

ENİNE ARAMA ALGORİTMASINI KULLANARAK EN KISA YOL PROBLEMİNİN ÇÖZÜMÜNÜN LEGO MINDSTORM İLE GERÇEKLENMESİ

Orhan KÜÇÜKCEYLAN¹, Tolga YÜKSEL², Abdullah SEZGİN³

¹ 6. Ana Jet Üs Komutanlığı
Uçak Üniteleri Bkm. K.1ığı
Bandırma-Balıkesir

okucukceylan@gmail.com, tyuksel@omu.edu.tr, asezgin@omu.edu.tr

^{2,3} Elektrik-Elektronik Mühendisliği Bölümü
Ondokuz Mayıs Üniversitesi
Kurupelit-Samsun

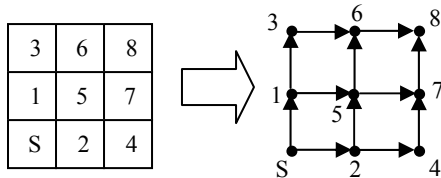
ABSTRACT

Shortest path planning is one of the most known problems in mobile robotics. Breadth-first algorithm is one of the methods used for this problem. In this study, solution of shortest path problem with Breadth-first algorithm is implemented with a Lego MindStorm RIS based mobile robot.

Anahtar kelimeler : Lego ,Enine arama,gezgin robot

1. GİRİŞ

Mobil robotlarda yol planlamada en temel problemlerden biri robotu en kısa yolu izleyecek böylece en az enerjiyi harcayacak şekilde gideceği noktaya yönlendirme problemidir. Bu problemin çözümü için kullanılan en temel yöntemler çizge-arama(graph-search) algoritmalarıdır. Bu algoritmalar hiperküp çoklu bilgisayar sistemlerinde, taşıma yönlendirmede, oyun programlamada, robotlarda yol planlamada ve labirent çözüme kullanılmaktadır. Bu algoritmaların mobil robotlarda kullanımı için robotun sahip olduğu haritada düğüm noktalarının belirlenmesi gerekmektedir. Düğüm noktalarını ayrı ayrı belirlemedeki zorluklardan dolayı bunun yerine tüm gezilebilecek alana ait haritanın ızgara tabanlı hale getirilmesi ve her bir ızgara gözünün en temel hareket birimi(hücre) olarak adlandırılması algoritmaların işleyişini kolaylaştırır. Şekil 1'de ızgara tabanlı bir harita ve onun çizge eşdeğeri gösterilmiştir.



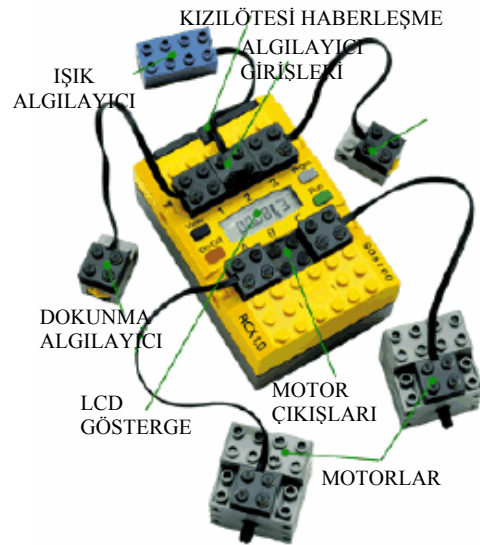
Şekil 1 ızgara tabanlı harita ve çizge eşdeğeri

Bu çalışmada çizge arama algoritmalarından Enine Arama algoritması incelenmiş, Enine Arama algoritmasının en kısa yol probleminin çözümünde nasıl kullanıldığından bahsedilmiş, algoritmanın

gerçeklenmesi için kullanılan Lego MindStorm gezgin robotu ve oluşturulan düzenek tanıtılmış, algoritmanın çalışması üzerine örnekler verilmiştir.

