Bilimde Modern Yöntemler Sempozyumu-BMYS 2010
14–16 Ekim 2010
Dicle Üniversitesi Kongre Merkezi

Otonom Bir Robotla Statik Ortamda Nesne Etiketleme

A. Alper Kaya[1], Yiğiter Yiğit[1] ve M. Fatih Amasyalı[1]

[1]Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü*, 34349 İstanbul, Türkiye*

ÖZET

Bu çalışmada otonom robot için eş zamanlı çalışan, içinde bulunduğu ortamdaki nesneleri tanıma, ortamın haritasını çıkarma ve bir hedefe giden en kısa yolu bulma sistemleri geliştirilmiştir. Projenin amacı, robotun belli başlangıç ve bitiş noktaları arasındaki nesneler ve duvarlar ile temas etmeden hareket ederken, etrafında gördüğün nesnelerin şekil özelliklerini tanıyıp konum bilgileri ile beraber etiketlenmesini sağlamaktır. Robotun önündeki kameradan aldığı görüntülerdeki nesneleri tanıyabildiği ve hareketlerini otonom değiştirebildiği bir sistem yapılmıştır. Nesne tanıma sistemi iki adımdan oluşmaktadır: Görüntülerden çıkarılan şekil bilgileri ile eğitim modeli oluşturma ve oluşturulan model ile ortamdaki bilinmeyen nesneleri gerçek zamanlı olarak etiketleme. Bu adımlar sonucunda robot, belirlenen nesnelerin konumlarını hesaplayıp, yeni bir rota belirlenmiştir. Elde ettiğimiz sonuçlar tanıma başarısı ve gerçek zamanlı hesaplama ihtiyaçlarını karşılamaktadır.

Anahtar Kelimeler*: Nesne tanıma, JSEG, sınıflandırma, konum belirleme, otonom robot, dinamik A**

Object Labeling in Static Environment by an Autonomous Robot

The aim of this project is to develop an algorithm which can take a robot from A to B using the shortest path possible, recognizing the objects on its way and labeling them. While moving, robot should not collide with any objects. Also, object locations are labeled along with object types. The robot captures images from the camera which is mounted on itself. Robot movement is autonomous. Object recognition system comprises two parts: 1. Extraction of pattern information from images and creation of training set. 2. Real-time detection of objects using the created model. After execution of the steps above, the robot is able to compute object locations and set a new route. Our test results indicate that recognition rate and real-time computing goals are met.

Key Words: *Object recognition, JSEG, classification, positioning, autonomous mobile robot, Dynamic A**

1. GİRİŞ

Bu çalışmada amaç, robotun başlangıç noktasından hedefe çarpmadan gidebilmesi; bunu yaparken kamera ile tespit ettiği nesneleri etiketlemesi ve yerlerini belirlemesidir. Robot hareketi için A* algoritmasından, nesne tanıma için JSEG algoritmasından yararlanılmıştır. İşlenmiş görüntülerden nesnenin çıkarılması ve kameradan nesne konumunun belirlenmesinde kullanılan yöntemler diğer bölümlerde detaylıca anlatılmıştır. Nesnelerin etiketlenmesi aşamasında önceden toplanmış örneklerle oluşturulmuş bir karar ağacı kullanılmıştır.

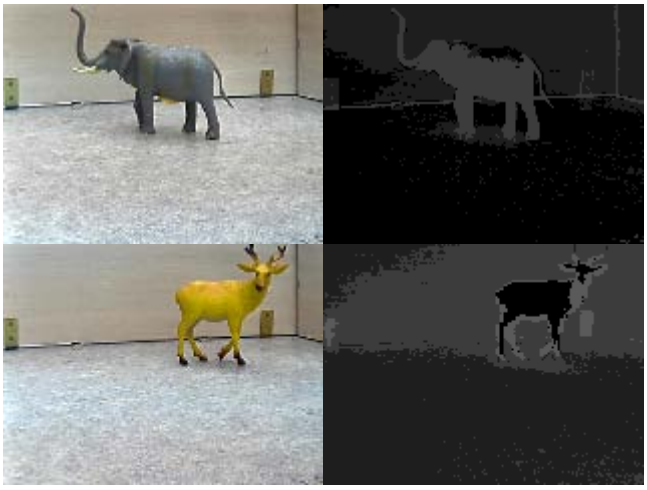
2. GÖRÜNTÜ İŞLEME

Kamera iki amaç için kullanılmıştır: Elde edilen görüntülerin işlenmesi sonucunda nesnelerin tanınması ve robotun otomatik hareketi için bilgi çıkarımı. Bu amaçlar için, kameradan alınan her görüntü bölütlenmiş ve ikili (binary) maskeleri çıkarılmıştır.

