
2:    $A \leftarrow 0$ ;  $B \leftarrow 0$ ;  $C \leftarrow 0$ ;
3:   for  $i \leftarrow 1$  to sizeof( $N$ )
4:     for  $j \leftarrow 1$  to sizeof( $N$ )
5:        $A \leftarrow (N(i,j) * G_{\text{çt}_{xx}}(i,j)) + A$ ;
6:        $B \leftarrow (N(i,j) * G_{\text{çt}_{yy}}(i,j)) + B$ ;
7:        $C \leftarrow (N(i,j) * G_{\text{çt}_{xy}}(i,j)) + C$ ;
8:     endfor
9:   endfor
10:   $A \leftarrow A / \text{sum}(N)$ ;
11:   $B \leftarrow B / \text{sum}(N)$ ;
12:   $C \leftarrow C / \text{sum}(N)$ ;
13:   $\text{KKOÖ} \leftarrow A * B - C^2 - k * (A + B)^2$ ;
14:  return  $\text{KKOÖ}$ ;
15: endfunction

```

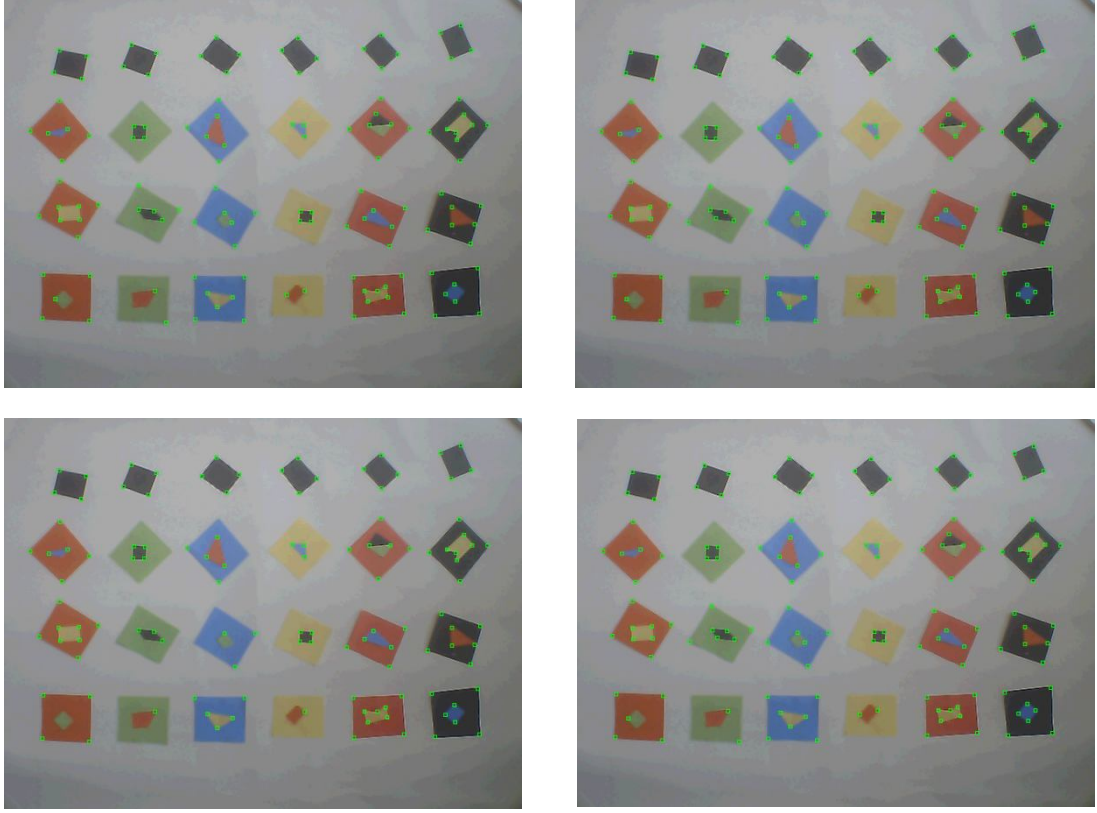
2.3. Test Aşaması

Algoritmanın uygulanacağı donanımı sentezlemeden önce algoritmadaki sabitlerin belirlenmesi için robot ile alınan test görüntüleri bilgisayar ortamına aktarılmıştır. Aktarılan bu görüntüler ile ilk aşamada kullanılan Gaussian maskesinin büyüklüğü ve değişkesi, ikinci aşamada kullanılan Gaussian maskesinin büyüklüğü ve değişkesi ile birlikte ikinci aşamada KKOÖ çıkarılırken kullanılan k sabitinin değeri hesaplanmıştır. Algoritmanın başarımını ve maskelerin donanımda kaplayacağı alan düşünülerek iki aşamada da kullanılan maskelerin büyüklüğünün 5x5 olmasına karar verilmiştir. Birinci ve ikinci aşamada kullanılacak olan Gaussian maskelerinin değişkesi için kaydedilen görüntü üzerinde testler yapılmıştır. Buna göre maskelerin değişkesi 1,2 olacak şekilde uygulanmasına karar verilmiştir. Şekil 2.5.'te, 5x5 boyutunda bir maskede 1,2'lik değişkeye sahip, ayrık, 2 boyutlu gaussian dağılımının katsayılarının tam sayıya dönüştürülmüş şekli gösterilmiştir.

1	2	3	2	1
2	6	8	6	2
3	8	12	8	3
2	6	8	6	2
1	2	3	2	1

Şekil 2.5. 5x5lik bir maske için değişkesi 1,2 olan, ayrık, 2 boyutlu Gaussian dağılımının katsayılarının tamsayıya dönüştürülmüş şekli

Şekil 2.6.'da gösterilen test sonuçlarında, farklı değişke değerleri için tespit edilen köşe pikselleri yeşil pencere ile işaretlenmiştir. Genellikle uygulamalarda 0,04 ile 0,06 arasında seçilen k sabiti ise ikilik sistemde uygulama kolaylığı oluşturması açısından 0,0625(=1/16) olarak seçilmiştir. Bu sabit ikinin tam katı olduğundan bu sabit ile yapılacak çarpma ve bölme işlemleri donanımda saat vuruşu(SV) gerektirmeden, bitleri kaydırarak gerçekleştirilir.



Şekil 2.6. Farklı özdeğerler için tespit edilen köşeler. ($\sigma_M=1.2$, $\sigma_N=1.2$ sol üst),
($\sigma_M=1.2$, $\sigma_N=1.8$ sağ üst), ($\sigma_M=1.8$, $\sigma_N=1.2$ sol alt), ($\sigma_M=1.8$, $\sigma_N=1.8$ sağ alt)

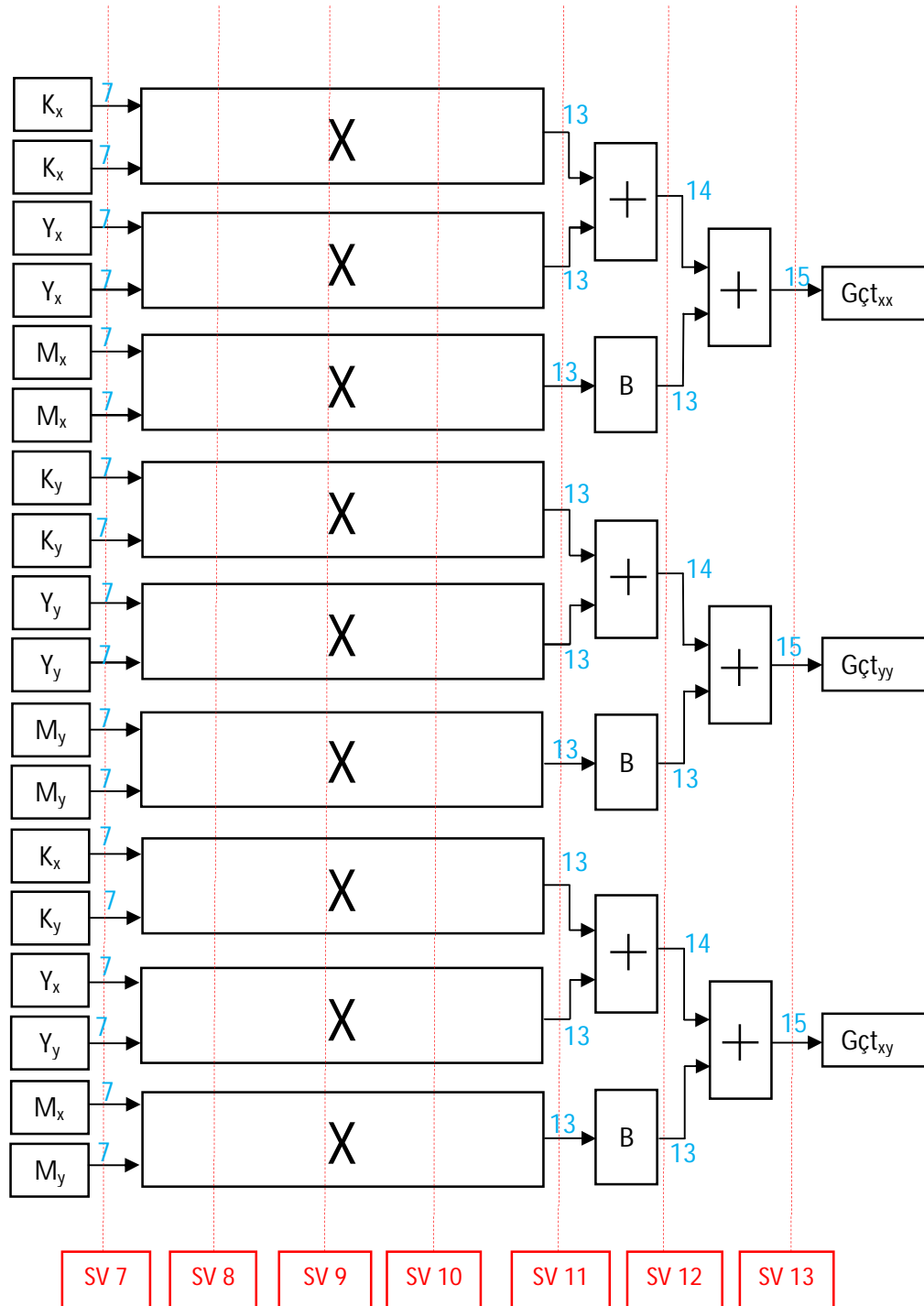
2.4. Algoritmaya Yönelik Boru Hattı Yapısının Oluşturulması

Daha önceki çalışmalarda [16] tasarlanan FPGA mimarisi bu çalışmada boru hattı yapısına dönüştürülerek algoritmanın çalışma hızı artırılmıştır. Donanım tasarlanırken toplama ve çıkarma işlemleri için Xilinx'in ISE webpack programının yazılan Verilog HDL kodundan donanımı sentezlemesine izin verilirken çarpma işlemleri için yine Xilinx'in Core Generator programıyla çarpıcı modüller oluşturulup bu modüller ana kodun içerisine gömülmüştür.

İlk aşama için oluşturulan modül, gradyan çarpım toplamı hesaplanacak olan pikselin 5x5'lik komşuluğunda bulunan piksellerin tüm parlaklık değerlerini paralel olarak alır. Bu durum şekil 2.7.'de gözlemlenebilir. Kırmızı banttaki parlaklık değerleri kullanılarak x yönündeki parlaklık değişimi K_x ' in hesaplanması için hazırlanan

donanımı gösteren modül, ilk SV'dan önce I_k ile belirtilen parlaklık değerlerini almaktadır. İlk SV'da alınan bu 5 bitlik parlaklık değerleri paralel çıkarıcılara aktarılır. Parlaklık değerleri arasındaki çıkarma işlemi 1 SV'da bitmektedir. Şekil 2.5.'te görüleceği gibi bu parlaklık farkları 1,2,3,6 ve 8 katsayılarıyla çarpılmaktadır. 2 ve 8 katsayılarıyla çarpılacak olan parlaklık farkları için SV gerektirmeyen kaydırma işlemleri kullanılmıştır. 1 ve 2 katsayıları ile çarpılan parlaklık farkları 2. SV'da paralel toplayıcılarla toplanmaya başlanmıştır. 8 katsayısı ile çarpılan parlaklık farkı ise 2. SV'da toplanacak eşinin bulunmaması sebebiyle 2. SV için geçici belleğe yazılmıştır. 3 ve 6 katsayıları ile çarpılacak farklar için ise oluşturulan 1 SV gecikmeye sahip çarpıcı modülleri kullanılmıştır. 3. SV'dan 6. SV'na kadar devam eden paralel toplama işlemleri ile 7. SV öncesi K_x değeri hesaplanır. Çizelge 2.1.'in 15 ile 20. satırları arasındaki bölme işlemleri hem modülün kapladığı alanı hem de modülün toplam SV sayısını arttırmamak için çizelge de belirtildiği gibi bir bölme işlemi uygulanmaz. Seçilen değişke değeri için çizelge de belirtilen toplam değeri 33'tür. Bu bölme işlemi yerine elde edilen 12 bitlik K_x değeri 32'ye bölünerek anlamsız bitlerin getireceği yük azaltılır.