2. LEGO MINDSTORM RIS , leJOS ve JMF

1980'lerde MIT'deki bir grup bilim adamının çocuklara basit programlanabilir bir gezgin robot yapma fikri gelişen süreç içinde 1998 'de Lego çatısı altında Lego MindStorm Robotics Invention System(RIS) setinin ortaya çıkmasıyla sonuçlanmıştır. Popülerliği, kolay programlanabilir olması, uygun fiyatı ve gezgin robotlar için temel oluşturan özellikleri sayesinde RIS seti kullanıcıları tarafından benimsenmiş ve geçen zaman içinde yazılımsal ve donanımsal açıdan birçok yeniliğe uğramıştır. Özellikle yazılımsal açıdan RIS seti birçok yeniliğe uğramış, kendine ait grafiksel programlama ortamından farklı olarak birçok bağımsız programlama ortamı ortaya çıkmış, gerçekten programlama mantığı ve bilgisi gerektiren C++, JAVA gibi yüksek seviye diller RIS setini programlamak için kullanılabilir hale gelmiştir.

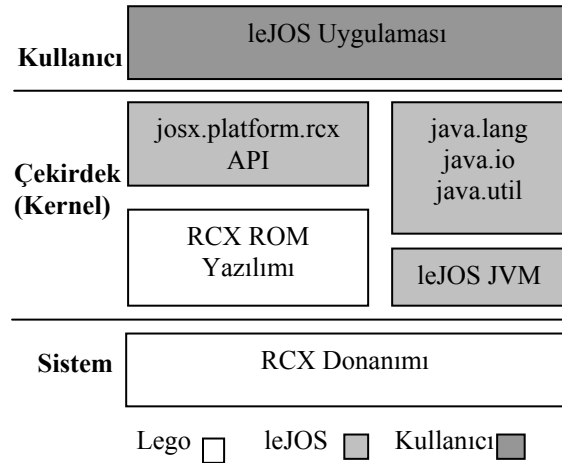


Şekil 2 Lego MindStorm Genel Bileşenleri

Şekil 2’de RIS setine ait temel bileşenler gösterilmiştir. Setin temelini RCX olarak adlandırılan 8 bitlik 16 MHz hızında Hitachi H8/3292 mikroişlemciye sahip modül oluşturmaktadır. RCX ve diğer bileşenlere ait bilgiler kaynaklardan edinilebilir[1],[2].

RIS seti ile oluşturulan robotların programlanmasında çeşitli diller ve ortamlar kullanılmaktadır. Bunların başında Lego RIS programlama ortamı, RoboLab gibi görsel arayüze sahip ortamlar gelmektedir[1],[2],[7]. Bu ortamlar görsel arayüz kullanmalarından dolayı RCX’e ait özelliklerden tam olarak yararlanamamaktadır. Bu yüzden yüksek seviye dilleri kullanan NQC,leJOS gibi programlama ortamları ortaya çıkmıştır.

Bu çalışmada programlama ortamı olarak tercih edilen leJOS JAVA programlama dilini kullanan bir programlama ortamıdır[1],[3],[5]. JAVA gibi nesne tabanlı ve internet üzerinden erişime imkan veren bir dil üzerine kurulması leJOS’u Lego robotları için popüler bir hale getirmiştir.leJOS’ta tanımlı komut sınıfları ve bunların RCX ile aralarındaki yapısal bağlantı Şekil 3’te verilmiştir.



Şekil 3 leJOS sınıfları ve RCX bağlantısı

JMF(Java Media Framework) JAVA programlama dili için görüntü ve ses gibi çoklu ortamlar için oluşturulmuş bir API’dir[4]. leJOS görüntü ve ses tanıma gibi özellikleri bünyesine alabilmek amacıyla bu API’den yararlanarak **Vision** oluşturmuştur. Bu çalışmada bu sınıflardan **Vision** sınıfı kullanılarak çeşitli görüntü işleme işlemleri yapılmıştır.