2.1. Bölütleme (JSEG)

Görüntüler üzerinde, birkaç homojen renk kümesi kalıncaya kadar bölütleme yapabilen denetimsiz bir metoda ihtiyaç vardır. İşlem hızı beklenenden yavaş olmasına rağmen, sonuçlarının daha güvenilir olması nedeniyle JSEG metodu uygulanmıştır (Deng ve diğ., 2002, [2].

Algoritma renk azaltma (color quantization) ve bölgesel bölütleme olarak iki bağımsız adımdan oluşmaktadır. İlk adımda, renkler, görüntünün kalitesini (komşu bölgeleri ayırt edebilecek kadar uygun) düşürmemesi göz önüne alınarak azaltılmıştır. Ardından yeni renkler için bir renk sınıfı haritası çıkartılmıştır (Şekil 1).



Şekil 1. Renk azaltma işlemi sonrasında oluşturulan renk sınıfları

2.1.1 Bölütleme için Kriterler

Sonraki adım, oluşturulan renk sınıflarını kullanan bir bölütleme kıstası kullanılmasıdır. Dolaştırılan lokal bir filtre ile, J-Değerleri adı verilen, bölgenin içerdiği farklı renk sınıfı sayısına göre elde edilen düşük veya yüksek değerler her piksel için hesaplanmıştır. Bölütleme kriteri, bölütlenen her bölge için bu hesaplanan değerlerin ortalamalarının minimize edilmesidir.

Bütün bir resim üzerinde minimize şartı aranması pratik değildir, resimler çok farklı şekillerde bölütlenebilir. Halbuki, lokal bölgelerde bu şartın kontrolü, bize bölgenin kenar veya iç bölgede bulunduğunu bildirir. Bölgelerin ayrılması da bu şarta göre kontrol edilmiştir. Bakılan lokal pencerelerin büyüklüğü kenar ve iç bölge ayrımının ayrıntısını belirlemektedir. Küçük pencereler, yoğun değişimlerin olduğu bölgelerde kullanışlı iken, büyük pencereler doku sınırlarının belirlenmesinde kullanışlıdır. Genellikle bir resmin doğru bölütlenmesi için birden fazla pencere boyutu gerekmektedir.

Resmin tamamı bir tek bölge kabul edilerek, belirlenen pencere boyutunda bölütleme işlemine başlanmıştır. Ardından pencere boyutu küçültülerek yeni bölütlenmiş bölgeler üzerinde aynı algoritma çalıştırılmıştır. Bu işlem sınır değere ulaşıncaya kadar pencere boyutu küçültülerek devam eder.

2.1.3 Bölge Genişletme ve Birleştirme

Bölge genişletme, başlangıç noktalarının belirlenmesi ve bölgelerin bu noktalardan yayılarak oluşturulması adımlarından oluşur. J-Değerleri belirlenen bir sınır değerinden düşük olan pikseller aday başlangıç noktası olarak işaretlenmiştir. Bu piksellerin 4-komşuluğuna göre aday başlangıç alanları belirlenmiştir. Alanlardan boyutları bir sınır değerden büyük olanlar başlangıç noktası olarak belirlenmiştir. Daha sonra klasik bir bölge genişletme algoritması uygulanmıştır.

Bölge genişletmenin ardından çokça bölütlenmiş bölgelerle karşılaşmak kaçınılmazdır (Şekil 2). Bölgeler, sınır değere ulaşıncaya kadar, renk benzerliklerine göre birleştirilmiştir. Renk azaltma adımında oluşturlan renklere göre, bölgelerin histogramı çıkarılmış ve komşu bölgelerin renk histogramları arasındaki uzaklık hesaplanmıştır. Minimum uzaklıktaki komşu bölgeler birleştirilmiştir (Şekil 3).