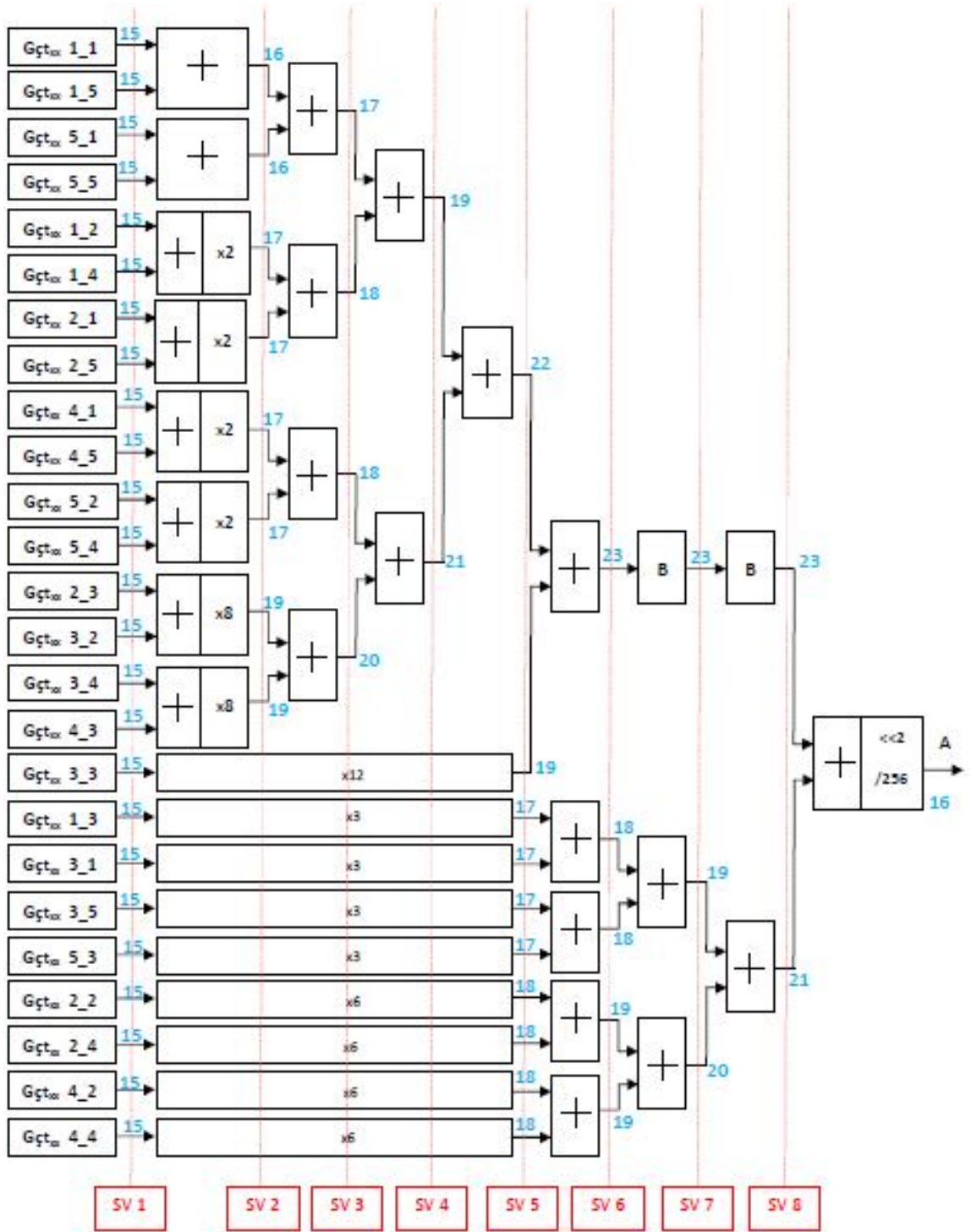
Şekil 2.7.'de gösterilen ve K_x hesaplamasında kullanılan modüle benzer yapıda ve bu modüle paralel olarak çalışan K_y , Y_x , Y_y , M_x , M_y hesaplamasında kullanılan modüller de oluşturulmuştur. İlk aşama içerisinde bütün bu modüllerin sağladığı gradyan çarpımları kullanılarak gradyan çarpım toplamları hesaplanır. Bu yapı şekil 2.8.'de gösterilmektedir. 7. SV'dan önce tüm gradyan değerleri 3 SV gecikmeye sahip çarpıcılara aktarılmıştır. 11. SV'da hesaplanan çarpımlar toplanmaya başlanır ve 13.SV'da boru hattı $G_{çt_{xx}}$, $G_{çt_{yy}}$ ve $G_{çt_{xy}}$ hesaplanmış olur. Oluşturulan bu yapı her SV'da yeni bir pikselin $G_{çt_{xx}}$, $G_{çt_{yy}}$ ve $G_{çt_{xy}}$ değerlerinin hesaplanmasına imkân sağlar. Eğer bu modül her SV'da yeni bir pikselin komşuluğunda bulunan parlaklık değerleriyle beslenirse modül 13 SV gecikme sonrasında söz konusu her pikselin $G_{çt_{xx}}$, $G_{çt_{yy}}$ ve $G_{çt_{xy}}$ değerlerini dış modüllere aktarabilir.



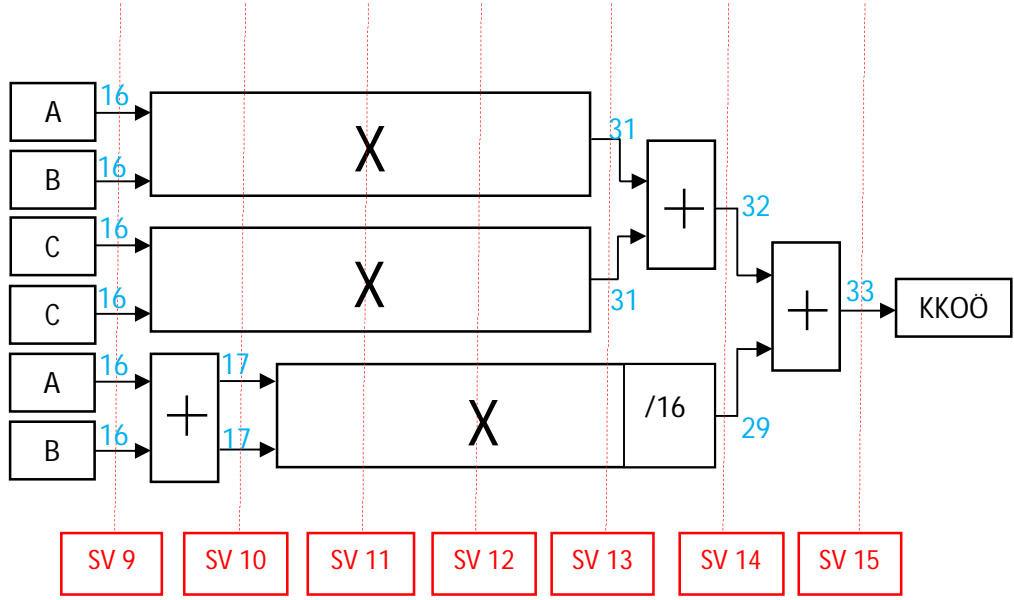
Şekil 2.8. Birinci aşama için oluşturulan modülün ikinci parçası

İkinci aşama için oluşturulan modül de birinci aşama için oluşturulan modüle benzer yapıdadır. KKOÖ hesaplanacak olan pikselin A değerinin hesaplanmasında kullanılmak üzere, pikselin kendi $G_{ct_{xx}}$ değeri ile birlikte, pikselin 5x5'lik komşuluğunda yer alan piksellerin $G_{ct_{xx}}$ değerleri Şekil 2.9.'da gösterilen ikinci modülün ilk parçasına paralel olarak aktarılır. Şekil 2.5.'te gösterilen katsayılara göre; $G_{ct_{xx}}$ değeri 1,2,4 ve 8 ile çarpılacak olan piksellerin $G_{ct_{xx}}$ değerleri toplanmaya başlanarak gerekli çarpma işlemleri SV gerektirmeyecek şekilde bitlerin kaydırılmasıyla sağlanır. $G_{ct_{xx}}$ değerleri 3,6 ve 12 katsayılarıyla çarpılacak olan pikseller ise 3 SV gecikmeye sahip çarpıcılara aktarılırlar. Çarpma işlemleri ve toplama işlemleri 8 SV da boru hattı yapısına uygun bir şekilde tamamlanır. 8 SV'da ortaya çıkan 24 bitlik A değerinin 24 biti olan işaret bitinden önce gelen 22 ve 23. bitlerinin 0 olacağı kesindir. Bu nedenle bu iki bit ihmal edilerek A değeri 22 bite indirgenmiş olunur. Çizelge 2.2.'nin 10 ile 12. satırları arasında yapılan bölme işlemleri, gradyan çarpımı hesabında olduğu gibi fazla SV istemesi ve fazladan donanım kaplaması sebebiyle gerçekleştirilmez. Bunun yerine 22 bite indirgenen A değeri 64'e bölünerek A değerinin 16 bit olması sağlanır. Bu işlemde 64'ün seçilmesinin sebebi, 64'ün 100'den küçük sayılar arasında 2'nin katı olan en büyük sayı olmasındandır.

KKOÖ hesaplanacak olan pikselin A değerinin hesaplanmasında kullanılan şekil 2.9.'daki donanım yapısına benzer yapıda, B ve C değerlerinin hesaplanmasında da kullanılmak üzere donanım parçaları sentezlenmiştir. Bu üç parça ikinci modül içerisinde birbirine paralel olarak çalışırlar. 16 bitlik bu A, B ve C değerleri şekil 2.10.'da gösterilen ikinci modülün diğer parçasına aktarılır. 9. SV 'da 3 SV gecikmeye sahip çarpıcılara aktarılan A, B ve C değerlerinin yanı sıra aynı saat vuruşunda A ve B değerleri 1 SV'da çalışan toplayıcıya aktarılır. 10.SV'da çarpma işlemleri devam ederken toplayıcı sonucu, sonucun karesini almak üzere 3 SV gecikmeye sahip başka bir çarpıcıya aktarılır.



Şekil 2.9. İkinci aşama için oluşturulan modülün A değerini hesaplayan parçası



Şekil 2.10. İkinci aşama için oluşturulan modülün KKOÖ'nü hesaplayan parçası

9. SV'da çarpıcılara aktarılan değerlerin sonucu 13. SV'da elde edilmiş olur ve sonuçlar bu SV'da toplanır. Daha önce belirtildiği gibi algoritmadaki k değişkeni 0,0625 olarak seçildiğinden 10. SV'da başlayan çarpma işleminin sonucun bitleri kaydırılarak k değişkeni ile çarpma işlemi gerçekleştirilmiş olunur. 14. SV'da son toplama işlemi gerçekleştirilerek söz konusu pikselin KKOÖ toplam 15 SV da hesaplanmış olur.

Bu çalışmada Plessey algoritmasının sadece köşe bulma özelliğinden yararlanıldığı için; KKOÖ hesaplandıktan sonra bu değer belirli bir eşik değeri ile kıyaslanır ve pozitif olup eşik değerinin üzerinde KKOÖ'ne sahip pikseller köşe adayı olarak kabul edilerek bu değerler belleğe aktarılır. KKOÖ bu kritere uymayan piksellerin KKOÖ 0 olarak kaydedilir. Daha sonra belleğe kaydedilmiş pikseller bellekten tekrar okunur ve belirli bir alanda yalnızca bir köşenin tespit edilmesini sağlayan maksimum olmayanı baskılama yöntemi uygulanır. Bu çalışmadaki uygulamalarda bu alan yatayda ve düşeyde en az 4 en fazla 8 piksel olacak şekilde belirlenmiştir.

2.5. Köşe Tespit Modüllerinin Performans Sonuçları

Sentez sonuçlarına göre toplam 13 boru hattı aşamasına sahip olan birinci modül 429MHz'lik bir maksimum çalışma frekansına sahipken, 15 boru hattı aşamasına sahip olan ikinci modülün maksimum çalışma frekansı 365MHz'dir. Buna göre bu iki modülden 1er adet kullanılarak oluşturulan bir yapı 640x480 çözünürlükte olan RGB555 formatındaki renkli görüntü üzerinde DDR2 bellek gecikmeleri hariç 1,53ms'de köşe tespiti gerçekleştirebilmektedir.

Çizelge 2.3. Kullanılan donanım miktarı

	1.Aşama		2.Aşama		Toplam		XC5VLX50
	Adet	%	Adet	%	Adet	%	Adet
Dilim Belleği	1834	6,37	3194	11,09	5028	17,46	28800
LUT	1912	6,64	2662	9,24	4574	15,88	28800
DSP48	0	0	3	6,25	6,25	6,25	48

Çizelge 2.3.'te bu modüller için gerekli olan donanım kaynakları listelenmiştir. Buna göre köşe tespiti için tasarlanan modüllerden 1er tane kullanılarak oluşturulacak sistemde köşe tespiti için gerekli donanım Virtex-5 XC5VLX50 gibi küçük bir çekirdeğin %20'sinden daha az bir kısmı kaplamaktadır.

Tasarlanan boru hattı yapısı, dış modüller tarafından verimli bir şekilde beslenebilirse, modüllerin aşama sayısı algoritmanın çalışma zamanını önemli seviyede etkilemeyecektir. Modüllerden 1er tane kullanıldığı ve bunların tam verimle beslendiği varsayımı yapılırsa, 200MHz'lik bir çalışma hızında bile RGB555 formatında, renkli ve 640x480 çözünürlüğünde olan bir görüntü üzerinde yaklaşık olarak 3ms'de, 1280x1024 çözünürlüğünde olan bir görüntü üzerinde 12,8ms'de ve 2592x1944 çözünürlüğünde olan bir görüntü üzerinde ise 50ms'de köşe tespiti gerçekleştirilebilir.

3. STERYO EŞLEME VE 3 BOYUTTA KONUM ÇIKARIMI

Sağ ve sol görüntü sensörlerinden gelen görüntülere Plessey köşe tespiti algoritması uygulandıktan sonra tespit edilen köşeler arasında stereo eşleme yapılmıştır. Eşleştirilen köşeler arasındaki uzaklık farkı bilgisi 3 boyutta konum çıkarımı aşamasına aktarılarak tespit edilen bu köşenin robota göre 3 boyuttaki konumu tespit edilecektir. Stereo eşleme, aynı anda farklı konumlardan alınan iki görüntü arasında, aynı konumdan farklı zamanlarda alınan iki görüntü arasında veya farklı konumlardan farklı zamanlarda alınan görüntüler arasında yapılabilir.

3.1. Stereo Eşleme Algoritmaları

Stereo eşleme algoritmaları bölge tabanlı çalışan algoritmalar ve özellik tabanlı çalışan algoritmalar olarak ikiye ayrılabilir [17]. Bölge tabanlı algoritmalar eşleştirilecek görüntülerden birini referans alır. Referans alınan görüntü belirlenen büyüklükteki parçalara bölünür. Daha sonra diğer görüntüde bu parçaların parlaklık değerlerine benzer parçalar aranır. Özellik tabanlı algoritmalarda ise görüntü üzerinde özellik çıkarımı yapıldıktan sonra çıkarılan bu özellik noktaları üzerinden stereo eşleme gerçekleştirirler. Özellik tabanlı algoritmaların karşılaştırma yaptıkları uzay, alan tabanlı algoritmalara göre daha küçük olduğundan özellik tabanlı algoritmalar daha kısa sürede eşleştirmeyi tamamlar. Bu algoritmaların başarımları da alan tabanlılara göre daha düşüktür.

