

Vector Field Histogram Algoritmasıyla Arama ve Kurtarma Robotlarında Engelden Sakınma

Obstacle Avoidance with Vector Field Histogram Algorithm for Search and Rescue Robots

Gürkan Şahin, Muhammet Balcılar, Erkan Uslu, Sırma Yavuz, M. Fatih Amasyalı

Bilgisayar Mühendisliği Bölümü

Yıldız Teknik Üniversitesi

İstanbul, Türkiye

{gurkan,muhammet,erkan,sirma,mfatih}@ce.yildiz.edu.tr

Özetçe—Arama-kurtarma robotları üzerine yapılan araştırmalarda en önemli konulardan bir tanesi robotların ortamdaki engellere çarpmadan otonom bir şekilde ilerleyebilmesidir. Bu çalışmamızda USARSim ortamındaki sanal robotlara engelden sakınma özelliği kazandırılmıştır. Buradaki amaç robotun önüne bir engel çıktığında hedeften en az uzaklaşacak şekilde manevralar yaparak engeli çarpmamasıdır. Engelden sakınma için gerçek zamanlı yol planlaması yapan Vector Field Histogram (VFH) yöntemi kullanılmıştır. Yapılan deneyler sonucunda bu yöntemin engellerden sakınmada başarılı olduğu ve birim zamanda gezilen mekan miktarını büyük ölçüde arttırdığı gözlemlenmiştir.

Anahtar Kelimeler — VFH algoritması; engelden sakınma; USARSim; otonom robotlar.

Abstract—The capability of avoid obstacles is the one of the key issues in autonomous search-and-rescue robots research area. In this study, the avoiding obstacles capability has been provided to the virtual robots in USARSim environment. The aim is finding the minimum movement when robot faces an obstacle in path. For obstacle avoidance we used a real time path planning method which is called Vector Field Histogram (VFH). After experiments we observed that VFH method is successful method for obstacle avoidance. Moreover, the usage of VFH method is highly increasing the amount of the visited places per unit time.

Keywords — VFH algorithm; obstacle avoidance; USARSim; autonomous robots.

I. GİRİŞ

USARSim (Unified System for Automation and Robot Simulation) oyun motoru tabanlı mobil robot simülasyon ortamıdır [1]. USARSim sayesinde ortamın haritasını çıkarma, otonom olarak gezinebilme, kurban tanıma gibi algoritmaların geliştirilmesi ve test edilebilmesi mümkün olmaktadır. Bu işlemlerin yapılmasında robot üzerinde bulunan lazer, sonar, kamera, odometri vb. sensörler kullanılmaktadır.

Sanal arama-kurtarma robotları USARSim ortamında kurban tespiti amacıyla kullanılmaktadır. USARSim ortamında çeşitli engeller bulunmakta ve robotun bu engellere çarpmadan ilerleyerek kurbanları bulması amaçlanmaktadır. Bu nedenle robotların etraftaki engellere çarpmayacak şekilde otonom hareket edebilmeleri büyük öneme sahiptir.

Bu çalışmamızda USARSim ortamında otonom şekilde hareket edebilen sanal arama-kurtarma robotlarına engelden sakınma özelliğinin kazandırılması hedeflenmiştir. Engelden sakınma için literatürde VFH [2] olarak geçen yöntem kullanılmıştır. İkinci bölümde USARSim ortamı ve sanal arama-kurtarma robotları hakkında bilgi verilmiş, üçüncü bölümde engelden sakınmadan otonom gezinim anlatılmıştır. Dördüncü bölümde engelden sakınma için kullanılan yöntem açıklanmış, beşinci bölümde elde edilen deneysel sonuçlar verilmiş, son bölümde ise sonuçlar yorumlanmıştır.

II. USARSIM ORTAMI VE SANAL ROBOTLAR

USARSim ortamında kurban arama işlemlerinde kullanılmak üzere çeşitli tiplerde robotlar bulunmaktadır. P3AT (Pioneer 3 Advanced Terrain), Kenaf, AirRobot bu türlere örnek olarak verilebilir. P3AT [3] 4 tekerlekli bir robottur. Kenaf [4] ise engebeli arazide üstün hareketlilik yeteneği kazanması için dizayn edilmiş paletli bir robottur. USARSim ortamında P3AT ve Kenaf robotlarına ait görüntüler Şekil 1’de verilmiştir.