Şekil 2. Fil, geyik ve araba nesnelerinin bölütleme sonuçları

Şekil 3. Fazlalık bölgelerin birleştirilme işlemi sonuçları

2.2. Nesnenin Görüntüden Çıkartılması

Bu adımda amaç, bölütlenen bölgelerin sayısını iki veya üçe indirmektir. Genellikle eğer görüntüde bir nesne varsa üç, sadece duvar ve yer bulunuyorsa iki bölge olmaktadır.

Öncelikle, JSEG metodunun birleştirme algoritması uygulanmıştır. Eğer hala 3'den daha fazla bölge varsa, bir alogoritma daha uygulanmıştır. En küçük bölüt bulunmuş ve bu bölgeye en yakın renkli komşusu ile 3 böllge kalıncaya kadar birleştirilmiştir (Şekil 4). Üç bölge olduğu durumlarda, en küçük bölge nesne olarak işaretlenmiştir.

Şekil 3. Üç bölgeye indirilmiş resimler

Bu yöntem, bölgelerin bir takım özelliklerine göre puanlanması ve bu puanın yüksek/düşük olmasına göre seçilmesini öngören algoritmalara kıyasla hatalara daha açık olsa da çok daha

hızlıdır. Testlerimiz, robotun nesnelere dönüş imkanı olmayacak kadar yakınlaşmadığı sürece en küçük bölgenin her zaman nesne olduğunu göstermiştir.

3. NESNE ETİKETLEME

3.1. Karar Ağacının Elde Edilmesi

Başlangıçta projede kullanılacak nesnelerin çeşitleri ve sayılarına karar verilmiştir. Bu aşamada ortamı sınırlayan duvarların yüksekliği ve robotun boyu dikkate alınarak oyuncakların kullanılması uygun bulunmuştur. 2 adet asker, 2 adet araba, otobüs, fil ve geyik olmak üzere 5 farklı çeşitte nesne kullanılmıştır.

Nesne bölütleri, çalışma süresi ve standart bir boyut kullanılması göz önüne alınarak 40x20 olacak şekilde yeniden boyutlandırılmıştır (Şekil 4). Bu bölüt üzerinden istenen özellikler çıkarılmıştır.

Şekil 4. Asker – Bölütlenmiş – Resimden Çıkarılmış – Yeniden Boyutlandırılmış

Çalışılacak ortam hazırlanmış, robot bu ortamda dolaşarak yüzlerce resim toplamıştır. Bu resimlerde nesne olduğu düşünülen bölütler için aşağıdaki özellikler çıkarılmıştır:

1. Nesne en – boy oranı
2. Siyah piksellerin beyaz piksellere oranı
3. Zincir kodu (chain code) uzunluğu
4. Zincir kodu frekansları (8)
5. Zincir kodu normalize edilmiş frekansları (8)
6. 40x20'lik resmin dikey histogramı (20)
7. 40x20'lik resmin yatay histogramı (40)
8. 40x20'lik resmin piksel değerleri (800)
9. 20x20'lik özet matrisinin dikey histogramı (20)
10. 20x20'lik özet matrisinin yatay histogramı (20)
11. 20x20'lik matris değerleri (400)

12. Etiket (Asker, araba, geyik, fil, otobüs, nesne yok) (6)

Bu özellikleri içeren bir ARFF dosyası oluşturulmuş ve Weka aracılığıyla ayırt edici özellikler belirlenmiştir. Belirlenen ayırt edici özelliklerin farklı alt kümelerini içeren ARFF dosyaları hazırlanmış, farklı karar ağacı algoritmaları denenmiş ve başarıları ölçülmüştür (Tablo 1).