3. ENİNE ARAMA ALGORİTMASI

Çizge arama algoritmaları çok çeşitli olsa da istenen aynı anda hem en kısa yolu bulan, hem de optimum aramayla bu işi yapan algoritmayı kullanmaktır. Bunlardan biri de Enine Arama (Breadth-first) algoritmasıdır[6],[8]. Enine Arama algoritması başlangıç hücresinden başlayarak hedef hücre bulunana kadar komşu hücrelere dallanma ilkesiyle çalışır. Bu çalışma esnasında hareket

edilemez(engel) hücreler gözardı edilir. Başlangıç hücresinden başlayarak tüm hareket edilebilir hücreler sırayla AÇIK LİSTE adı verilen bir diziyeye dahil edilir. AÇIK LİSTE dizisi hedef hücreyi bulmak amacıyla eklenen komşu hücreleri tutan dizidir ve bu dizide eklenen yeni elemanların hedef hücre olup olmadığı kontrol edilir. Eklenen hücreler kontrol edildikten sonra bunlara ait yeni komşu hücreler AÇIK LİSTE dizisine eklenir ve bu prosedür hedef hücre bulunana kadar devam eder. Başlangıç hücresinin maliyeti 0’dır. Eklenen her bir komşu hücrenin toplam maliyeti ise kendisini ekleyen hücre+1(veya +tanımlanan sabit değer) şeklinde olacaktır. Hücrelerin maliyetleri harita boyutlarında bir maliyet matrisinde saklanacaktır. Eklenen yeni komşu hücreler incelendikten sonra kontrol edilmiş eski hücreler AÇIK LİSTE’den çıkarılacak, bu sayede hem dizinin boyutu fazladan büyümeyecek hem de gözden geçirilen eski hücreler tekrar gözden geçirilmeyecektir. Bu esnada eğer AÇIK LİSTE boş diziyeye dönüşürse bu, mümkün bir yolun olamayacağı anlamına gelir. Hedef hücre AÇIK LİSTE’ye eklendiğinde en kısa yolu bulmak için maliyet matrisi üzerinde hedef hücresinden başlangıç hücresine doğru adım adım hücrelerin maliyetlerini izlemek yeterli olacaktır. Enine arama algoritmasının adımları aşağıda verilmiştir.

Enine arama algoritması

1. Başlangıç ve hedef hücreyi tanımla.
2. Harita matrisi tanımla.
3. Başlangıç hücresini AÇIK LİSTE’ye ekle.
4. Komşu hücreleri AÇIK LİSTE’ye ekle,maliyet matrisine maliyetlerini yaz..
5. Eğer AÇIK LİSTE boşsa olası yol yok , çalışmayı durdur.
6. Eğer hedef hücre AÇIK LİSTE’ye eklenmişse YOL’u maliyet matrisine bakarak bul , değilse devam et.
7. Gözden geçirilenleri AÇIK LİSTE’den çıkar.
8. Adım 4’e dön.

-	6	-	-	-	-	-	-	-
6	5	6	-	-	-	-	-	-
5	4	5	6	-	-	-	-	-
4	3	4	-	6	-	-	-	-
3	2	3	-	5	6	6	-	-
2	1	2	-	4	5	6	-	-
1	S	1	2	3	4	5	6	-
2	1	2	3	4	5	6	-	-
3	2	3	4	5	6	-	-	-

Şekil 4 Enine Arama algoritması ile oluşturulan harita

Enine Arama algoritmasının yürütülmesiyle oluşan maliyet haritası ve bulunan yola ait bir gösterilim Şekil 4'te verilmiştir. Şekil 4 incelendiğinde hedef hücre bulunduğu aramanın sonlandırıldığı ve bu sayede gereksiz hücre maliyet hesaplarından kaçınıldığı açıkça görülmektedir.

4. DENEYSEL ÇALIŞMALAR

Lego MindStorm ile birçok görevi yerine getiren gezgin robotlar gerçekleştirmek mümkündür. Bu görevlerden biri de belirli bir giriş ve çıkış noktaları bulunan ve giriş çıkış arasında en az bir olası yol bulunan sınırlanmış bir alan içinde oluşturulmuş engelli bir alanda başlangıç noktasından bitiş noktasına en kısa yolu kullanarak varabilmektir. Bu çalışmada bu görevin gerçekleştirilecek bir robot dizgesinin tasarlanması ve gerçekleştirilmesi hedeflenmiştir. Tasarlanan bu dizgeye ait bileşenler sırasıyla açıklanmıştır.

a) Gerçeklenen Robot(RoverBot)

Öncelikle bu görev için bir gezgin robot(RoverBot) gerçekleştirilmiştir[2].