Şekil 1. USARSim ortamında P3AT ve Kenaf robotları

III. ENGELDEN SAKINMADAN OTONOM GEZİNİM

Engellerden sakınmadan otonom gezinme yönteminde robotun ve gidilecek hedef noktanın 2 boyutlu (x, y) konum

bilgileri kullanarak sağ ve sol tekerlek hızları hesaplanmaktadır. Robotun konum bilgisi her değiştiğinde hedefe gitmek için yeni teker hızları hesaplanmaktadır. Robotun hareketi sırasında belirlenen hedef noktalara Astar algoritmasının belirlediği yol ile gidilmektedir. Engellerden sakınmadan otonom gezinme yöntemimizde robot hareketi sırasında lazer sensör verileri kullanılmadığı için robot sık sık engellere çarpmaktadır. Bu çalışmamızda lazer sensör bilgilerinin de kullanılarak robotun engellere takılma sayısının azaltılması ve harita içinde daha fazla noktaya otonom olarak gidebilmesi amaçlanmıştır.

IV. VFH İLE ENGELDEN SAKINARAK OTONOM GEZİNİM

USARSim ortamındaki robotlarımızın engelden sakınmaları için literatürde VFH olarak geçen yöntem kullanılmıştır. Gerçek zamanlı hareket planlaması yapan bu algoritma 1991 yılında Johann Borenstein ve Yoram Koren tarafından önerilmiştir [5]. Bu yöntem hedefe olan en kısa yolu vermemektedir.

Yöntem robotun üzerinde bulunan lazer sensör bilgilerini kullanmakta ve robotun geçebileceği boş bölgeleri tespit etmektedir. Bu boş bölgelerden en uygun olanını (en düşük maliyetlisini) seçerek robotu bu noktaya doğru yönlendirmektedir. Yöntemin 4 önemli parametresi bulunmaktadır. Bunlardan ilki robotun kaç metre önüne bakılacağını belirten lazer eşik mesafe değeridir. Belirlenen bu eşik mesafe değerine kadar robotun önü açıksa robotun bu noktalara hareketi mümkün olmaktadır. Lazer verilerinden gelen mesafe bilgisi eşik mesafe değerinin altında ise bu lazer verisi doğrultularında engel olduğu anlaşılmakta ve robotumuzun bu yönler gitmesi engellenmektedir.

Diğer 3 önemli parametre ise daha önce de belirttiğimiz maliyet fonksiyonunun parametreleridir. Belirlenen lazer eşik mesafe değeri ve robotun eni göz önüne alınarak robotun geçebilmesi için gereken minimum açı aralık değeri hesaplanmaktadır. USARSim ortamında kullandığımız robotun tipi P3AT ve eni de yaklaşık olarak 0.7 m' dir. Yaptığımız hesaplamalar sonucunda lazer mesafe eşik değerinin 2 m alınması durumunda robotun geçebileceği minimum mesafenin yaklaşık olarak 0.7 m olabilmesi için herhangi iki doğrultudaki eşik değerinden büyük lazer ölçümleri arasındaki açının yaklaşık olarak 20 derece olması gerektiği görülmüştür.

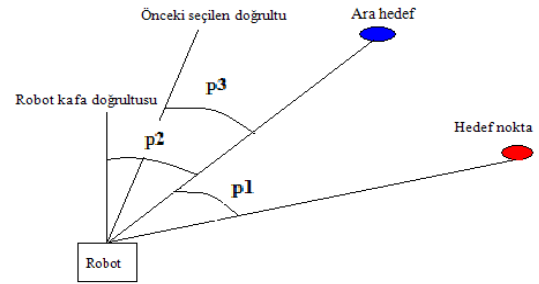
Robotumuz üzerindeki lazer sensörü 1 derece aralıklarla 0-180 derece olmak üzere 181 adet mesafe ve açı değeri döndürmektedir. Belirlenen eşik mesafe değerine göre 0-180 derece arasındaki robotun geçebileceği her aralığın orta noktası bulunarak ara hedef olarak belirlenmektedir. Belirlenen herbir ara hedef noktanın gideceğimiz hedefe olan maliyeti hesaplanarak en düşük maliyetli olan ara hedef nokta seçilmekte ve robot bu noktaya yönlendirilmektedir. Belli bir gezinme sonrasında seçilen en düşük maliyetli ara noktanın gideceğimiz hedef noktası ile çıkışması ve hedefe ulaşılması beklenmektedir. Ara hedef noktaların maliyetlerinin hesaplanmasında aşağıdaki ifade kullanılmaktadır.

$$G = a*p1 + b*p2 + c*p3 \quad [6]$$

- **p1:** Ara hedef doğrultusu ile hedef nokta doğrultusu arasındaki açı farkı değeridir. Bu açı farkını minimum yapan ara hedeflerin seçilmesi amaçlanır.