3.2. Kullanılan Eşleme Algoritması

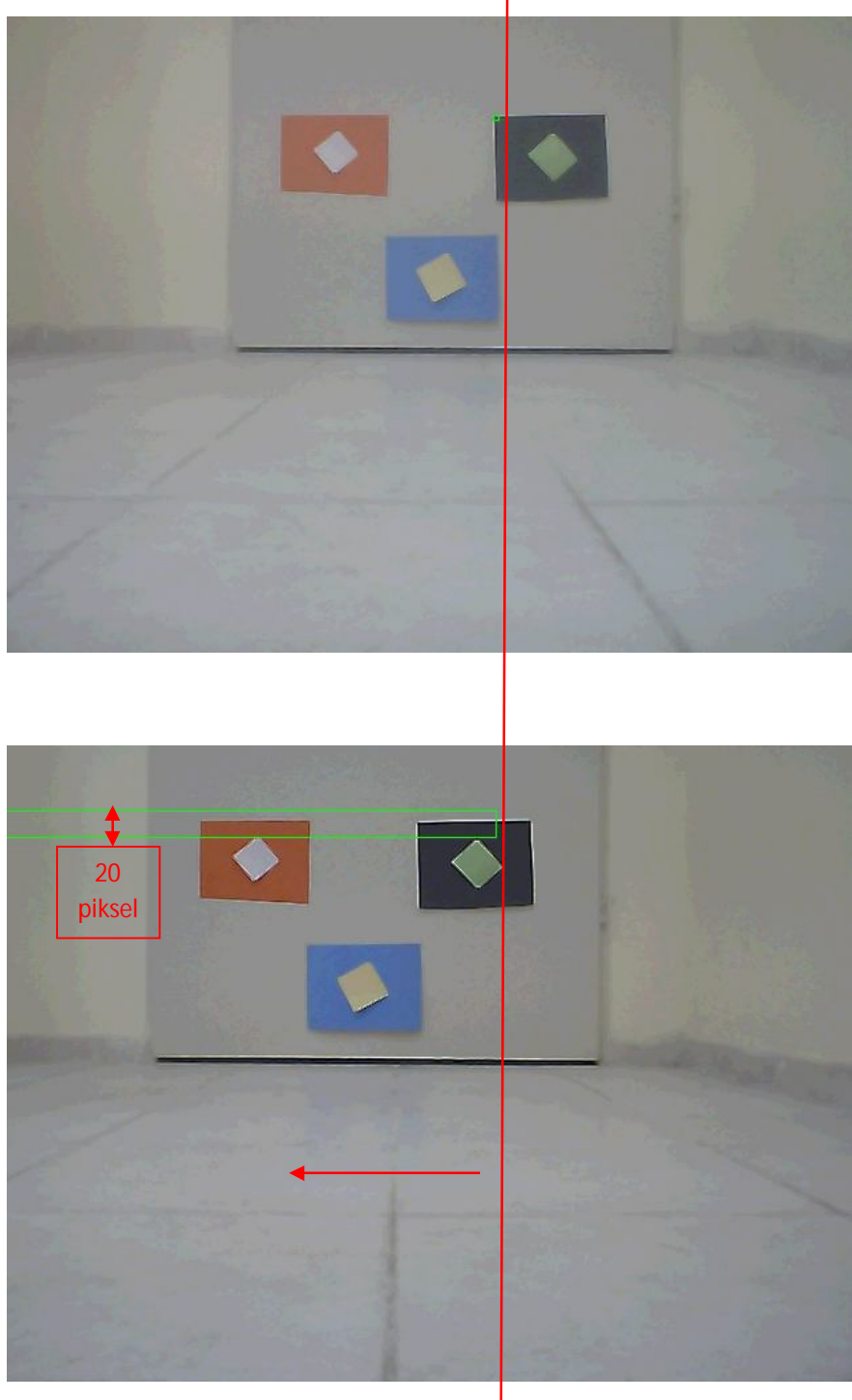
Bu çalışmada, Barnard ve Thompson tarafından 1980 yılında geliştirilen ve köşe noktaları arasında stereo eşleme gerçekleştiren olasılıksal algoritma kullanılmıştır [18]. Bu algoritma ile stereo eşleme yaparken kameraların birbirlerine göre yönleri ve konumları, kameraların lenslerinden kaynaklanan bozunmalar, kameraların odak uzaklıkları ve kameraların izdüşüm merkezleri gibi bilgilerin kesin değerlerine gerek duyulmamaktadır. Algoritmada temel olarak eşleştirilecek olan görüntülerdeki farklılık, benzerlik ve tutarlılık özelliklerinden yararlanılır. Görüntülerden biri

referans olarak seçilir. Daha sonra bu görüntüde daha önceki aşamada tespit edilmiş her bir köşe için, diğer görüntüde bu köşelere eş olabilecek köşelerin eş olma olasılıklarının ilk değerleri hesaplanır. Hesaplanan bu ilk olasılık değerleri belirli sayıda güncellenerek eş adaylarından eş olmaya uygun olanın olup olmadığı; eğer uygun olan varsa bunun hangi eş adayı olduğunu tespit edilir.

Algoritmada görüntülerin farklılık özelliğinden yararlanmak için bir köşe tespiti algoritması kullanılmaktadır. Böylece köşe pikseller ile diğerleri arasındaki fark açıkça ortaya koyulmaktadır. Önerilen Moravec algoritması yerine bu çalışmada Moravec algoritmasının geliştirilmiş hali olan ve daha önce belirtildiği gibi ona göre daha iyi sonuçlar vermekte olan Plessey köşe tespit algoritması kullanılmıştır. Görüntüler arasındaki benzerlik özelliği köşe adaylarının ilk olasılık değerleri atanırken kullanılır. Referans görüntüsündeki köşeleri merkez kabul eden ve bu köşelerin belirli bir büyüklükteki komşuluğunda bulunan alandaki piksellerin parlaklık değerleri ile diğer görüntüde bu köşelere eş olabilecek köşeleri merkez kabul eden ve bu köşelerin belirli bir büyüklükteki komşuluğunda bulunan alandaki piksellerin parlaklık değerleri arasındaki fark hesaplanır. Bu fark köşe adaylarına ilk değer atanırken kullanılır. Eşleştirme yapılacak görüntüler arasında olduğu belirtilen üçüncü ve son özellik olan tutarlılık ise eş adaylarının ilk olasılık değerlerinden yola çıkılarak yapılan olasılık güncellemelerinde kullanılır. Tutarlılık özelliğine göre referans görüntüsündeki bir köşenin diğer görüntüde kendisine belirli sütun sayısında uzakta olan bir eşi varsa; referans görüntüsünde bu köşeye belirli bir yakınlıkta olan köşelerin de diğer görüntüde benzer sütun sayısı uzaklığında eşleri vardır. Algoritmada bu tutarlılığa göre köşelerin doğru eşleriyle eşleşme olasılıkları artmaktadır.

Referans alınan görüntü G1 eş adaylarının bulunduğu görüntü de G2 olarak tanımlanacak olursa; hatasız çalışan herhangi bir stereo eşleme algoritması gibi kullanılan bu algoritma da G1 ve G2 görüntüsündeki her bir köşenin diğer görüntüde en fazla 1 eşini bulmaktadır. Köşelerin hiç eşinin bulunamaması durumu görüntülerdeki köşelerin gölgelenmesinden, objelerin arkasında kalmasından veya

bir görüntünün kapsadığı alandaki köşenin diğer görüntünün kapsama alanına girememesi durumlarından kaynaklanabilir.



Şekil 3.1. Eş adaylarının bulunabileceği alan (G1 üstte, G2 altta)

Algoritma, köşe adaylarına ilk eş olma olasılıklarının atanması ve bu olasılıkların belirli sayıda güncellenmesi şeklinde iki bölümde incelenebilir. Birinci aşamada G1 görüntüsünde tespit edilen her köşe için G2 görüntüsünde tespit edilen köşelerden konumu G1 görüntüsündeki köşeye eş olabilecek adaylar seçilir.

Şekil 3.1.'de gösterilen test görüntülerinde G1 sol, G2 ise sağ kamera ile alınan görüntüleri göstermektedir. Şekilde üstte G1 üzerinde tespit edilen bir köşe gösterilirken altta bu köşenin G2'de bulunabilecek eş adaylarının konumu yeşil pencere ile işaretlenmiştir. G1'de bir köşenin G2'deki eş adayları şekilde kırmızı okla gösterildiği gibi kendisinin daha solunda bulunmak zorundadır. Ayrıca eş adaylarının bulunabileceği alanın şekilde görüldüğü gibi dikeyde de bir toleransa sahip olması gerekmektedir. Bu çalışmada dikeydeki tolerans 20 piksel olarak seçilmiştir.

G1'deki her köşe için G2'de kendisine eş olabilecek her köşe için eş olma olasılıklarının ilk değerleri ve G1'deki köşenin hiç eşi olmaması olasılığının ilk değerleri atanır. Bunun için ilk olarak her köşenin her eş adayı için bir ağırlık hesaplanır. G1'deki i köşenin koordinatları (x_i, y_i) , G2'deki köşelerin koordinatları (x', y') , i köşesinin (x', y') ile eşleşme durumu $l=(x_i-x', y_i-y')$ ve i köşesinin hiçbir köşeye eşleşmeme durumu da l^* ile gösterilmek üzere her i köşesinin kendinden l uzaklıktaki eş adayı ile eşleşme ağırlığı eşitlik 3.1'e göre hesaplanır.

$$w_i(l) = \frac{1}{1 + c * s_i(l)}, l \neq l^* \quad (3.1)$$

Eşitlikteki c sabiti 1/128 olarak alınmıştır. Eşitlikteki $s_i(l)$ değeri ise i köşesini merkez kabul eden ve bu çalışmada 5 olarak alınan bir komşuluğundaki piksellerin parlaklık değerleri ile l eş adayı merkez kabul eden komşuluktaki piksellerin parlaklıkları arasındaki farktır. Bu eşitlikle hesaplanan ağırlıkların 1'den küçük olacağı kesindir. Bu ağırlıkların en büyük değere sabit olanı 1'den çıkarılarak i köşesinin eşi olmaması olasılığı eşitlik 3.2'deki gibi hesaplanır.

$$p_i^0(l^*) = 1 - \max_{l \neq l^*} (w_i(l)) \quad (3.2)$$

Eşi olmaması olasılığının ilk değerleri hesaplanan köşelerin l uzaklıklarındaki eş adayları ile eş olması olasılıklarının ilk değerleri ise eşitlik 3.3'deki gibi hesaplanır ve her bir i köşesi için bu olasılıkları içeren bir L_i kümesi oluşturulur. Böylece her bir i köşesinin eşi olmaması olasılığının ilk değeri ile i köşesinin l uzaklıklarındaki eş adaylarıyla eşleşmesi olasılıklarının ilk değerleri toplamının 1 olması sağlanır.

$$p_i^0(l) = p_i(l|i) * (1 - p_i^0(l^*)) \quad (3.3)$$

Bu eşitlikteki $(1-p_i^0(l^*))$ değeri köşenin eş adaylarının herhangi biriyle eşleşebilme olasılığını ifade eder. Eşitlikte $p_i(l|i)$ değeri, i köşesinin eşleştiği varsayımına göre i köşesinin l eş adayıyla eşleşmesi olasılığını belirtmektedir. Bu değer 3.4 eşitliğindeki gibi 3.2 eşitliğinde hesaplanan ağırlık değerleri yardımıyla hesaplanır. Bu eşitlikte hesaplanan tüm ağırlık değerlerinin toplamı içerisinde her bir $w_i(l)$ ağırlığının oranı hesaplanır.

$$p_i(l|i) = \frac{w_i(l)}{\sum_{l' \neq l^*} w_i(l')} \quad (3.4)$$

Bu şekilde $G1$ 'deki tüm i köşelerinin $G2$ 'deki tüm l eş adayları ile eşleşme olasılıklarının ilk değerleri $(p_i^0(l))$ ve i köşelerinin $G2$ 'deki hiçbir köşeye eşleşmeme olasılığının ilk değeri $(p_i^0(l^*))$ hesaplanmış olur.

İkinci aşamada ise her i köşesi için hesaplanan bu ilk değerler belirli sayıda güncellenerek i köşelerinin varsa eşleri bulunur. Hesaplanan ilk olasılık değerlerinde

köşelerin komşuluklarındaki piksellerin köşe adaylarının komşuluklarındaki pikseller ile ne kadar benzer oldukları karşılaştırılırken; ikinci aşamada ise köşeler arasında bulunan tutarlılık ilkesinden yararlanır. Buna göre G1'deki her köşenin G2'deki eş köşesiyle arasındaki uzaklık ile G1'deki köşelere yakın olan diğer köşelerin G2'deki eşleriyle aralarında bulunan uzaklıklar arasında bir tutarlılık olması gerekir. Buradaki yakınlık, i ve j G1'de birbirine yakınlığı tartışılan iki köşe olmak üzere, köşelerin düşey ve yataydaki konumları arasındaki farklardan büyük olanının eşitlik 3.5'te gösterildiği gibi belirli bir R değerinden küçük olması durumudur. Bu R değeri çalışmada 50 piksel belirlenmiştir.

$$\max(|x_i - x_j|, |y_i - y_j|) \leq R \quad (3.5)$$

Yukarıdaki kısıtlara göre, G1'de birbirine yakın olan iki köşe ile bunların herhangi iki eş adayı arasındaki yatay ve dikey mesafelerden büyük olanı arasındaki fark belirli bir eşik değerinin altındaysa bu iki köşe ve iki eş adayı arasında tutarlılıktan söz edilebilir. Bu eşik değeri çalışmada 2 piksel alınmıştır ve tutarlılık kısıtlaması eşitlik 3.6'da ifade edilmektedir. Eşitlikte l ve l' değerleri birbirine yakın olan iki köşeyi ifade etmektedir.

$$\|l - l'\| = \max(|l_x - l'_x|, |l_y - l'_y|) \leq 2 = \Theta \quad (3.6)$$

Buna göre G1'de birbirine yakın olan iki köşenin G2'de tutarlı olan iki eş adayı varsa bu adaylar birbirlerinin G1'deki eşleriyle eşleşme ihtimallerini arttırmaları. Bunun için her bir güncellemede G1'deki her köşenin G2'deki her eş adayı ile G1'deki köşeye yakın olan köşelerin G2'deki eş adayları arasında bir tutarlılık olup olmadığına bakılır. Bu durum eşitlik 3.7'de ifade edilmiştir. Eşitlikteki $q_i^k(l)$ değeri k 'nıncı güncellemede i köşesinin l eş adayı ile eşleşmesi olasılığının artmasını sağlayan değerdir. Ayrıca k_i ve k_j G1'deki i 'ninci ve j 'ninci köşeleri ifade etmektedir.

$$q_i^k(l) = \sum_{\substack{j \in \\ k_i \text{ yakin } k_j \\ j \neq i}} \left[\sum_{\substack{l' \in \\ \|l-l'\| < \Theta}} p_j^k(l') \right], \quad l \neq l^*. \quad (3.7)$$

3.7 eşitliğine göre hesaplanan $q_i^k(l)$ değerleri ancak G_1 'deki i köşelerinin l eş adayları ile tutarlı olan hiçbir eş adayı bulunmaması durumunda 0 olurlar. Bu değerler yardımıyla, her i köşesinin her l eş adayı ile eşleşme olasılıkları ile hiçbir eş adayı ile eşleşmeme olasılıklarının normalize edilmemiş halleri 3.8 ve 3.9 eşitliklerindeki gibi güncellenir.