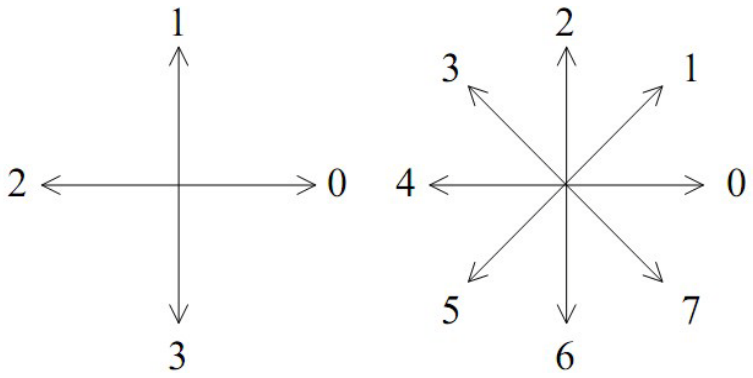
Tablo 1. Weka'ya verilen veri setlerinin başarıları

ARFF İsmi	CART	PART	1NN	Özellik Sayısı
CFS-P1	%66.4	%76.7	%76.1	68
CFS-P2	%65.8	%74.3	%76.1	46
CFS-P3	%67	%76.1	%72.9	41
CFS-P4	%66.8	%72.7	%79.6	33
CFS-P6	%66	%73.1	%79.2	27
CFS	%65.6	%75.4	%77.2	35

1NN (One Nearest Neighbor – En Yakın Komşu) sonuçları referans olarak verilmiştir. Her bölüt için yüzlerce örneğe mesafe hesaplamak zaman alacağından bu yöntem kullanılmamıştır. Başarı oranının mümkün olduğunca yüksek olması gözetilerek CFS-P1 veri seti üzerinde PART algoritmasının oluşturduğu karar ağacının kullanılmasına karar verilmiştir. Elde edilen karar ağacı koda eklenmiş, gelen nesne bölütleri bu karar ağacına göre değerlendirilerek var ise resimdeki nesneler etiketlenmiştir.

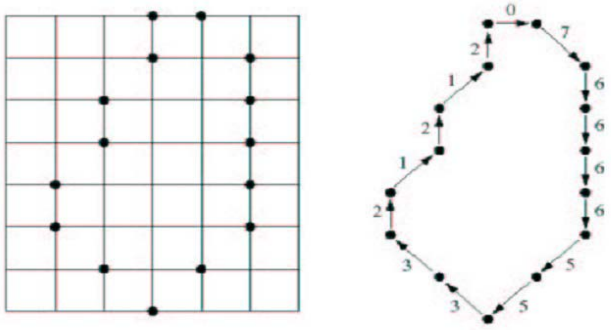
3.2 Zincir Kodu (Chain Code)

Zincir kodu nesne şekillerini dijital olarak ifade etmek için kullanılabilecek özel bir kod yapısıdır (Şekil 5, Şekil 6) [3]. Kapalı bir şekil oluşana kadar bölütün sınırlarını takip eder [4]. Yönler aşağıdaki gibi ifade edilir :



Şekil 5. 4 ve 8 yönlü zincir kod yönleri

Zincir kodu küçültülmemiş resme uygulamak kod uzunluğunu çok artıracağından algoritma 40x20'ye küçültülmüş resim üzerine uygulanmıştır.



Şekil 6. 8-yön ile zincir kodlanmış bir şekil

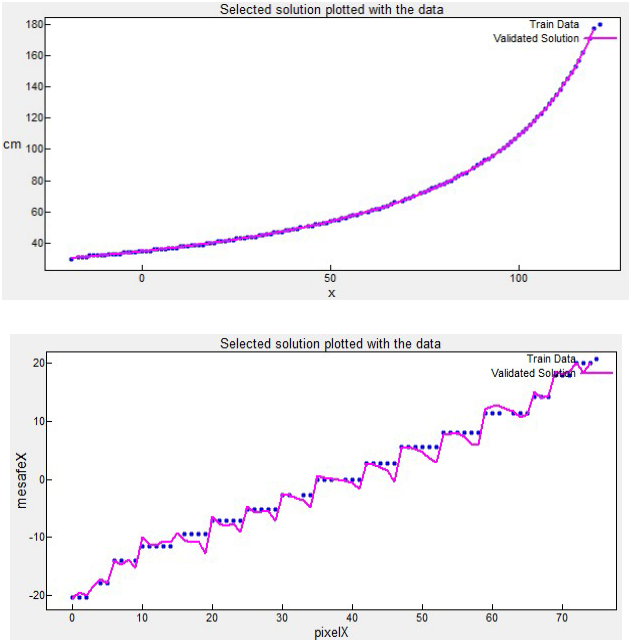
Zincir kodu başlangıç noktasının seçimine göre farklı oluşacaktır. Ayrıca şekle bakış açısı değiştiğinde de kod değişecektir. Bu durumlardan etkilenmeyi en aza indirmek için zincir kod frekansları kullanılmıştır. Doğrudan iki zincir kod karşılaştırılmamıştır. Weka özellik seçicisi zincir kod frekanslarının ayırt edici özellikler olduğunu göstermiştir.