- **p2:** Robotun kafa doğrultusu ile ara hedef doğrultusu arasındaki açı farkıdır. Bu parametre robotun seçeceği ara hedef noktasını en az manevra ile seçmesini amaçlar.
- **p3:** Bir önceki durumda seçilen doğrultu ile bir sonraki seçilen ara hedef doğrultusu arasındaki açı farkıdır. Bu parametre değeri robotun önceki seçtiği doğrultudan çok sapmamasını sağlamayı amaçlar.

Bu parametrelerin önündeki a, b, c katsayıları ise parametrelerin ağırlık değerleridir. Bu ağırlık değerlerinin belirlenmesiyle ilgili çalışmalar Deneysel Sonuçlar bölümünde verilmiştir. Bu parametreler daha anlaşılır bir biçimde Şekil 2' de gösterilmiştir.



Şekil 2. Parametre değerleri

V. DENEYSEL SONUÇLAR

Bu bölümde maliyet fonksiyonundaki parametrelerin ağırlıklarının (a, b, c) değiştirilmesinin robotun hareketine etkileri incelenmiştir. Çeşitli ağırlık değerleri için denemeler yapılmış ve optimum ağırlık değerleri elde edilmeye çalışılmıştır. Bulunan optimum ağırlık değerleri ile çeşitli haritalar üzerinde test yapılmış, sonuçlar yorumlanmıştır. Yapılan bütün deneylerde lazer eşik mesafe değeri 2 m olarak kullanılmıştır.

A. Parametre Katsayılarının Değişiminin Etkileri

Ara hedef nokta seçmede kullandığımız maliyet fonksiyonundaki parametrelerin katsayılarından bir tanesi artırılıp diğerleri sabit tutulmuştur. Böylece herbir parametrenin robotun yaptığı harekete etkileri gözlenmiştir.

p1 parametre katsayısının (a) değiştirilmesi sonucu robotun hedefe ulaşıp ulaşmadığı (+/-), geçen süre bilgileri Tablo 1' de verilmiştir.

Katsayılar			Süre(sn)	Sonuç	Şekil
a	b	c			
1	1	1	24	+	3.a
2	1	1	22	+	3.b
3	1	1	24	+	3.c

Tablo 1. p1 parametre katsayısının değişiminin etkileri

Robotun izlediği yol görünümü Şekil 3' de verilmiştir. Şekillerde siyah ile gösterilen yerler engelleri, yeşil nokta robotun başlangıç konumunu, sarı noktalar robotun izlediği yolu, gri bölgeler robotun lazer sensörü ile taradığı bölgeleri, mavi bölgeler lazer sensör ile taranmayan bölgeleri, kırmızı nokta

robotu, kırmızı üçgen robotun baktığı yönü, koyu mavi nokta ise hedefi temsil etmektedir.



Şekil 3. p1 parametre katsayısının harekete etkileri
(3.a) (3.b) (3.c)

Şekillerden görüldüğü gibi p1 parametresinin önem katsayısı arttıkça robot hedef doğrultusuna en yakın olan ara hedefleri seçtiğinden hedefe daha düz bir yol çizerek gitmeye çalışmıştır.

p2 parametre katsayısının (b) değiştirilmesi sonucu robotun hedefe ulaşip ulaşmadığı, geçen süre bilgileri Tablo 2' de verilmiştir.

Katsayılar			Süre(sn)	Sonuç	Şekil
a	b	c			
1	1	1	24	+	4.a
1	2	1	57	-	4.b
1	3	1	59	-	4.c

Tablo 2. p2 parametre katsayısının değişiminin etkileri

Robotun izlediği yol görünümü Şekil 4' de verilmiştir.