$$\hat{p}_i^{k+1}(l) = p_i^k(l) * (A + B * q_i^k(l)), \quad l \neq l^* \quad (3.8)$$

$$\hat{p}_i^{k+1}(l^*) = \hat{p}_i^k(l^*) \quad (3.9)$$

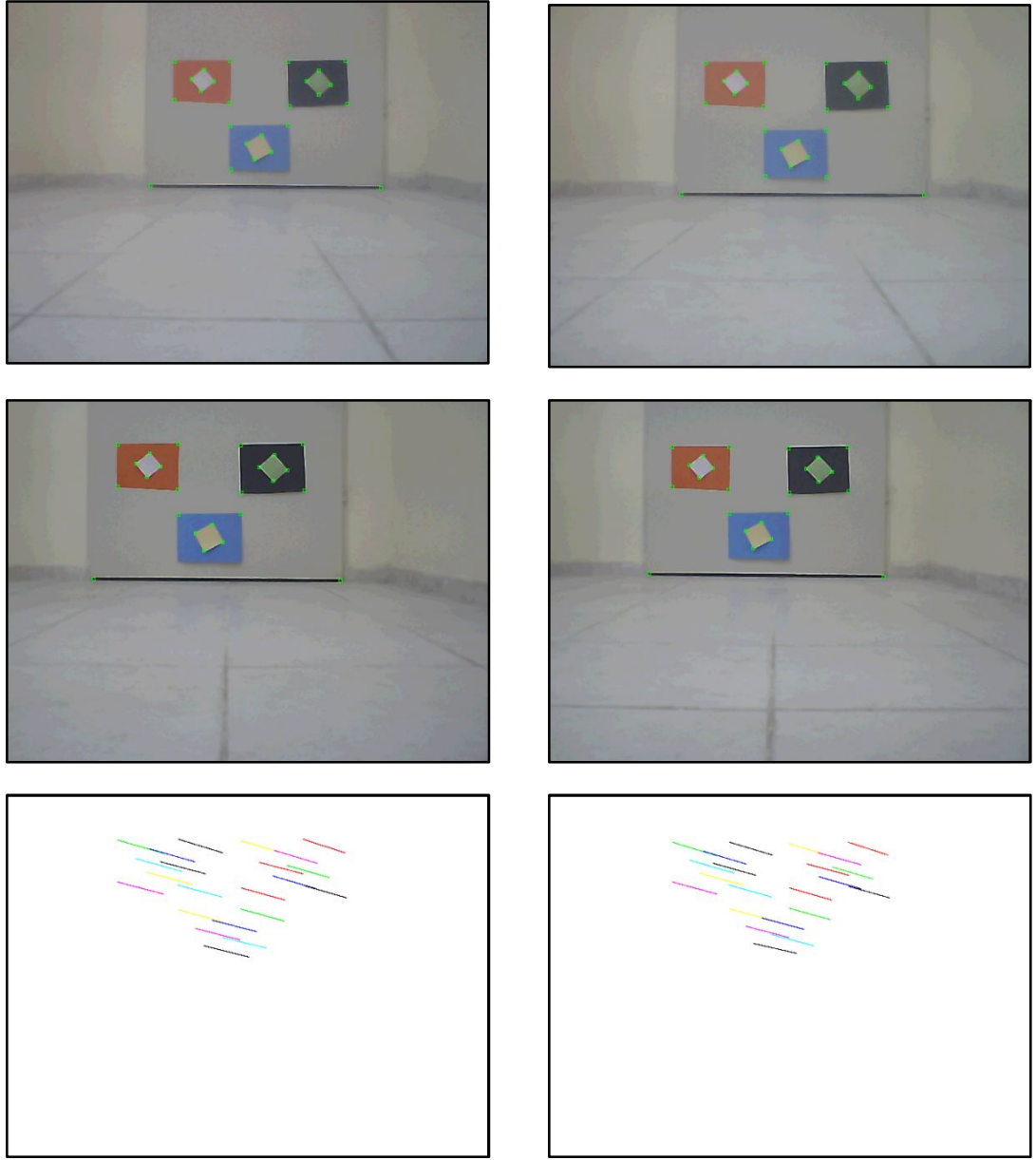
A sabiti bir eş adayının tutarlı hiçbir eş adayına sahip olmaması durumunda eş adayının eşleşme olasılığının 0'a yaklaşma hızını belirlerken B sabiti ise tutarlı eş adaylarına sahip bir eş adayının eşleşme olasılığının 1'e yaklaşma hızını belirler. Çalışmada A sabiti 0,6; B sabiti ise 25 olarak belirlenmiştir. Her bir i köşesinin eş adaylarının eşleşme olasılıklarını ve köşenin eşi olmaması olasılığını içiren L_i kümesindeki değerler eşitlik 3.10'daki gibi normalize edilerek bir L_i kümesindeki olasılıkların toplamının 1 olması sağlanır.

$$p_i^{k+1}(l) = \frac{\hat{p}_i^{k+1}(l)}{\sum_{l' \in L_i} \hat{p}_i^{k+1}(l')} \quad (3.10)$$

Böylece eşitlik 3.9'da bir önceki değeri korunan köşelerin hiç eşi olmama olasılıkları eşitlik 3.10'da normalleştirilerek güncellenmiş olur. Bu güncellemeler sırasında eş olma olasılığı 0,7'yi aşan köşe adayları eşleşmiş kabul edilir ve eş olma olasılıkları 1'e tamamlanarak diğer eşleşmelerdeki eşleşme hızı arttırılmış olur.

3.3. Eşleşme Algoritmasının Benzetim Sonuçları

Algoritmada kullanılan sabitleri tespit etmek için robotta bulunana stereo kameralarla alınan görüntüler XBee modülü yardımıyla bilgisayardaki benzetim ortamına kablosuz olarak aktarılmıştır. Şekil 3.2'de görüntüdeki kapıya 242 ve 256cm uzaklıktan alınan sağ ve sol kamera görüntülerinde tespit edilen köşeler ve bu köşelerle oluşturulan fark haritası gösterilmektedir. Fark haritasında sol görüntüdeki köşelerin sağ görüntüdekileri arasına farklı renklerde eşleştirici çizgiler çizilmiştir. 242cm uzaklıktan alınan test görüntülerinde tespit edilmesi gereken 26 köşeden sol görüntüde 25 tanesi tespit edilirken sağ görüntüde 24 tanesi tespit edilmiştir. Bu köşelerden 22 tanesi stereo eşleştirme algoritması ile başarılı bir şekilde eşleştirilmiştir. 256cm uzaklıktan alınan test görüntülerinde ise tespit edilmesi gereken 26 köşeden sol görüntüde 26 tanesi sağ görüntü de ise 24 tanesi tespit edilmiştir. Bu köşelerden 22 tanesi 242cm uzaklıktaki görüntülerde olduğu gibi başarılı bir şekilde eşleştirilmiştir.

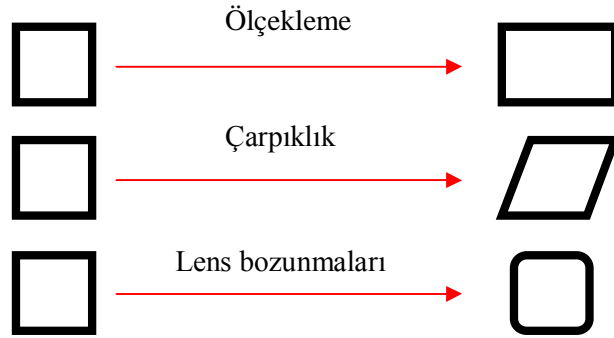


Şekil 3.2. 242 ve 256cm uzaklıktan alınan görüntülerde tespit edilen köşeler ve eşleştirme sonucu oluşturular fark haritası.(sol üst 242cm sol kamera, sol orta 242cm sağ kamera, sol alt 242cm fark haritası, sağ üst 256cm sol kamera, sağ orta 256cm sağ kamera, sağ alt 256cm fark haritası)

3.4.3 Boyutta Konum Çıkarımı

Stereo eşleme sonrası eşleştirilen köşelerin 3 boyutta robota göre konumları tespit edilerek Genişletilmiş Kalman Filtresinin ölçüm aşamasında kullanılmaktadır. Stereo eşlemede kullanılan algoritma için kameraların birbirlerine göre konum ve yönleri, lens bozunmaları gibi bilgilere ihtiyaç duyulmadığı önceden belirtilmişti. Ancak köşelerin 3 boyuttaki konumlarının küçük hatalarla tespit edilebilmesi için bu parametrelerin gerçek değerlerinin bilinmesi gerekmektedir. Bundan dolayı 3 boyutta konum çıkarımı aşamasında hem kameraların iç ölçümleme parametrelerine hem de kameraların birbirlerine göre konumu ve yönünü içeren dış ölçümleme parametrelerine ihtiyaç vardır. Bu çalışmada Jean-Yves Bouguet [19] tarafından geliştirilen ölçümleme kütüphanesi kullanılarak hem kameraların iç ölçümleme parametreleri hem de dış ölçümleme parametreleri elde edilmiştir.

Temel olarak 5 grup iç ölçümleme parametresi vardır. Bunlar kameranın izdüşüm merkezinin konumu, kameranın odak uzaklığı, ölçekleme değişkeni, çarpıklık değişkeni ve lens bozunmalarıdır. Kameranın yatay ve dikeydeki çözünürlüğü x ve y olarak belirtilirse, çoğu kamera sisteminin izdüşüm merkezinin konumu $(x/2, y/2)$ konumunda yer almaz. Bu konumun doğru değeri ile birlikte kameranın odak uzaklığı, 3 boyutta konum çıkarımında başarımı etkileyen en önemli iki faktördür.



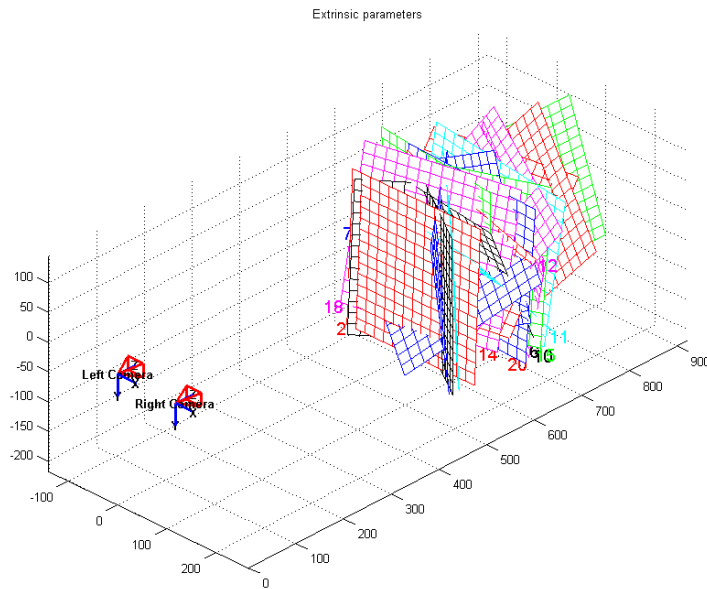
Şekil 3.3. Piksellerdeki ölçekleme ve çarpıklık hatalarından ve lens bozunmalarından kaynaklanan bozunmalar

Şekil 3.3'te gerçekte kare şeklinde olan soldaki dörtgenlerin piksellerdeki ölçekleme ve çarpıklık hatalarından ve lens bozunmalarından kaynaklanan hatalar sonucu

alabilecekleri şekiller sağda gösterilmiştir. Ölçekleme değişkeni görüntü sensöründeki piksellerin yatay ve dikeydeki büyüklükleri arasındaki oranı göstermektedir. Çarpıklık değişkeni piksellerin yatay ve dikey doğrultuları arasındaki açı farkına göre değişmektedir. Lens bozunmaları ise lensin oluşturulduğu materyallerin sebep olduğu görüntü kaymalarıdır. Kalibrasyon algoritması tarafından ölçekleme değişkeninin ve odak uzaklığının ayrı ayrı kestirilmesi zor olduğundan, tek odak uzaklığı ve ölçekleme değişkeni yerine, ölçekleme değişkenine göre oranlanmış iki odak uzaklığı sağlanmaktadır.

Dış ölçümleme parametreleri ise iki tanedir. Bunlar kameralardaki odak doğrultularının paralel hale gelmesini sağlayan R rotasyon vektörü ve kameraların izdüşüm merkezleri arasındaki mesafeyi belirten T konum farkı vektörüdür.

Tüm iç ve dış ölçümleme parametreleri Jean-Yves Bouguet tarafından oluşturulan program tarafından sağlanabilmektedir. Bunun için sağ ve sol kameralar sabitlenip bir dama tahtasının 15-20 görüntüsü kaydedilir ve görüntüler üzerindeki köşelerden sağ üst, sağ alt, sol üst ve sol altta olanlar işaretlenir. Program tüm görüntüleri kullanarak tekrarlayıcı algoritmalarla iç ve dış ölçümleme parametrelerini kullanıcıya sunar. Şekil 3.4'te bu program ile ölçümleme parametreleri bulurken kullanılan dama tahtası görüntülerinin program tarafından hesaplanan konumları gösterilmiştir.



Şekil 3.4. Kalibrasyon görüntülerinin 3 boyuttaki konumları(eksenlerin birimi mm)

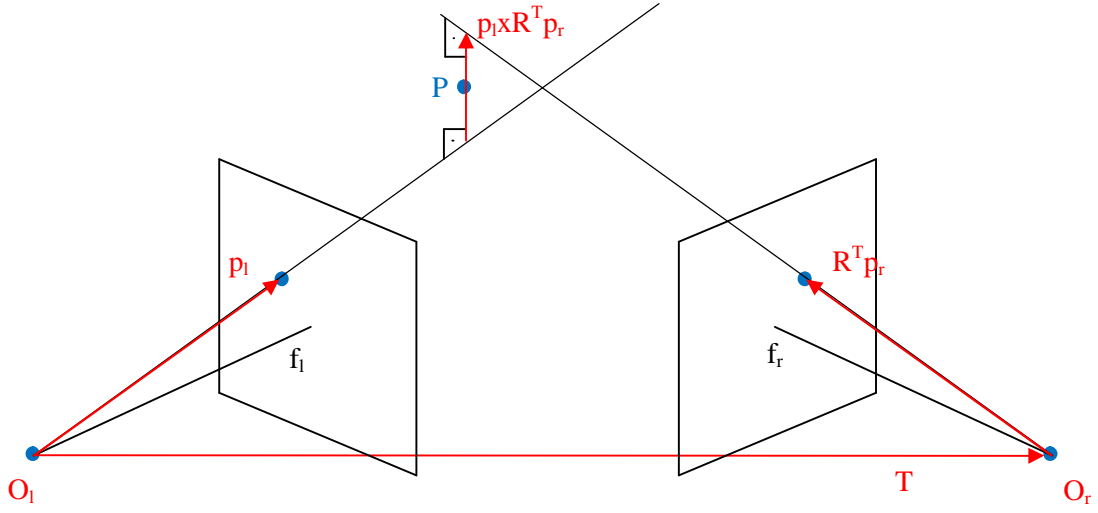
Çizelge 3.1.'de elde edilen ölçümleme sonuçları gösterilmiştir.