4. ROBOT KONTROLÜ

Robot'un hedefine ulaştığının bilinmesi için, başlangıç konumunun yanı sıra anlık konum değerleri de bilinmelidir. Anlık konumun hesaplanması için gereken değerler servo ve enkoder kullanılarak elde edilmiştir. Doğrusal hareketin hesaplanması için enkoder değerleri yeterli olurken, dairesel hareket için servo değerleri de kullanılmıştır. Bu değerler öncelikle lineer bir hesap ile derece cinsine çevrilmiş, ardından robotun doğrultusu ve yeni konumu kinematik denklemleri ile hesaplanmıştır.

4.1. Engeller ve Robot Arasındaki Mesafenin Ölçümü

Engeller ile robot arasındaki mesafenin ölçümü için, robot üzerinde sadece bir kamera bulunduğu ve görüntü yakalama hızı göz önüne alınarak bir yöntem uygulanmıştır. Öncelikle sadece dikey uzaklık için bir model oluşturulmuştur. Örnek resimler alınmış ve bu resimler üzerindeki piksel değerleri ile gerçek uzaklık arasındaki ilişki bulunmuştur. Ardından yatay uzaklık içinde benzer bir ilişki çıkarılmıştır. Ancak burada hem dikey hem de yatay uzaklık değerleri modele dahil edilmiştir (Şekil 7).



Şekil 7. Mesafe fonksiyonunun grafiği

4.2. Robotun Otonom Hareketi

Robotun otonom hareket edebilmesi için çevresi hakkında bilgiye ihtiyacı vardır. Bu çalışmada, bu bilgiler robotun üzerindeki kamera ile elde edilen görüntülere dayanmaktadır. Labirentteki duvarlar ve objeler, görüntülerin bölütlenmesi sonucunda belirlenmiştir. Alınan her görüntü ve bunların bölütlerinin bir şekilde farklı olabilmesi nedeniyle olasılıksal bir yaklaşım kullanılmıştır.

4.2.1. Olasılıksal Hücreler

Bölütleme nedeniyle oluşabilecek konum hatalarını gidermek için harita hücrelere bölünmüştür. 100 x 100 boyutunda hücreler tanımlanarak meydana gelen küçük sapmaların aynı hücre içinde bulunması sağlanmıştır.

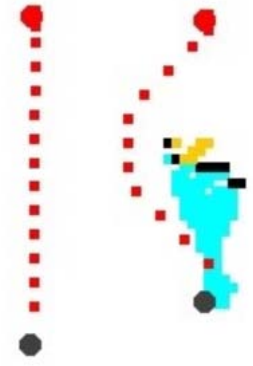
Her hücre duvar ve objeler için ayrı olasılık değerlerine sahiptir. Olasılık değerlerinin belirlenen sınır değerleri aştığı durumlarda, hücre karşılık gelen etiket ile işaretlenmiştir. Ayrıca kameranın gördüğü her bölge "görüldü" sarı etiketi ile işaretlenmiştir. Bu sayede, engellerden kaçınmanın yanı sıra, bilinen bölgeler üzerinden hareket edilmesi de sağlanmıştır (Şekil 8).

Şekil 8. Örnek ortam ve üretilen haritası

4.2.2. En Kısa Yolun Bulunması

Harita üzerindeki 100 x 100'lük hücreler ve bunların içindeki bilgileri (duvar, cisimler, robot, hedef) kullanarak hedefe en kısa yol hesaplanmış ve bir rota belirlenmiştir. En kısa yol hesaplaması için, yüksek performansı nedeniyle Dinamik A* algoritması kullanılmıştır [5]. A* algoritması temelde tüm ortamın bilindiği durumlarda kullanılmaktadır. Ancak bizim problemimizde robot sadece o ana kadar algıladığı ortama ait bilgileri (engel konumlarını) bilmektedir. Bu nedenle t anında hesapladığı yolda ilerlerken t+k anında karşısına bir engel çıktığında (ortam bilgisi değiştiğinde) yeni ortam için hedefe giden yol yeniden hesaplanmaktadır.