Şekil 4. p2 parametre katsayısının harekete etkileri
(4.a) (4.b) (4.c)

Şekillerden görüldüğü gibi p2 parametresinin önem katsayısı arttıkça robot kafa doğrultusunu en az hareket ettireceği ara hedef noktalarını seçmiştir. Burada robotun başlangıç kafa doğrultusu sağ yöne doğrudur. Dikkat edilirse robot Şekil 4.b ve 4.c' de 4.a' dan farklı bir yol izlemiştir. 4.a' da robotumuz kafa doğrultusunu hemen döndürerek hareket ederken 4.b ve 4.c' de ise p2 parametresi katsayısı arttığından dolayı kafa doğrultusunu çok fazla değiştirmeyen ara hedefleri seçmiş ve farklı bir rota izlemiştir.

p3 parametre katsayısının (c) değiştirilmesi sonucu robotun hedefe ulaşip ulaşmadığı, geçen süre bilgileri Tablo 3' de verilmiştir.

Katsayılar			Süre(sn)	Sonuç	Şekil
a	b	c			
1	1	1	24 sn	+	5.a
1	1	2	26 sn	+	5.b
1	1	3	25 sn	+	5.c

Tablo 3. p3 parametre katsayısının değişiminin etkileri

Robotun izlediği yol görünümü Şekil 5' de verilmiştir.



Şekil 5. p3 parametre katsayısının harekete etkileri
(5.a) (5.b) (5.c)

Şekillerden görüldüğü gibi p3 parametresinin önem katsayısı arttıkça robotun seçeceği ara hedefin doğrultusu bir önceki seçtiği doğrultuya daha bağımlı hale geldiğinden Şekil 5.a' da hedefe ulaşırken kıvrımlı bir yol izlemişken, 5.b ve 5.c' de daha az kıvrımlı (daha yumuşak) bir yol izlemiştir.

B. Uygun Parametre Katsayılarının Bulunması

Uygun parametre katsayılarını bulmak amacıyla çeşitli denemeler yapılmıştır. Arama uzayı çok geniş olduğundan sadece 1, 2, 3 katsayıları için denemeler yapılmıştır. Herbir olası katsayı dizisi için 3'er deneme yapılmış hedefe ulaşılan en küçük süre alınmıştır. Bu işlemin yapılmasının nedeni sensör verilerine eklenen gürültünün değişmesinden dolayı robotun aynı katsayı değerleri için farklı yollar izleyebilmesidir. Hedefe ulaşmada 60 saniye üst sınır değeri olarak kullanılmış, bu süre içinde hedefe varanlar değerlendirmeye alınmıştır. Yapılan denemeler sonucu elde edilen sonuçlar Tablo 4' de verilmiştir.

Parametre katsayıları (a b c)	Sonuç	Süre(sn)
1 1 1	+	25.52
1 1 2	+	26.43
1 1 3	-	60.00
1 2 1	+	41.80
1 2 2	+	25.28
1 2 3	+	26.32
1 3 1	-	60.00
1 3 2	-	60.00
1 3 3	+	25.94
2 1 1	+	26.78
2 1 2	+	25.83
2 1 3	+	25.43
2 2 1	+	25.21
2 2 3	+	25.63
2 3 1	-	60.00
2 3 2	+	25.39
2 3 3	+	26.51
3 1 1	+	52.25
3 1 2	+	25.33
3 1 3	+	26.75
3 2 1	+	25.57
3 2 2	+	25.54
3 2 3	+	24.75
3 3 1	+	26.46
3 3 2	+	25.85

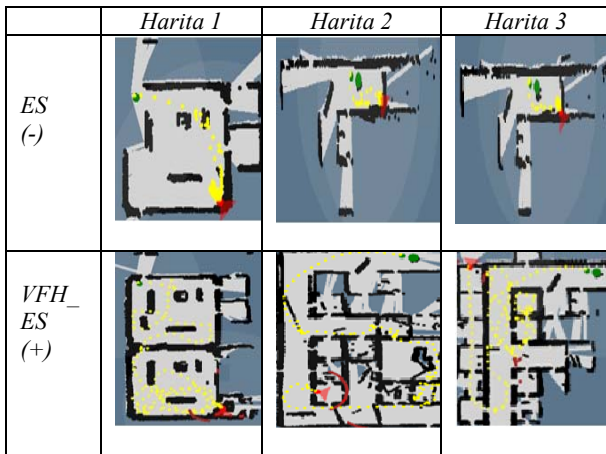
Tablo 4. Parametre katsayılarındaki değişikliğin hedefe ulaşma süresine etkileri

Tablodaki verilere bakıldığında hedefe ulaşmada en az sürenin “3, 2, 3” katsayıları için 24.75 saniye olarak elde edildiği görülmektedir. Bu katsayıları kullanan VFH ile engelden sakınarak otonom gezinme ve engelden sakınmadan otonom gezinme yöntemleriyle 4 farklı harita üzerinde 5 dakika boyunca 3'er kez gezinme yapılmış, toplam gezilen nokta sayısı, engele takılma sayısı bilgileri karşılaştırılmıştır. Tablo ve şekillerde engelden sakınmadan otonom gezinim ‘ES(-)’ ile VFH ile engelden sakınarak otonom gezinim ise ‘VFH_ES(+)’ ile ifade edilmiştir.