Çizelge 3.1. Ölçümleme sonuçları

İç Ölçümleme Parametreleri		
	Sol	Sağ
Odak Uzaklığının Yatay Piksel Büyüklüğü Cinsinden Değeri	765,72805	772,77627
Odak Uzaklığının Dikey Piksel Büyüklüğü Cinsinden Değeri	767,65633	774,91315
İzdüşüm Merkezinin Konumu X	319,57713	306,84305
İzdüşüm Merkezinin Konumu Y	192,06768	195,22386
Çarpıklık Katsayısı	0	0
Lens Bozunması k1	-0,12995	-0,12112
Lens Bozunması k2	0,08416	0,14237
Lens Bozunması k3	0,00208	0,00167
Lens Bozunması k4	0,00449	0,00133

Dış Ölçümleme Parametreleri			
	1	-0,0014	-0,0098
R	0,0012	0,9998	-0,0203
	0,0098	0,0203	0,9997
T	-119,02822	-2,57842	-6,2994

Stereo eşleme aşamasında eşleştirilmiş olan köşe noktalarının 3 boyutta robota göre konumları Trucco'nun [20] da bahsettiği gibi üçgenleyerek yeniden kurma yöntemiyle tespit edilmiştir. Bu yöntemde kameralardan biri referans alınır ve 3 boyutta tespit edilen köşenin konumu bu kameraya göre hesaplanır. Bu çalışmada sol kamera referans alınmıştır. Amaç kameraların izdüşüm merkezleri ve köşe noktasının gerçekteki konumu arasında vektörlerle üçgenimsi bir yapı oluşturarak, gerçek konumları bilinen kameralarla konumu bilinmeyen köşe noktasının konumunu tespit etmektir.



Şekil 3.5. Üçgenleyerek yeniden kurma yöntemi

Şekil 3.5'te gösterildiği gibi her bir köşe noktasının konumunun tespitinde 4 vektörden yararlanır. Görüntü aslında izdüşüm merkezlerini arkasında ve ters bir şekilde oluşmasına rağmen çoğu kamera sisteminin tanımlanmasında yapıldığı gibi bu şekilde de görüntünün izdüşüm merkezinin önünde ve düz bir şekilde olduğu varsayılmıştır. Köşe noktalarının konumlarının tespitinde kullanılan vektörlerden birincisi kameralar arasındaki mesafeyi belirten ve ölçüleme aşamasında hesaplanan T vektörüdür. İkinci vektör referans alınan sol kameranın izdüşüm merkezi ile tespit edilmeye çalışılan köşenin görüntü üzerindeki düzeltilmiş konumu arasında çizilen p_1 vektörüdür. Köşenin görüntü üzerindeki konumunun düzeltilmesi aşamasından daha sonra bahsedilecektir. Üçüncü vektör sağ kameranın izdüşüm merkezi ile köşenin sağ görüntü üzerindeki düzeltilmiş konumu arasında çizilen p_r vektörünün rotasyon matrisi ile çarpılarak döndürülmüş şekli olan $R^T p_r$ vektörüdür. Burada dış ölçüleme parametrelerinden olan R matrisinin kullanılmasının sebebi farklı eksnelere göre tanımlanan p_1 ve p_r vektörlerini aynı eksen sistemi üzerinde tanımlamaktır. Bunun için p_r vektörü döndürülerek p_1 vektörünün tanımlanmasında kullanılan eksnelere göre tanımlanmış hale dönüştürülür. Şekildeki konumu tespit edilecek olan P noktasının üzerinden geçecekmiş gibi ilerleyen p_1 ve $R^T p_r$ gerçekte bu noktanın koordinatlarından hatasız bir şekilde geçemeyebilirler. Bunun hatayı engellemek için bu iki vektöre de dik olan ve bu vektörlerin çapraz çarpımlarıyla

oluşturulmuş olan $p_l \times R^T p_r$ vektörüne ihtiyaç duyulur. Böylece tüm bu vektörler yardımıyla eşitlik 3.11 oluşturulabilir.

$$ap_l - bR^T p_r + c(p_l \times R^T p_r) = T \quad (3.11)$$

Eşitlikteki a,b ve c katsayıları vektörlerin üçgenimsi şekli tamamlaması için çarpılması gereken katsayıları göstermektedir. Üç bilinmeyene sahip denklem çözülerek a,b ve c katsayıları elde edilir. Köşenin 3 boyuttaki konumu ise eşitlik 3.12'deki gibi bulunur.

$$P = ap_l + c(p_l \times R^T p_r) / 2 \quad (3.12)$$

Üçgenleyerek yeniden kurma yöntemi ile köşelerin konumunun az hatayla tespit edilebilmesi için tespit edilen köşelerin konumlarının kameraların iç ölçümleme parametreleri ile düzeltilmesi gerekmektedir. Bunun için ilk önce, iki kamera ile tespit edilen köşelerin kameraların izdüşüm merkezlerine göre yatayda ve dikeydeki konumları bulunur. Bu konumlar kalibrasyon sonuçlarında yatay ve dikey piksel büyüklüklerine göre hesaplanan odak uzaklıklarına bölünerek bu konumların yatayda ve dikeyde kaç odak uzaklığı büyüklüğünde olduğu (I'_x , I'_y) bulunur. I_x ve I_y görüntülerde tespit edilen köşelerin düzeltilmemiş yatay ve dikey konumlarını, O_x ve O_y kameraların izdüşüm merkezlerinin yatay ve dikey konumlarını ve f_x ve f_y kameraların odak uzaklıklarının yatay ve dikey piksel büyüklükleri cinsinden değerini belirtmek üzere yukarıda uygulanan işlemler eşitlik 3.13 ve 3.14'te gösterilmektedir.

$$I'_x = (I_x - O_x) / f_x \quad (3.13)$$

$$I'_y = (I_y - O_y) / f_y \quad (3.14)$$

Daha sonra I'_x değeri iç ölçüleme değişkenlerinden çarpıklık parametresine(çp) göre güncellenir. Çalışmadaki çp 0 olarak elde edildiğinden bu işlem hesaplamalarda bir güncelleme sağlamamaktadır.

$$I'_x = I'_x - (\text{çp} * I'_y) \quad (3.15)$$

Son olarak lens bozunmalarından dolayı oluşan hatalar giderilmektedir. Hala hata içeren köşe konumlarından(I'_x, I'_y) düzeltilmiş köşe konumlarına(D_x, D_y) eşitlik 3.16 yardımıyla eşitlik 3.17 ve 3.18'deki gibi erişilir.

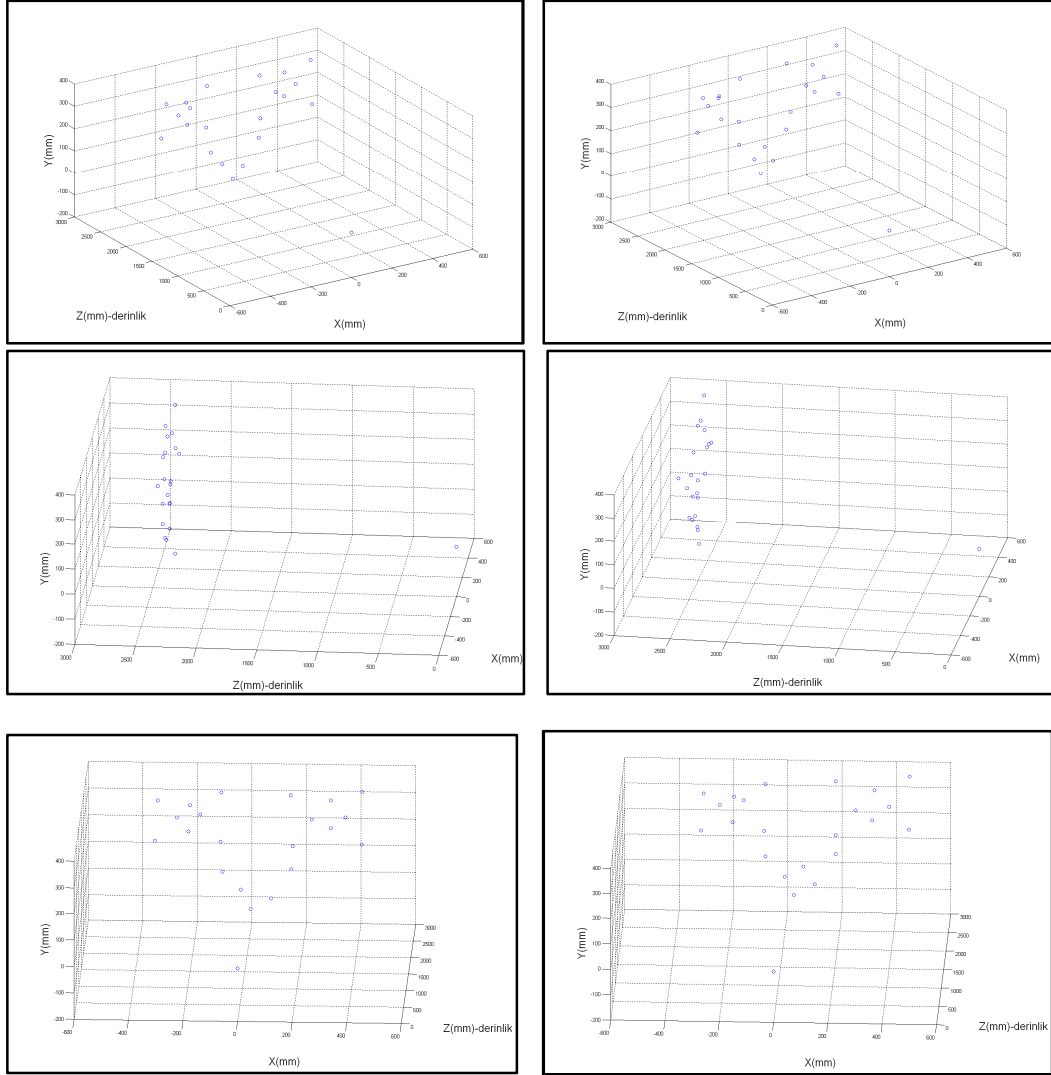
$$r^2 = I_x'^2 + I_y'^2 \quad (3.16)$$

$$I'_x = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) D_x + 2k_3 I_x I_y + k_4 (r^2 + 2I_x^2) \quad (3.17)$$

$$I'_y = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) D_y + 2k_4 I_x I_y + k_3 (r^2 + 2I_y^2) \quad (3.18)$$

3.5. 3 Boyutta Konum Çıkarımının Benzetim Sonuçları

Şekil 3.2.'de verilen 242cm ve 256cm uzaklıktan alınmış görüntüler için 3 boyutta konum çıkarımı sonunda elde edilen benzetim sonuçları şekil 3.6.'da verilmiştir.



Şekil 3.6. 3 Boyutta robota göre konumları çıkarılan köşe noktaları
(242cm soldaki görüntüler, 256cm sağdaki görüntüler)

4. EŞ ZAMANLI KONUMLAMA VE HARİTA ÇIKARIMI

Özellik noktaları için yapılan ölçümler ve robotun enkoderlarından alınan konum değişimi bilgileri kesin doğrulukta değildir. Bu verilerin dağılımları incelendiğinde dağılımların Gaussian dağılımıyla uyumlu olduğu gözlemlenmiştir. Dolayısıyla robotun konumunun tespitinde ve harita oluşturmasında enkoder bilgisindeki ve köşelerin 3 boyuttaki konumlarında bulunan hataları göz önüne alan ve Gaussian dağılımlarında çalışan olasılıksal bir algoritma kullanılmıştır.

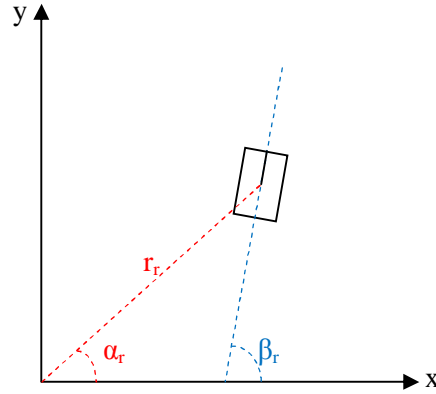
4.1. Genişletilmiş Kalman Filtresi

Bu çalışmada EZKH gerçekleştirmek için lineer Gaussian sistemlerde çalışabilen Kalman filtresinin lineer olmayan sistemler için geliştirilmiş şekli olan Genişletilmiş Kalman Filtresi kullanılmıştır. Sistemde robotun konumu ve etrafındaki köşe noktaların oluşturacağı harita bilgisi ortamda var olduğu kabul edilen bir orijin noktasına göre tanımlanmakta ve robot bu merkez etrafında serbestçe gezinebilmektedir. Dolayısıyla hem robotun yapacağı ölçümler robotun konumuna, hem de robotun bir sonraki konumu robotun bir önceki konumuna lineer bir bağımlılık göstermediklerinden bu problemin çözümünde GKF kullanılmıştır. GKF’de ve KF’de sistemin durum parametreleri hakkındaki kanı Gaussian dağılımındadır. Filtrelerde her bir durum parametresinin ortalama değeri bir vektör(μ) halinde tutulur. Ayrıca bu ortalama değerlerin belirsizliklerini tutan ve parametrelerin belirsizliklerinin birbirlerini hangi oranda etkileyeceğini belirleyen bir kovaryans matrisi(Σ) sayesinde sistemdeki parametrelerin kanıları güncellenir. GKF hareket güncellemesi ve ölçüm güncellemesi olmak üzere iki aşamadan oluşur. Hareket güncellemesinde robottaki hareket değişimini ölçen enkoderlardan elde edilen konum değişimi bilgisi kullanılır. Ölçüm güncellemesinde ise stereo kameralarla tespit edilen köşe noktalarının koordinatları kullanılır. Filtredeki bilgilerin tutulduğu μ vektöründe ve Σ matrisinde bulunan robot bilgileri hem hareket hem de ölçüm güncellemesinden etkilenirken; harita bilgileri ise sadece ölçüm güncellemesinden etkilenir.

4.2. Hareket Güncellemesi

Hareket güncellemesi için iki temel yaklaşım kullanılabilir [1]. Birinci yaklaşım enkoderlarından gelen bilgiye göre robotun yer değiştirme ve dönme hızlarının ölçülmesine dayanan yaklaşımdır. Bu yaklaşımda elde edilen hız bilgileri tutulan zaman bilgisiyle beraber yer değişimini vermektedir. İkinci yaklaşım doğrudan enkoderlardan gelen konum bilgisinin kullanılmasına yönelik yaklaşımdır. İkinci yaklaşım hız çıkarımı için enkoder bilgisi üzerinde herhangi bir ortalama alma işlemi gerçekleştirmediğinden ilk yönteme göre daha başarılıdır. Bu çalışmada konum bilgisini direk kullanan bir hareket güncellemesi kullanılmıştır.

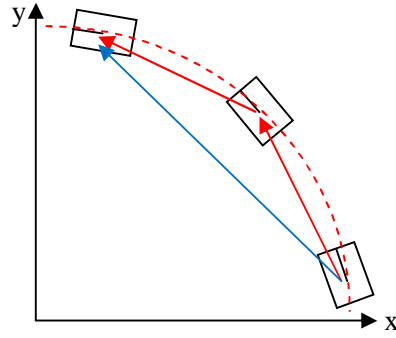
Hareket güncellemesi ile sadece robota ait veriler güncelleneceğinden oluşturulan sistemde robotun o anki konum bilgisi, robotun bir önceki konum bilgisi ve o anki hareket eylem bilgisi kullanılarak ifade edilmelidir. Oluşturulan sistemde robotun 2 boyuttaki konumunu ifade etmek için 3 değişkenden yararlanılmıştır. Şekil 4.1.'de gösterilen bu değişkenlerden r_r robotun konumu ile robotun dünyasındaki orijin noktasını birleştiren doğru parçasının uzunluğu, r_a bu doğru parçasının robotun dünyasındaki x eksenine yaptığı açı ve r_b da robotun doğrultusunun robotun dünyasındaki x eksenine yaptığı açı olarak tanımlanır.