Şekil 5 Gerçeklenen robot(RoverBot)

Şekil 5'te verilen bu robot diferansiyel sürüş yöntemiyle sürülmektedir. Robotun yolda düz gitmesini sağlamak amacıyla her iki tarafta bulunan bir motora birer adet konum algılayıcı(rotation sensor) yerleştirilmiştir. Konum algılayıcıları belirli zaman aralıklarında konum bilgilerinin karşılaştırarak düz gitme işlemini sağlamak amacıyla gerekli komutların motora yollanmasını sağlamaktadır.

b) Çalışma alanı

RoverBot'un çalışma alanı olarak 100cm×130cm boyutlarında bir beyaz alan oluşturulmuştur. Alan 6*8 boyutlarında 48 eşit hücreye bölünmüştür. Burada hücre sayısını sınırlayan iki öğe bulunmaktadır. Bunlardan birinci robotun boyutlarıdır. Tasarlanan sistemde hücrenin boyutu robotun boyutundan mutlaka daha büyük olmak zorundadır. İkinci olarak kısıtlayıcı öğe görüş sistemidir. Bu kısıta ait bilgiler ikinci bölümde verilecektir.

c) Görüş sistemi

Gezgin robotlarda görüş sistemi oluşturmak için iki yol izlenebilir. Birincisi görüntüyü alacak olan kameranın robot üzerine yerleştirilmesi, ikincisi ise

kameranın tüm çalışma alanının görecektek şekilde GPS benzeri şekilde yerleştirilmesidir. Her iki yöntemde uygulamalara bağlı olarak çeşitli üstünlükler ve zayıflıklar içermektedir. Bu çalışma için ikinci yol benimsenmiş ve bu görevi yerine getirmek için 1.3 MP çözünürlüğe sahip bir webcam kullanılmıştır.

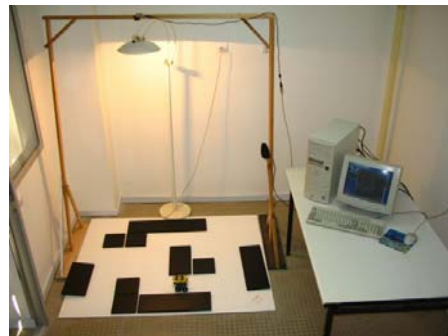
Alanı görecektek şekilde yerleştirilen kamera bilgisayardan gelen komut yardımıyla görüntüyü almakta ve eşit boyutlu hücrelere bölmektedir. Hücrelerin boyutu, başlangıç konumu ve sayısı gibi değişkenler kullanıcı tarafından tanımlanmaktadır. Bu işlem için kullanılacak bölge sayısı **Vision** sınıfı tarafından 32 bölge ile sınırlanmıştır. Bu sınıf üzerinde yapılan modifikasyonlarla bu sınır 64'e çıkarılmıştır. Üst sınırı daha üst sayıya çıkarmak mümkün olsa da renk hesabı için kullanılan algoritmanın bu durumlarda üstünlüğünü yitirdiği gözlemlenmiştir.

Alınan alan görüntüsü hücrelere bölündükten sonra her bir hücreye ait RGB renk ortalamaları **Vision** sınıfı tarafından hesaplanmakta ve bir dizi altında sunulmaktadır. Alanda engel olarak tanımlanacak siyah cisimler bu ortalamalardan en fazla kontrasta sahip yeşil renk ortalamaları kullanılarak bulunmuş, alandaki her bir hücre siyah-beyaz ayrımıyla engel ve engel harici görüntü şeklinde tanımlanmıştır. Kullanılan program yardımıyla alınan görüntüye ait bir örnek Şekil 6'da verilmiştir.