Harita no	Yöntem	Takılma sayısı ^{1,2,3}	Gezindiği nokta sayısı ^{1,2,3}
1	ES(-)	32, 30, 31	54, 75, 72
	VFH ES(+)	22, 6, 16	165, 233, 198
2	ES(-)	37, 34, 35	47, 46, 41
	VFH ES(+)	8, 8, 13	167, 212, 212
3	ES(-)	14, 14, 14	131, 142, 139
	VFH ES(+)	5, 26, 17	211, 145, 142
4	ES(-)	29, 20, 31	147, 148, 130
	VFH ES(+)	14, 11, 14	213, 211, 207

Tablo 5. Her iki otonom gezinme yönteminin karşılaştırılması

Yapılan denemeler sonucunda VFH ile engelden sakınarak otonom gezinme yöntemiyle engelden sakınmadan otonom gezinim yönteme göre haritalar içinde robotun daha fazla yeri gezdiği ve daha az sayıda engellere takıldığı gözlemlenmiştir. Engelden sakınmadan otonom gezinim yöntemi ile robotun önüne engel çıkması durumunda engele çarpıp kurtulamadığı ve hareketine devam edemediği gözlenirken, VFH ile engelden sakınarak otonom gezinme yöntemiyle önüne engel çıkması durumunda manevra yaparak engelden kaçtığı görülmüştür. Engelden kaçma sayesinde robot haritalar içinde daha fazla yeri gezebilmekte ve kurban bulma olasılığı engelden sakınmadan otonom gezinme yöntemine göre artmaktadır. Engelden sakınmasız ve VFH ile sakınmalı otonom gezinme yöntemleri ile 3 farklı harita üzerinde elde edilen örnek gezinme sonuçları Şekil 6’da verilmiştir.



Şekil 6. Her iki otonom gezinme ile elde edilen sonuçlar

VI. SONUÇLAR

Bu çalışmamızda USARSim ortamındaki sanal robotlara engelden sakınma özelliğinin kazandırılmasıyla ilgili çalışma yapılmıştır. Bunun için literatürde bulunan VFH yöntemi kullanılmıştır. Bu yöntemde bulunan bazı kritik parametrelerin robotun hareketine olan etkileri incelenmiştir. Engelden sakınmadan ve VFH ile engelden sakınarak otonom gezinme yöntemlerinin 4 farklı harita üzerindeki başarıları incelenmiştir. Engelden sakınmadan otonom gezinme yöntemi ile yaşanan engellere takılma probleminin VFH yöntemiyle büyük ölçüde giderildiği, engel görüldüğünde robotun manevra yaparak rotasını değiştirdiği görülmüştür. Ayrıca VFH yöntemi ile robotun engelden sakınmadan otonom gezinim yöntemine göre harita içinde daha fazla yer gezdiği de görülmüştür. Gelecek çalışmalarda USARSim ortamındaki sanal robotların engelden sakınmalarıyla ilgili yeni yöntemlerin kullanılması hedeflenmektedir.

TEŞEKKÜR

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu TÜBİTAK (EEEAG-113E212) ve Yıldız Teknik Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü'nce (2013-04-01-KAP01) desteklenmiştir.

KAYNAKÇA

- [1] http://iweb.dl.sourceforge.net/sourceforge/usarsim/USARSim-manual_3.1.3.pdf
- [2] Johann Borenstein, “The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots”, EEE Transactions on Robotics and Automation, Vol. 7, No.3, June 1991.
- [3] Richard Rozeboom. “April Robotics Laboratory modules for USARSim”, University of Amsterdam Faculty of Science.
- [4] B. Balaguer, S. Balakirsky, S. Carpin, M. Lewis, C. Scrapper, “USARSim: a validated Simulator for research in robotics and automation”.
- [5] http://en.wikipedia.org/wiki/Vector_Field_Histogram
- [6] <http://web.eecs.utk.edu/~parker/Courses/CS594fall08/Lectures/Oct-21-Obstacle-Avoidance-I.pdf>