Şekil 4.1. Robotun konumunun tanımlanması

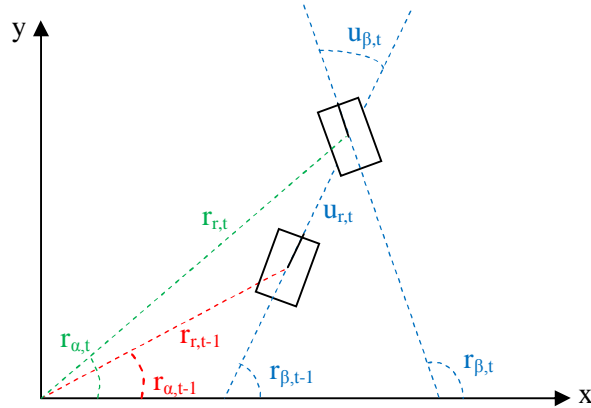
Robotun düz ilerlerken sonsuz yarıçaplı bir çember üzerinde ilerlediği, dönerek ilerlerken de yarıçapı belirli bir çember üzerinde ilerlediği varsayımıyla FPGA üzerinde enkoderlardan gelen bilgileri işleyen bir modül oluşturulmuştur. Bu modül robotun üzerinde ilerlediği çemberi şekil 4.2.'de kırmızı oklarla belirtildiği gibi

hareket ve hareketi takip eden dönme hareketleriyle kestirmektedir. EZKH modülü enkoder modülünden konum değişimi bilgisini talep ettiğinde, enkoder modülü kestirimler sonucu elde ettiği konum değişimini iki değişken olarak EZKH modülüne aktarır. Şekilde de mavi okla gösterilen bu değişim, konum değişimi u_r ve doğrultu değişimi u_β şeklinde aktarılmaktadır.



Şekil 4.2. Robotun konum değişiminin kestirilmesi

Buna göre, robotun herhangi bir t anındaki $r_{r,t}$, $r_{a,t}$ ve $r_{\beta,t}$ bileşenlerinden oluşan konumu, $(t-1)$ anındaki konumunu oluşturan $r_{r,t-1}$, $r_{a,t-1}$ ve $r_{\beta,t-1}$ bileşenleri ve t anındaki konum değişimi bilgisini oluşturan $u_{r,t}$ ve $u_{\beta,t}$ bileşenleriyle ifade edilebilir. Şekil 4.3'te gösterilen bu durum, geometrik özellikler kullanılarak 4.1-4.5 eşitlikleri arasında ifade edilmiştir.



Şekil 4.3. Robotun konum değişiminin kestirilmesi

$$T_1 = (r_{r,t-1} \cos(r_{\alpha,t-1}) + u_{r,t} \cos(r_{\beta,t-1}))^2 \quad (4.1)$$

$$T_2 = (r_{r,t-1} \sin(r_{\alpha,t-1}) + u_{r,t} \sin(r_{\beta,t-1}))^2 \quad (4.2)$$

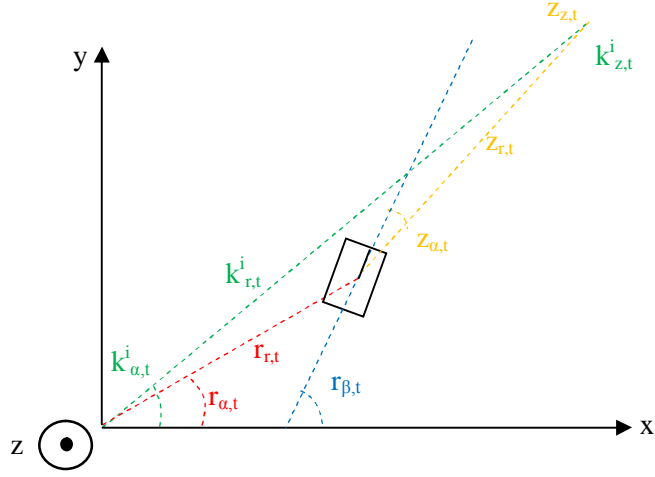
$$r_{r,t} = \sqrt{T_1 + T_2} \quad (4.3)$$

$$r_{\alpha,t} = a \tan 2 \left(\frac{r_{r,t-1} \sin(r_{\alpha,t-1}) + u_{r,t} \sin(r_{\beta,t-1})}{r_{r,t-1} \cos(r_{\alpha,t-1}) + u_{r,t} \cos(r_{\beta,t-1})} \right) \quad (4.4)$$

$$r_{\beta,t} = r_{\beta,t-1} + u_{\beta,t} \quad (4.5)$$

4.3. Ölçüm Güncellemesi

Önceki bölümlerde oluşturulan algoritmalar sonucu köşelerin robota göre 3 boyuttaki konumları bulunmuştu. Yapılan ölçüm sonucu elde edilen köşe koordinatlarının, robotun o anda sahip olduğu haritadaki köşelerin koordinatlarıyla aynı olup olmadığını kıyaslayabilmek gerekmektedir. Bunun için robot tarafından gözlemlenen tüm köşelerin robotun haritasındaki eksnlere ve haritanın orijinine göre konumları haritaya kaydedilmektedir. Bu konumlar 3 değişkenle ifade edilir. Bir i köşesi için bunlar şöyledir: köşeyi haritanın z eksenine bağlayan doğru parçasının uzunluğu olan $k_{r,t}^i$, bu doğru parçasın $z=0$ düzlemi üzerindeki izdüşümünün x eksenini ile yaptığı açı $k_{\alpha,t}^i$ ve köşenin yüksekliğini belirten köşenin $k_{z,t}^i$ koordinatı.



Şekil 4.4. Robotun konum değişiminin kestirilmesi

Bu değerleri tanımlayabilmek için robotun konum parametreleri olan $r_{r,t}$, $r_{\alpha,t}$ ve $r_{\beta,t}$ ile birlikte algılanan köşenin robota göre konumunu belirten $z_{r,t}$, $z_{\alpha,t}$ ve $z_{z,t}$ parametrelerine ihtiyaç duyulur. Köşenin robota göre konumunu belirten parametrelerden $z_{r,t}$ köşenin $z=0$ düzlemi üzerindeki izdüşümünü robota bağlayan doğru parçasının uzunluğunu, $z_{\alpha,t}$ bu doğru parçası ile robotun doğrultusu ile arasındaki açıyı ve $z_{z,t}$ de köşenin robot tarafından ölçülen yüksekliğini belirtmektedir. Bu parametrelerin tümü şekil 4.4.'te gösterilmiştir. Ölçüm güncellemesi sırasında yapılan ölçümün daha önce tespit edilen köşelerden birine ait olup olmadığını anlamak için 4.6-4.10 eşitliklerinde gösterildiği gibi $\underline{z}_{r,t}$, $\underline{z}_{\alpha,t}$ ve $\underline{z}_{z,t}$ parametreleri hesaplanarak gerçek ölçüm parametreleri olan $z_{r,t}$, $z_{\alpha,t}$ ve $z_{z,t}$ ile kıyaslanır.

$$T_3 = (k_{r,t}^i \cos(k_{\alpha,t}^i) - r_{r,t} \cos(r_{\alpha,t}))^2 \quad (4.6)$$

$$T_4 = (k_{r,t}^i \sin(k_{\alpha,t}^i) - r_{r,t} \sin(r_{\alpha,t}))^2 \quad (4.7)$$

$$\underline{z}_{r,t} = \sqrt{T_3 + T_4} \quad (4.8)$$

$$\underline{z}_{\alpha,t} = a \tan 2 \left(\frac{k_{r,t}^i \sin(k_{\alpha,t}^i) - r_{r,t} \sin(r_{\alpha,t})}{k_{r,t}^i \cos(k_{\alpha,t}^i) - r_{r,t} \cos(r_{\alpha,t})} \right) - r_{\beta,t} \quad (4.9)$$

$$\underline{z}_{z,t} = k_{z,t}^i \quad (4.10)$$

4.4. Algoritma

Sistemde, μ durum vektörünün ilk 3 elemanında robotun konum parametreleri olan $r_{r,t}$, $r_{\alpha,t}$ ve $r_{\beta,t}$ tutulur. Robot köşe noktaları gözlemledikçe, her yeni köşe noktasının $k_{r,t}^i$, $k_{\alpha,t}^i$ ve $k_{z,t}^i$ parametreleri durum vektörüne eklenir. Bu çalışmada 100 farklı köşeyi tutabilecek büyüklükte olan 303×1 büyüklüğünde bir durum vektörü kullanılmıştır ve bu vektör eşitlik 4.11’de gösterilmiştir. 303×303 büyüklüğünde olan Σ matrisinin ise köşegen üzerindeki elemanları durum vektöründe tutulan bilgilerin belirsizliklerini belirtirken köşegen üzerinde bulunmayan elemanları ise durum vektöründeki iki parametre arasında bulunan etkileşimi belirtmektedir.

$$\mu_t = \left[r_{r,t} \ r_{\alpha,t} \ r_{\beta,t} \ k_{r,t}^1 \ k_{\alpha,t}^1 \ k_{z,t}^1 \ k_{r,t}^2 \ k_{\alpha,t}^2 \ k_{z,t}^2 \ \dots \right]^T \quad (4.11)$$

Robot çalışmaya başlayacağı anda ortamdaki konumu ve ortamın haritası hakkında hiçbir bilgiye sahip olmadığından μ vektörünün tüm elemanlarının ilk değerleri eşitlik 4.12’deki gibi 0 olarak atanarak robot başlangıç anında oluşturulacağı haritanın orijinine yerleştirilmiş olunur.

$$\mu_0 = \left[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \dots \right]^T \quad (4.12)$$

Robotun başlangıç konumu orijin noktası olarak belirlendiğinden ve bu bilgi %100 doğru olduğundan Σ matrisinin robotun belirsizliğini tutan elemanlarının ilk değerleri 0 olarak atanır. Robotun haritası ilk durumda boş olduğundan Σ matrisinin köşe noktalarının belirsizliklerini tutan değerleri pratikte çok büyük bir değer teorikte ∞ olarak atanır.

$$\Sigma_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & \infty & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & . & . & \infty \end{bmatrix} \quad (4.13)$$

Eşitlik 4.14'te tanımlanan 3x303 büyüklüğündeki F_x matrisi kullanılarak ölçüm güncellemesine ait daha sonraki eşitliklerde sadece robota ait bilgilerin güncellenmesi sağlanır.

$$F_x = \begin{bmatrix} 1 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & 0 & . & . & . & 0 \end{bmatrix} \quad (4.14)$$

Sistem enkoderlardan konum değişimi bilgisini her aldığı anda tutulan μ vektörü ve Σ matrisi güncellenir. 4.3-4.5 eşitlikleri yardımıyla μ vektörü eşitlik 4.15'deki gibi güncellenir.

$$\mu_t = F_x^T \left[a \tan 2 \left(\frac{\sqrt{T_1 + T_2} \left(\frac{r_{r,t-1} \sin(r_{\alpha,t-1}) + u_{r,t} \sin(r_{\beta,t-1})}{r_{r,t-1} \cos(r_{\alpha,t-1}) + u_{r,t} \cos(r_{\beta,t-1})} \right)}{r_{\beta,t-1} + u_{\beta,t}} \right) \right] \quad (4.15)$$

Σ matrisi güncellenirken ise G_t , V_t ve M_t matrislerine ihtiyaç duyulmaktadır. Bu matrislerden G_t 4.3-4.5 eşitliklerinin robotun bir önceki konumunun parametreleri olan $r_{r,t-1}$, $r_{\alpha,t-1}$ ve $r_{\beta,t-1}$ 'e göre kısmi türevlerinin eşitlik 4.16'daki gibi alınmasıyla oluşturulmuştur. G_t matrisindeki elemanların 4.17-4.25 eşitlikleri arasında verilmiştir.

$$G_t = \begin{pmatrix} \frac{\partial r_{r,t}}{\partial r_{r,t-1}} & \frac{\partial r_{r,t}}{\partial r_{\alpha,t-1}} & \frac{\partial r_{r,t}}{\partial r_{\beta,t-1}} \\ \frac{\partial r_{\alpha,t}}{\partial r_{r,t-1}} & \frac{\partial r_{\alpha,t}}{\partial r_{\alpha,t-1}} & \frac{\partial r_{\alpha,t}}{\partial r_{\beta,t-1}} \\ \frac{\partial r_{\beta,t}}{\partial r_{r,t-1}} & \frac{\partial r_{\beta,t}}{\partial r_{\alpha,t-1}} & \frac{\partial r_{\beta,t}}{\partial r_{\beta,t-1}} \end{pmatrix} \quad (4.16)$$

$$\frac{\partial r_{r,t}}{\partial r_{r,t-1}} = \frac{r_{r,t-1} + u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}{\sqrt{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}} \quad (4.17)$$

$$\frac{\partial r_{r,t}}{\partial r_{\alpha,t-1}} = \frac{r_{r,t-1}u_{r,t} \sin(r_{\beta,t-1} - r_{\alpha,t-1})}{\sqrt{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}} \quad (4.18)$$

$$\frac{\partial r_{r,t}}{\partial r_{\beta,t-1}} = \frac{r_{r,t-1}u_{r,t} \sin(r_{\alpha,t-1} - r_{\beta,t-1})}{\sqrt{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}} \quad (4.19)$$

$$\frac{\partial r_{\alpha,t}}{\partial r_{r,t-1}} = \frac{u_{r,t} \sin(r_{\alpha,t-1} - r_{\beta,t-1})}{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})} \quad (4.20)$$

$$\frac{\partial r_{\alpha,t}}{\partial r_{\alpha,t-1}} = \frac{r_{r,t-1}^2 + r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})} \quad (4.21)$$

$$\frac{\partial r_{\alpha,t}}{\partial r_{\beta,t-1}} = \frac{u_{r,t}^2 + r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})} \quad (4.22)$$

$$\frac{\partial r_{\beta,t}}{\partial r_{r,t-1}} = 0 \quad \frac{\partial r_{\beta,t}}{\partial r_{\alpha,t-1}} = 0 \quad \frac{\partial r_{\beta,t}}{\partial r_{\beta,t-1}} = 1 \quad (4.23-4.25)$$

Σ matrisi güncellemesinde kullanılan V_t matrisleri ise 4.3-4.5 eşitliklerinin t anındaki konum değişimleri olan $u_{r,t}$ ve $u_{\beta,t}$ parametrelerine göre eşitlik 4.26 'da gösterildiği gibi kısmi türevlerinin alınmasıyla elde edilir. V_t matrisindeki elemanların 4.27-4.32 eşitlikleri arasında verilmiştir.