Şekil 6 Görüş sisteminden alınan örnek görüntü

Bahsedilen görüş sistemi ve görüntü işleme ile ilgili işlemler robotla bilgisayar arasında haberleşmeyi sağlayan RIS setine ait kızılötesi haberleşme cihazı IR tower'ın bağlı olduğu bilgisayarda yürütülür. Şekil 7'de tüm sistemin bir görüntüsü verilmiştir.



Şekil 7 Oluşturulan sistemin genel görünüşü

d) Algoritma

Çalışmada robottan istenen giriş(1.) hücresinden başlayarak engellere çarpmadan en kısa yolu izleyerek çıkış(48.) hücresine varabilmesidir. Bu görevi gerçekleştirmek için Bölüm 3'te anlatılan Enine Arama algoritması kullanılmıştır. Görüş sistemi ile alınan görüntü ile çalışma alanının matris şeklinde bir haritası oluşturulur. Oluşturulan matriste kapalı bir alan tanımı için birer satır ve sütun engel satırı eklenir. Tamamlanan harita matrisi için Enine Arama algoritması yürütülür. Algoritma izlenecek yolu belirledikten sonra robotun izleyeceği yola dair hareket dizisi oluşturulur. Oluşturulan hareket dizisine bağlı olarak da gerekli hareket komutları robota gönderilir ve RCX üzerindeki program yardımıyla bu komutlar yürütülür. Şekil 6 ile verilen örnek alan için robotun oluşturduğu maliyet matrisi ve izleyeceği yolun matris üzerinde gösterilimi Şekil 8'de verilmiştir.

100	100	100	100	100	100	100	100	100	100	100
100	1	2	3	100	100	12	0	0	0	100
100	100	100	4	100	10	11	100	0	0	100
100	100	100	5	100	9	100	100	12	100	
100	8	7	6	7	8	9	10	11	100	
100	9	100	100	100	100	100	100	12	100	
100	10	11	12	0	0	0	0	0	99	100
100	100	100	100	100	100	100	100	100	100	100

a) Maliyet matrisi

100	100	100	100	100	100	100	100	100	100	100
100	1	1	1	100	100	0	0	0	0	100
100	100	100	1	100	0	0	100	0	100	
100	100	100	1	100	0	100	100	0	100	
100	0	0	1	1	1	1	1	1	1	100
100	0	100	100	100	100	100	100	1	100	
100	0	0	0	0	0	0	0	0	1	100
100	100	100	100	100	100	100	100	100	100	100

b) İzlenecek yolun gösterilimi

Şekil 8 Örnek alan için elde edilen maliyet ve yol matrisleri

Tasarlanan sistem yukarıdaki aşamalarla gerçekleşmiştir. Robot istenen görevi başarıyla yerine getirmektedir. Ancak içinde iki önemli problemi barındırmaktadır. Birincisi kullanılan webcam'e bağlı olarak elde edilen görüntüde ortamdaki ışığın miktarına göre görüntü renk değerleri değişmekte ve bazı durumlarda engel olmayan beyaz kısımlar engel olarak algılanabilmektedir. İkincisi ise robotun düz gitme problemi. Kullanılan eşdeğer motorlara ve eklenen konum algılayıcılarına rağmen robot düz çizgi üzerinde ilerlerken tekrar edilebilirliğinde büyük sapmalar görülmektedir.

5. SONUÇLAR

Bu çalışmada belirli bir çalışma alanında başlangıç ve bitiş noktaları arasında Enine Arama algoritmasını kullanarak bulduğu en kısa yolu izleyen Lego MindStorm RIS seti ile inşa edilmiş bir robot sistemi tasarlanmış ve gerçekleşmiştir. Sistem istenilen başarıyı elde etmiştir. İleriki çalışmalarda sistemde görülen görüntü bozulmalarına dair problemler daha iyi merceğe

sahip bir kamera kullanılarak ve motorlardaki eşdeğer olmama problemi çeşitli on-off zamanlama algoritmaları kullanılarak giderilmesi hedeflenmektedir. Ayrıca yapılan çalışmada tanımlanan hücrelerin maliyetleri aynı(+1) olarak kabul edilmiştir. İleriki çalışmalarda farklı engeller için(su,buz,pürüzlü yüzey) farklı maliyetli hücrelere sahip çalışma alanları için farklı algoritmalar kullanarak çalışmanın genişletilmesi hedeflenmektedir.