Kamera tarafından görüntülenmiş alanlar için daha düşük bir maliyet kullanılarak, robotun bu bölgelere öncelik vermesi sağlanmıştır. Çarpışmaları önlemek için, bir uzaklık tolerans değeri kullanılmıştır. Engellerin çevrelerindeki birkaç hücre de engel olarak işaretlenip; bu sayede A* algoritmasının oluşturduğu rotanın, duvarlara ve cisimlere çok yaklaşmaması sağlanmıştır (Şekil 9).

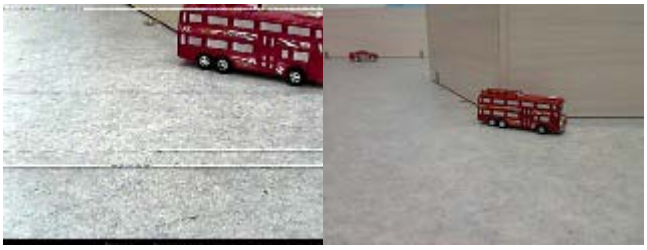


Şekil 9. Dinamik A algoritmasının oluşturduğu rotalar, t anında ortamda engel yokken hesaplanan yol (sol), t+k anında ortamda bir engel görüldüğünde hesaplanan yol (sağ)*

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada bilmediği bir ortamda kendisine verilen bir hedef koordinata doğru otonom hareket ederken gördüğü nesneleri etiketleyen ve ortamın haritasını çıkaran bir robot sistemi geliştirilmiştir. Robot olarak PARS grubu (Yavuz ve diğ., 2006) tarafında tasarlanan robot kullanılmıştır. Robota kontrol işaretleri bir PC üzerinden kablosuz olarak gönderilmekte, robot kamerasından alınana görüntüler, tekerlerin dönüş sayıları ve teker açısı yine kablosuz olarak PC'ye gönderilmektedir.

Sistemin kısıtları incelenecek olursa Şekil 10'da görülebileceği gibi, kullanılan kameranın görüş açısının darlığı rota oluşturma aşamasında bazı hatalı rotalar oluşmasına yol açmıştır. Bu problem geniş açılı bir kamera kullanılmasıyla çözülebilir.



Şekil 10. Robot kamerası (sol) ve cep telefonu (sağ) kamerası ile aynı mesafeden alınan iki görüntü

Ayrıca JSEG algoritmasının çalışma hızı (ortalama 0.5 frame/sn), robotun dönerken yeterince sık görüntü alamamasına ve o bölgenin haritaya yansımamasına neden olmuştur. Bu çalışma farklı bir kamera ve farklı bir resim bölütleme algoritması kullanılarak devam edildiğinde daha başarılı sonuçlar elde edilebileceği düşünülmektedir. Yine gelecek bir çalışma olarak

robotun kamerasından elde edilen ortama ait bilgilerle, robota takılacak mesafe ölçücü bir lazer algılayıcının bilgileri birleştirilerek kullanılabilir.

6. KAYNAKLAR

1. Deng, Y., Manjunath, B.S., (2001), "Unsupervised segmentation of color-texture regions in images and video", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.23, no. 8.
2. UCSB Vision Research Lab - JSEG http://vision.ece.ucsb.edu/segmentation/jseg/
3. University of Calgary Laboratory for Computer Vision – Chain Codes http://pages.cpsc.ucalgary.ca/~parker/soft.htm
4. Umea University – Chain Code Lectures - www8.cs.umu.se/kurser/TDBC30/VT04/material/lect13_kap_11.pdf
5. A* Algorithm – http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html#S3
6. Sırma YAVUZ, M. Fatih AMASYALI, Muhammet BALCILAR, Gökhan BİLGİN,Tarkan DİNÇ, Zeyneb KURT, 2006, "Eş Zamanlı Konum Belirleme ve Harita Oluşturma Amaçlı Otonom Bir Robot", ELECO 2006, Bursa, Türkiye