$$V_t = \begin{pmatrix} \frac{\partial r_{r,t}}{\partial r_{r,t}} & \frac{\partial r_{r,t}}{\partial r_{\beta,t}} \\ \frac{\partial u_{r,t}}{\partial r_{r,t}} & \frac{\partial u_{\beta,t}}{\partial r_{\beta,t}} \\ \frac{\partial r_{\alpha,t}}{\partial r_{r,t}} & \frac{\partial r_{\alpha,t}}{\partial r_{\beta,t}} \\ \frac{\partial u_{r,t}}{\partial r_{r,t}} & \frac{\partial u_{\beta,t}}{\partial r_{\beta,t}} \\ \frac{\partial r_{\beta,t}}{\partial r_{r,t}} & \frac{\partial r_{\beta,t}}{\partial r_{\beta,t}} \\ \frac{\partial u_{r,t}}{\partial r_{r,t}} & \frac{\partial u_{\beta,t}}{\partial r_{\beta,t}} \end{pmatrix} \quad (4.26)$$

$$\frac{\partial r_{r,t}}{\partial u_{r,t}} = \frac{u_{r,t} + r_{r,t-1} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}{\sqrt{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})}} \quad (4.27)$$

$$\frac{\partial r_{r,t}}{\partial u_{\beta,t}} = 0 \quad (4.28)$$

$$\frac{\partial r_{\alpha,t}}{\partial u_{r,t}} = \frac{r_{r,t-1} \sin(r_{\beta,t-1} - r_{\alpha,t-1})}{r_{r,t-1}^2 + u_{r,t}^2 + 2r_{r,t-1}u_{r,t} \cos(r_{\alpha,t-1} - r_{\beta,t-1})} \quad (4.29)$$

$$\frac{\partial r_{\alpha,t}}{\partial u_{\beta,t}} = 0 \quad \frac{\partial r_{\beta,t}}{\partial u_{r,t}} = 0 \quad \frac{\partial r_{\beta,t}}{\partial u_{\beta,t}} = 1 \quad (4.30-4.32)$$

Σ matrisinin güncellenmesinde kullanılan son matris olan M_t ise 2x2 boyutlarında diyagonal bir matristir. 4.33 eşitliğinde gösterilen bu matris ile $u_{r,t}$ ve $u_{\beta,t}$ konum değişimi ölçümlerinin robotun konumundaki belirsizliği ne ölçüde arttırdığı belirlenir. Eşitlikteki k_1 , k_2 , k_3 ve k_4 katsayıları çalışmada sırasıyla 3×10^{-4} , 1×10^{-1} , 5×10^{-7} ve 1×10^{-2} olarak belirlenmiştir.

$$M_t = \begin{pmatrix} k_1 u_{r,t} + k_2 u_{\beta,t} & 0 \\ 0 & k_1 u_{r,t} + k_2 u_{\beta,t} \end{pmatrix} \quad (4.33)$$

Σ matrisi G_t , V_t ve M_t matrisleri kullanılarak her konum değişikliğinde eşitlik 4.34'deki gibi güncellenir. Bu işlem sonucunda robotun t-1 anındaki konumunun belirsizliği ile konum değişiminin kattığı belirsizlik birleştirilerek robotun ölçüm güncellemesi öncesi belirsizlik değerleri yenilenmiş olur.

$$\Sigma_t = G_t \Sigma_{t-1} G_t^T + F_x^T V_t M_t V_t^T F_x \quad (4.34)$$

Hareket güncellemesi tamamlandıktan sonra ölçüm güncellemesi gerçekleştirilmektedir. Bunun için 3 boyutta konum çıkarımı aşaması sonrası robota göre konumlar belirlenmiş olan her bir köşenin öncelikle daha önce gözlenip gözlenmediği tespit edilmektedir. Eğer köşe ilk defa gözleniyorsa μ vektöründe ve Σ matrisinde bu yeni köşe için yeni parametreler yerleştirilirken, köşe daha önce gözlemlenmiş bir köşeye 3 boyutta eşleşiyorsa, eşleşme sağlanan köşenin konum bilgileri yeni gözlemindeki konum bilgileriyle güncellenir. Bunun için gözlemlenen her bir j köşesi haritada o an bulunan her i köşesi ile kıyaslanır. Bunun için haritadaki her i köşesinin robotun o anki konumundan yapılacak gözlemler nasıl görüneceği hesaplanarak, bunların yapılan j gözlemi ile örtüşüp örtüşmediği kontrol edilir. Bu kontrol için robotun gözlemindeki, köşe noktalarının konumlarındaki ve robotun konumundaki belirsizlikler önemlidir. Hesaplama 4.8-4.10 eşitliklerinin robotun konumunu belirten $r_{r,t}$, $r_{\alpha,t}$ ve $r_{\beta,t}$ parametrelerine göre ve i köşe noktasının konumunu belirten $k_{r,t}^i$, $k_{\alpha,t}^i$ ve $k_{z,t}^i$ parametrelere göre kısmi türevlerine ihtiyaç vardır. Bu hesaplamalar 4.35-4.53 eşitlikleri arasında gösterilmiştir.

$$h_t^i = \begin{pmatrix} \frac{\partial z_{r,t}}{\partial r_{r,t}} & \frac{\partial z_{r,t}}{\partial r_{\alpha,t}} & \frac{\partial z_{r,t}}{\partial r_{\beta,t}} & \frac{\partial z_{r,t}}{\partial k_{r,t}^i} & \frac{\partial z_{r,t}}{\partial k_{\alpha,t}^i} & \frac{\partial z_{r,t}}{\partial k_{z,t}^i} \\ \frac{\partial z_{\alpha,t}}{\partial r_{r,t}} & \frac{\partial z_{\alpha,t}}{\partial r_{\alpha,t}} & \frac{\partial z_{\alpha,t}}{\partial r_{\beta,t}} & \frac{\partial z_{\alpha,t}}{\partial k_{r,t}^i} & \frac{\partial z_{\alpha,t}}{\partial k_{\alpha,t}^i} & \frac{\partial z_{\alpha,t}}{\partial k_{z,t}^i} \\ \frac{\partial z_{z,t}}{\partial r_{r,t}} & \frac{\partial z_{z,t}}{\partial r_{\alpha,t}} & \frac{\partial z_{z,t}}{\partial r_{\beta,t}} & \frac{\partial z_{z,t}}{\partial k_{r,t}^i} & \frac{\partial z_{z,t}}{\partial k_{\alpha,t}^i} & \frac{\partial z_{z,t}}{\partial k_{z,t}^i} \end{pmatrix} \quad (4.35)$$

$$\frac{\partial z_{r,t}}{\partial r_{r,t}} = \frac{r_{r,t} - k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)}{\sqrt{r_{r,t}^2 + k_{r,t}^i{}^2 - 2r_{r,t}k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)}} \quad (4.36)$$

$$\frac{\partial \underline{z}_{r,t}}{\partial r_{\alpha,t}} = \frac{r_{r,t} k_{r,t}^i \sin(r_{\alpha,t} - k_{\alpha,t}^i)}{\sqrt{r_{r,t}^2 + k_{r,t}^{i2} - 2r_{r,t} k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)}} \quad (4.37)$$

$$\frac{\partial \underline{z}_{r,t}}{\partial r_{\beta,t}} = 0 \quad (4.38)$$

$$\frac{\partial \underline{z}_{r,t}}{\partial k_{r,t}^i} = \frac{k_{r,t}^i - r_{r,t} \cos(r_{\alpha,t} - k_{\alpha,t}^i)}{\sqrt{r_{r,t}^2 + k_{r,t}^{i2} - 2r_{r,t} k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)}} \quad (4.39)$$

$$\frac{\partial \underline{z}_{r,t}}{\partial k_{\alpha,t}^i} = \frac{-r_{r,t} k_{r,t}^i \sin(r_{\alpha,t} - k_{\alpha,t}^i)}{\sqrt{r_{r,t}^2 + k_{r,t}^{i2} - 2r_{r,t} k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)}} \quad (4.40)$$

$$\frac{\partial \underline{z}_{r,t}}{\partial k_{z,t}^i} = 0 \quad (4.41)$$

$$\frac{\partial \underline{z}_{\alpha,t}}{\partial r_{r,t}} = \frac{k_{r,t}^i \sin(r_{\alpha,t} - k_{\alpha,t}^i)}{r_{r,t}^2 + k_{r,t}^{i2} - 2r_{r,t} k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)} \quad (4.42)$$

$$\frac{\partial \underline{z}_{\alpha,t}}{\partial r_{\alpha,t}} = \frac{r_{r,t}^2 - r_{r,t} k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)}{r_{r,t}^2 + k_{r,t}^{i2} - 2r_{r,t} k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)} \quad (4.43)$$

$$\frac{\partial \underline{z}_{\alpha,t}}{\partial r_{\beta,t}} = -1 \quad (4.44)$$

$$\frac{\partial \underline{z}_{\alpha,t}}{\partial k_{r,t}^i} = \frac{-r_{r,t} \sin(k_{\alpha,t}^i - r_{\alpha,t})}{r_{r,t}^2 + k_{r,t}^{i^2} - 2r_{r,t}k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)} \quad (4.45)$$

$$\frac{\partial \underline{z}_{\alpha,t}}{\partial k_{\alpha,t}^i} = \frac{k_{r,t}^{i^2} - r_{r,t}k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)}{r_{r,t}^2 + k_{r,t}^{i^2} - 2r_{r,t}k_{r,t}^i \cos(r_{\alpha,t} - k_{\alpha,t}^i)} \quad (4.46)$$

$$\frac{\partial \underline{z}_{\alpha,t}}{\partial k_{z,t}^i} = 0 \quad \frac{\partial \underline{z}_{z,t}}{\partial r_{r,t}} = 0 \quad \frac{\partial \underline{z}_{z,t}}{\partial r_{\alpha,t}} = 0 \quad (4.47-4.49)$$

$$\frac{\partial \underline{z}_{z,t}}{\partial r_{\beta,t}} = 0 \quad \frac{\partial \underline{z}_{z,t}}{\partial k_{r,t}^i} = 0 \quad \frac{\partial \underline{z}_{z,t}}{\partial k_{\alpha,t}^i} = 0 \quad \frac{\partial \underline{z}_{z,t}}{\partial k_{z,t}^i} = 1 \quad (4.50-4.53)$$

Hareket güncellemesinde F_x matrisinin eşitliklerde sadece μ vektörünün ve Σ matrisinin belirli bir kısmının işleme dâhil etme görevini ölçüm güncellemesinde de $F_{x,i}$ matrisi gerçekleştirir. Bu matris ile robotun parametrelerinin yanı sıra i değerine göre her defasında haritadaki köşelerden yalnızca birinin parametreleri üzerinden işlem yapılmaktadır. 6×303 büyüklüğündeki $F_{x,1}$ ve $F_{x,2}$ matrisleri örnek olarak eşitlik 4.54 ve 4.55 'de verilmiştir.

$$F_{x,1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix} \quad (4.54)$$

$$F_{x,2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix} \quad (4.55)$$

$F_{x,i}$ ve h_t^i matrisleri çarpılarak h_t^i deki bilginin eşitlik 4.56'da belirtilen 3x303 büyüklüğündeki H_t^i matrisine geçmesi sağlanır.

$$H_t^i = h_t^i * F_{x,i} \quad (4.56)$$

Böylece eşitlik 4.57'deki gibi Σ ile çarpılan H_t^i de Σ gibi 3x303 boyutlarında olur. H_t^i 'deki 0 olmayan elemanlar ne kadar büyük olurlarsa çarpılacakları belirsizlik değerleri ile birlikte haritadaki i köşesinin gözlemlenen köşe ile eşleşmesi olasılığını düşürür.

$$\Psi_i = H_t^i \Sigma_t [H_t^i]^T + Q_t \quad (4.57)$$

Eşitlik 4.57'deki Q_t matrisi ölçümdeki belirsizlikleri belirten diyagonal bir matristir ve çalışmada diyagonal elemanları sabit alınmıştır. Ψ_i değerleri kullanılarak eşitlik 4.58'deki gibi hesaplanan Π_i değerleri ile haritadaki her i köşesinin j ölçümüyle eşleşip eşleşmediği kontrol edilmektedir.

$$\Pi_i = \left(\begin{pmatrix} z_{r,t} \\ z_{\alpha,t} \\ z_{z,t} \end{pmatrix} - \begin{pmatrix} \underline{z}_{r,t} \\ \underline{z}_{\alpha,t} \\ \underline{z}_{z,t} \end{pmatrix} \right)^T \Psi_i^{-1} \left(\begin{pmatrix} z_{r,t} \\ z_{\alpha,t} \\ z_{z,t} \end{pmatrix} - \begin{pmatrix} \underline{z}_{r,t} \\ \underline{z}_{\alpha,t} \\ \underline{z}_{z,t} \end{pmatrix} \right) \quad (4.58)$$

N_t , t anında haritada bulunan köşe sayısını belirtmek üzere, eşitlik 4.58'deki gibi hesaplanan N_t tane Π_i değerlerine ek olarak N_t+1 'ci sabit bir Π değeri de ölçülen köşe noktasının haritadaki hiçbir köşeye eşleşmemesi durumu için Π değerleri arasına eklenir. Böylece toplam N_t+1 tane Π değeri elde edilmiş olur. Bu değerlerden en küçüğü seçilerek indeksi(ind) kayıt edilir. Seçilen Π eğer N_t+1 'ci Π değeri ise haritaya yeni bir köşe eklenecektir. Eğer diğer köşelerden biri seçilmiş ise ölçüm ile haritada bulunan köşelerden birinin bilgileri güncellenir. Haritaya yeni bir köşe eklenirken köşenin durum vektörünün ilk boşlukları eşitlik 4.59-4.63'e göre doldurulur.

$$T_5 = (r_{r,t} \cos(r_{\alpha,t}) + z_{r,t} \cos(r_{\beta,t} + z_{\alpha,t}))^2 \quad (4.59)$$

$$T_6 = (r_{r,t} \sin(r_{\alpha,t}) + z_{r,t} \sin(r_{\beta,t} + z_{\alpha,t}))^2 \quad (4.60)$$

$$\mu_t(3N_t + 4) = \sqrt{T_3 + T_4} \quad (4.61)$$

$$\mu_t(3N_t + 5) = a \tan 2 \left(\frac{r_{r,t} \sin(r_{\alpha,t}) + z_{r,t} \sin(r_{\beta,t} + z_{\alpha,t})}{r_{r,t} \cos(r_{\alpha,t}) + z_{r,t} \cos(r_{\beta,t} + z_{\alpha,t})} \right) \quad (4.62)$$

$$\mu_t(3N_t + 6) = z_{z,t} \quad (4.63)$$

Sistemde tüm Ψ değişkenlerini tutmak gereksiz yere bellek işgal edeceğinden Ψ değerleri bellekte tutulmaz ve ind değerine göre eşleştirme sonrası tekrar hesaplanarak eşitlik 4.64'deki Kalman kazancı matrisinin hesabında kullanılır.