6. EK BİLGİ

Bu çalışma O.M.Ü. E.E.M. Bölümü'nde yürütülen MF104 no'lu projenin bir parçasıdır.

7. KAYNAKLAR

- [1] M. P Scholz., B.Bagnall , L. Griffiths , “ The leJOS Tutorial “ ,2003 , www.lejos.org/tutorial
- [2] The LEGO Company , “ Lego Mindstorms Robotic Invention System 2.0 Constructopedia” 2000
- [3] leJOS API Docs, http://switch.dl.sourceforge.net/sourceforge/lejos/lejos_win32_2_1_0.doc.zip ,2003
- [4] The Sun Microsystems , “ JAVA Media Framework API Guide “ http://java.sun.com/products/java_media/jmf/2_1.1/apidocs , 1999
- [5] Deitel H.M. , Deitel P. J. , “ JAVA How to program ” , Deitel Books , 2003
- [6] Kelly Manley ,” Pathfinding: From A* to LPA” , Computer Science Seminar Spring , University of Minnesota ,2003
- [7] Tolga Yüksel , “ Özel Amaçlı Gezgin Robotların Denetimi ” Yüksek Lisans Tezi , Ondokuz Mayıs Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü , 2004
- [8] Tolga Yüksel , Abdullah Sezgin , “ Mobil Robotlar İçin Izgara Tabanlı Haritalarda En Kısa Yol Algoritmalarını Kullanarak Yol Planlama “ , TOK'06 , TOBB ETÜ , 2006
- [9] Orhan Küçükceylan, “Gezgin Robotların Engelli Bir Ortamda Hedefe Ulaşma Probleminin İncelenmesi”, Yüksek Lisans Tezi ,Ondokuz Mayıs Üniversitesi Elektrik-Elektronik Müh. Bölümü, 2007

Orhan KÜÇÜKCEYLAN 1981 yılında Ankara'nın Elmadağ ilçesinde doğdu. 2003 yılında Ondokuz Mayıs Ün. Elektrik-Elektronik Müh. Bölümü'nde lisans, 2007 yılında aynı bölümde yüksek lisans eğitimini tamamladı. Şu anda Balıkesir 6. Ana Jet Üs Komutanlığı'nda teğmen olarak görev yapmaktadır. İlgi alanları arasında robotik ve programlama yer almaktadır.

Tolga YÜKSEL 1980 yılında Karabük'te doğdu. 2002 yılında Ondokuz Mayıs Ün. Elektrik-Elektronik Bölümü'nde lisans, 2004 yılında aynı bölümde yüksek lisans eğitimini tamamladı. 2002 yılında aynı bölüme araştırma görevlisi olarak atanmış ve halen aynı görevine devam etmektedir. Çalışma alanları arasında otomatik kontrol sistemleri, robotik ve otomasyon yer almaktadır.

Abdullah SEZGİN 1957 yılında Trabzon'da doğdu. 1978 Yılında Karadeniz Teknik Üniversitesi Elektrik – Elektronik Müh. Bölümünü bitirdi. 1983 yılında Yüksek Lisans, 1993 yılında Doktora eğitimini aynı kurumda tamamladı. 1979 – 1993 Yılları arasında KTÜ 'de görev yaptı. 1993 yılından bu yana OMÜ Mühendislik Fakültesi, Elektrik – Elektronik Müh. Bölümü, Kontrol ve Kumanda Sistemleri Anabilim Dalı 'nda öğretim üyesi olarak çalışmaktadır. İlgi alanları arasında; Elektrik Makinaları, Güç Elektroniği, Kontrol Sistemleri ve Robotik konuları yer almaktadır.