$$K_t^j = \Sigma_t [H_t^{ind}]^T \Psi_{ind}^{-1} \quad (4.64)$$

Kalman kazancı matrisi ölçülen j köşesi için hesaplanmış olur ve bu değer yardımıyla μ vektörü ve Σ matrisi eşitlik 4.65 ve 4.66'daki gibi güncellenir.

$$\mu_t = \mu_t + K_t^j \left(\begin{pmatrix} z_{r,t} \\ z_{\alpha,t} \\ z_{z,t} \end{pmatrix} - \begin{pmatrix} \underline{z}_{r,t} \\ \underline{z}_{\alpha,t} \\ \underline{z}_{z,t} \end{pmatrix} \right) \quad (4.65)$$

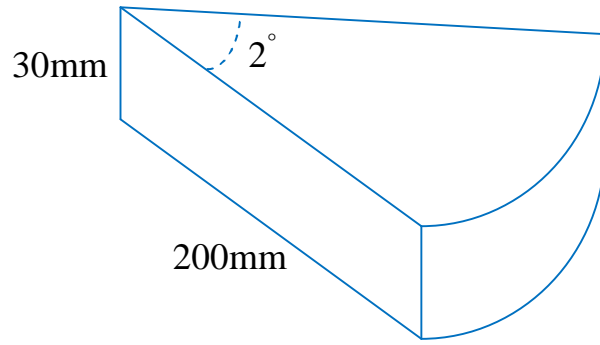
$$\Sigma_t = (I - K_t^j H_t^{ind}) \Sigma_t \quad (4.66)$$

Ölçüm güncellemesi t anında tespit edilen 3 boyuttaki her i köşesi için tekrarlanır ve tespit edilen köşeler tamamlandığında tekrar hareket güncellemesi gerçekleştirilir.

4.5.Harita Organizasyonu

Yapılan benzetimler sırasında haritadaki köşeler organize edilirken Plessey köşe tespiti algoritmasında kullanılan maksimum olmayanı baskılama yöntemi benzeri bir kısıt getirilmesi gerektiği görülmüştür. 3 boyutta oluşturulan haritalarda belirli bir hacim içerisinde sadece 1 köşe bulunabileceği varsayımı yapılmıştır. Ölçüm aşamasında bir köşe haritadaki hiçbir köşeye eşleşmemesine rağmen köşelerden

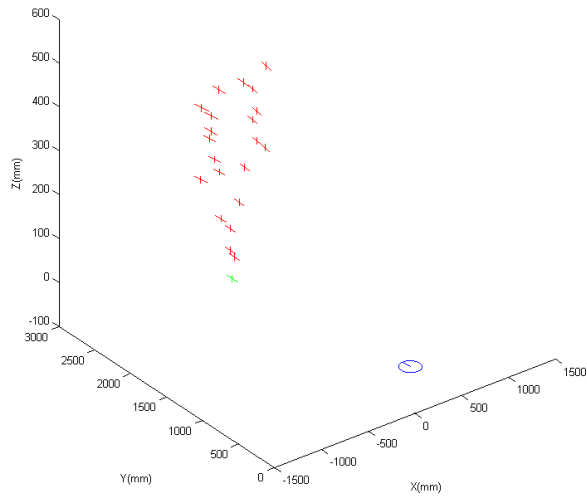
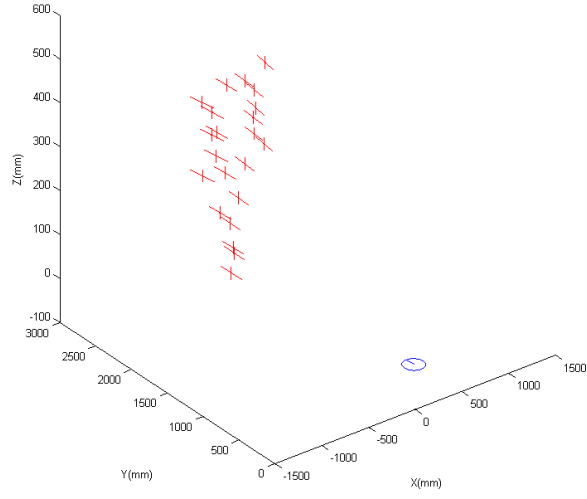
birinin içinde bulunduğu belirli büyüklükteki hacim içerisinde yer alıyorsa bu köşenin haritada yeni bir köşe olarak yerleşmesi algoritmada engellenmiştir. Böylece herhangi bir köşeye eşleşmemesine rağmen haritadaki köşelerden birinin yakınında yanlış köşelerin oluşup filtrenin yanlış çalışması engellenmiştir. Buna göre deneysel olarak bulunan hacmin boyutları



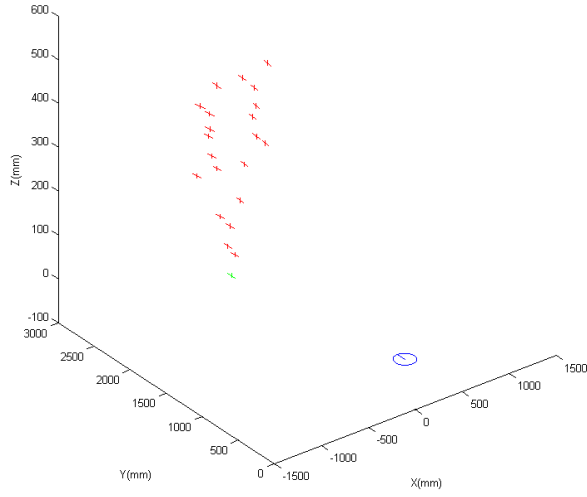
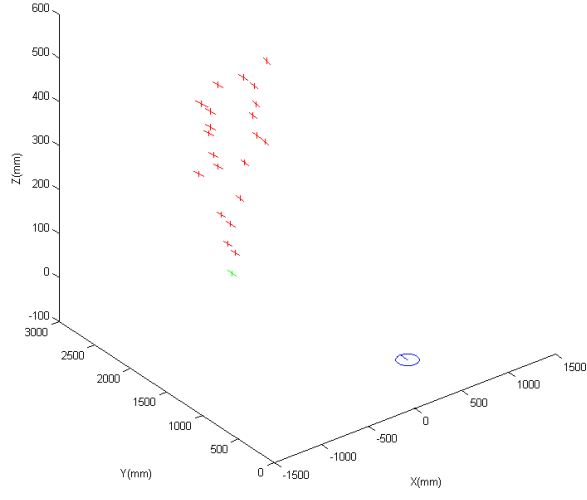
Şekil 4.5. Haritada en fazla bir köşenin bulunabileceği hacim

4.6. EZKH Benzetim Sonuçları

Daha önceki aşamalarda görüntüleri görünen bir ortamda, görüntülerdeki kapıya 242cm ile 256cm arası uzaklıklarda 2şer cm aralıkla alınmış görüntüler robot üzerindeki XBee kablosuz haberleşme modülüyle bilgisayara aktarılmıştır. Aktarılan görüntüler üzerinde köşe tespiti, stereo eşleme ve 3 boyutta konum çıkarımı aşamaları sonucu oluşan ölçümler GKF'nin ölçüm güncellemesine aktarılırken robotun konum değişimi ise hassas bir şekilde sabitlenerek sağlanmıştır. Simülasyon ortamında robot 256 cm uzaklıkta tespit edilen köşelerden başlayarak tüm görüntülerdeki köşeler yardımıyla haritasını oluşturup konumunu güncellemiştir. Simülasyon ortamındaki robot kapıya 256 cm, 252cm, 248cm ve 242cm uzaktayken robotun sahip olduğu haritadaki köşelerin konumu ve konumlarındaki belirsizlikle birlikte robotun konumu şekil 4.6. ve 4.7.'de gösterilmiştir.



Şekil 4.6. Robot kapıya 256cm(üstte) ve 250cm(alta) uzaklıktayken simülasyon ortamındaki robotun konumu ve haritasındaki köşelerin konumu ve belirsizlikleri



Şekil 4.7. Robot kapıya 246cm(üstte) ve 242cm(altta) uzaklıktayken simülasyon ortamındaki robotun konumu ve haritasındaki köşelerin konumu be belirsizlikleri

5. SONUÇLAR

Bu çalışmada bulunduğu ortam hakkında hiçbir ön bilgisi olmayan, renkli stereo kameralara sahip mobil bir robotun GKF kullanarak EZKH yapması incelenerek, kullanılan algoritmalara yönelik FPGA donanımları oluşturulmuştur. Robotun içinde bulunduğu ortamdaki köşeler özellik noktaları olarak kabul edilmiş ve bu köşelerin tespiti için Plessey köşe tespiti algoritması kullanılmıştır. Kullanılan algoritma FPGA üzerinde paralelleştirilerek çalışma zamanı kısaltılmıştır. Oluşturulan boru hattı modülleri dış modüllerden yeteri kadar yoğun beslenebilirse 640x480 çözünürlüğünde RGB555 formatındaki dijital bir görüntü üzerinde 1.5ms'de köşe tespiti gerçekleştirilebilmektedir. Daha sonra robot ile elde edilen görüntüler bilgisayar ortamına arttırılmış ve Barnard'a ait stereo eşleme algoritmasıyla sol ve sağ kameradan gelen görüntülerde bulunan köşeler arasında eşleştirilme sağlanmıştır. Kameraların iç ve dış ölçümlene parametreleri Jean-Yves Bouguet'e ait ölçümlene kütüphanesi yardımıyla elde edilmiştir. Bu parametreler yardımıyla üçgenleyerek yeniden kurma yöntemiyle stereo görüntülerdeki köşe noktalarının robota göre konumları tespit edilmiştir. Tespit edilen bu köşeler GKF'nin ölçüm aşamasına aktarılarak robottaki enkoderlardan gelen konum değişimi bilgisi yardımıyla benzetim ortamında EZKH gerçekleştirilmiştir.

Bu çalışma sonrası FPGA üzerine aktarılan Plessey köşe tespiti algoritmasıyla beraber benzetim aşamasında gerçekleştirilmiş olan stereo eşleme, üçgenleyerek yeniden oluşturma ve EZKH aşamaları için de FPGA donanımı geliştirilerek gerçek zamanda çalışan bir sistem oluşturulacaktır.

6. KAYNAKLAR

- [1] Thrun, S., Burgard, W., Fox, D., Probabilistic Robotics, *MIT Press*, 2005.
- [2] Se, S., Barfoot, T., Jasiobedzki, P., Visual Motion Estimation and Terrain Modeling for Planetary Rovers, International Symposium on Artificial Intelligence, Robotics and Automation in Space, Munich, Germany, Eylül 2005.
- [3] Williams, S., Newman, P., Dissanayake, G., Durrant-Whyte H., Autonomous Underwater Simultaneous Localization and Mapping, Proceedings of the IEEE International Conference on Robotics and Automation, 1793-1798, San Francisco, USA, Nisan 2000.
- [4] Lee, C.R., Salcic, C., High-performance FPGA-based implementation of Kalman filter, *Microprocessors and Microsystems*, 21, 257-265, 1997.
- [5] Lee, C.R., Salcic, C., A Fully-hardware-type Maximum-parallel Architecture for Kalman Tracking Filter in FPGAs, International Conference on Information, Communications and Signal Processing, ICICS97, 1243-1247, Singapore, 1997.
- [6] Piovoso, M., Laplante, P.A., Kalman filter recipes for real-time image processing, *Real Time Imaging*, 9, 433-439, 2003.
- [7] Bonato, V., Peron, R., Wolf, D., de Holanda, J., Marques, E., Cardoso, J., An fpga implementation for a kalman filter with application to mobile robotics, IEEE Second International Symposium on Industrial Embedded Systems, 148-155, Lisbon, Portugal, Temmuz 2007.
- [8] Bonato, V., Marques, E., Constantinides, G.A., Cardoso, J., A Floating-point Extended Kalman Filter Implementation for Autonomous Mobile Robots, *Journal of Signal Processing*, 56, 41-50, 2009.
- [9] Mokhtarian, F., Mohanna, F., Performance evaluation of corner detectors using consistency and accuracy measures, *Computer Vision and Image Understanding*, 102, 81-94, 2006.
- [10] Harris, C., Stephens, M., A Combined Corner and Edge Detector, In *Alvey Vision Conference*, 147-151, 1988.
- [11] Zou, L.H., Chen, J., Zhang, J., Dou, L.H., The comparison of two typical corner detection, Second International Symposium on Intelligent Information Technology Application, 211-215, 2008.
- [12] Wang, W., Dony, R.D., Evaluation of image corner detectors for hardware implementation, *Canadian Conference on Electrical and Computer Engineering*, 3, 1285-1288, 2004.

- [13] Tissainayagam, P., Suter, D., Assessing the performance of corner detectors for point feature tracking applications, *Image and Vision Computing*, 22, 663–679, 2004.
- [14] Moravec, H., *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*, Tech. Teport CMU-RI-TR-80-03, Robotics Institute, Carnegie-Mellon University, Robotics Institue, September, 1980.
- [15] Montesinos, P., Gouet, V., Deriche, R., Differential Invariants for Color Images, *International Conference on Pattern Recognition*, 14, 838-840, 1998.
- [16] Aydođdu, M.F., Demirci, M.F., Kasnakođlu., C., FPGA tabanlı paralelleřtirilmiř Plessey Kōře Tespiti, 5. Savunma Teknolojileri Kongresi SAVTEK 2010, Ankara,Türkiye, Haziran 2010.
- [17] Lane, R.A., Thacker, N.A., *Tutorial: Overview of Stereo Matching Research*, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 1998.
- [18] Barnard, S.T., Thompson, W.B., Disparity Analysis of Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4), 333-340, 1980.
- [19] Eriřim adresi: http://www.vision.caltech.edu/bouguetj/calib_doc/ eriřim tarihi:27 Temmuz 2010
- [20] Trucco, E., Verri, A., *Introductory Techniques for 3-D Computer Vision*, *Prentice Hall*, Upper Saddle River, New Jersey, 1998.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : AYDOĞDU, Muhammet Fatih
Uyruğu : T.C.
Doğum tarihi ve yeri : 10.04.1984 Elazığ
Medeni hali : Bekar
Telefon : 0 (537) 222 88 28
Faks : -
e-mail : mfatihaydogdu@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	ODTÜ/Elektrik ve Elektronik Mühendisliği	2008

İş Deneyimi

Yıl	Yer	Görev
2009-2010	Fırat Üniversitesi	Araştırma Görevlisi
2008-2009	TOBB Ekonomi ve Teknoloji Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar

Aydoğdu, M.F., Demirci, M.F., Kasnakoğlu, C., FPGA tabanlı paralelleştirilmiş Plessey Köşe Tespiti, 5. Savunma Teknolojileri Kongresi SAVTEK 2010, Ankara, Türkiye, Haziran 2010.