

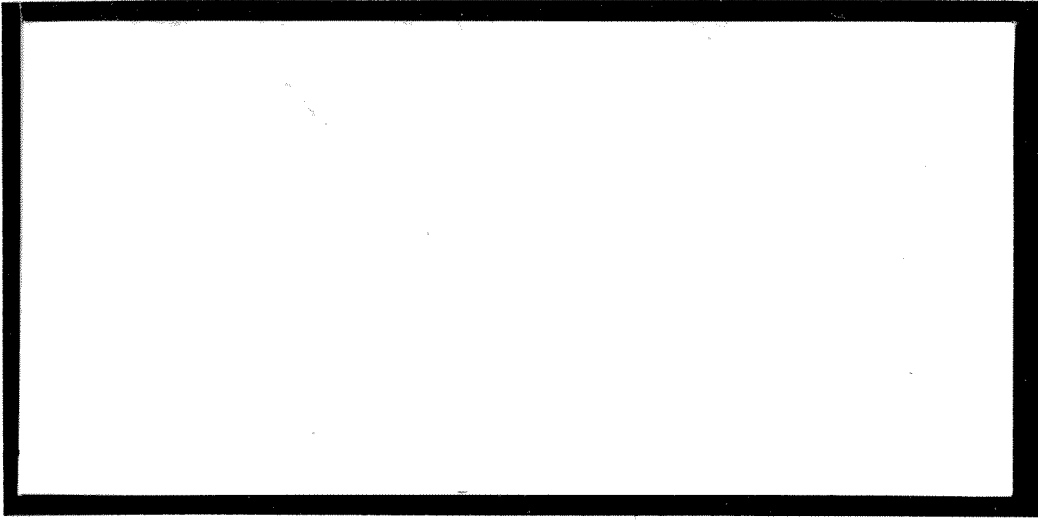
1997 - 35

Dup



TÜRKİYE BİLİMSEL VE
TEKNİK ARAŞTIRMA KURUMU

THE SCIENTIFIC AND TECHNICAL
RESEARCH COUNCIL OF TURKEY



Elektrik, Elektronik ve Enformatik Araştırma Grubu

Electric, Electronics and Informatics Research
Grant Committee

NESNELERİN TANINMASI

PROJE NO: EEEAG-9

191E008

*PROF. DR. METE SEVERCAN
HÜSEYİN UZUNALIOĞLU*

*ŞUBAT 1994
ANKARA*

ÖNSÖZ

Nesnelerin tanınması, konum ve yönlerinin saptanması endüstriyel görüntü işlemede en çok karşılaşılan uygulamalardan birisidir. Nesnelerin hareketli olması durumunda tanıma işleminin süresi önem kazanmaktadır. Bu çalışmada biçimleri önceden bilinen hareketli nesnelerin tanınmasını sağlayan model-tabanlı bir tanıma sistemi geliştirilmiştir. Sistem, tanınması istenen nesnenin herhangi bir görünümünü ve veri tabanında bulunan nesnelerin modellerini kullanarak hızlı bir eşleme işlemini gerçekleştirmektedir.

Bu çalışma TÜBİTAK EEEAG-9, ODTÜ AFP-91-03-01-04 ve TWAS (Third World Academy of Sciences, ICTP, Trieste, İtalya) RGMP 90-092 sayılı projelerle desteklenmiştir. Proje araştırmacılarından Hüseyin Uzunalioğlu'nun Y. Lisans tez çalışması bu proje kapsamında gerçekleştirilmiştir.

İÇİNDEKİLER

ÖNSÖZ	i
TABLO LİSTESİ	iv
ŞEKİL LİSTESİ.....	v
ÖZ	viii
ABSTRACT	ix
BÖLÜM 1: GİRİŞ.....	1
BÖLÜM 2: 3-BOYUTLU MODEL TABANLI NESNE TANIMA.....	5
2.1. Modeller, Nitemler ve Eşleme.....	6
2.1.1. Modeller.....	7
2.1.2. Nitemler.....	8
2.1.3. Eşleme.....	9
2.2. İki Boyutlu Nesne Gösterimi.....	9
2.2.1. Global Nitem Yöntemi.....	10
2.2.2. Yapısal Nitem Yöntemi.....	11
2.2.3. İlişkisel Çizge Yöntemi.....	12
2.3. 2½-D Yüzey Gösterimi.....	12
2.4. Üç Boyutlu Nesne Gösterimi.....	13
2.4.1. Tam Gösterim.....	13
2.4.2. Çokgörünümlü Gösterim.....	14
BÖLÜM 3: ÖNİŞLEME.....	16
3.1. Görüntü Yumuşatma.....	16
3.2. Kenar Sezme.....	18
3.2.1. Global Bir kenar Sezme Yöntemi.....	18
3.2.2. Kenar Sezmede Türevsel Yöntemler.....	19

3.3. Kenar İnceltme.....	22
3.4. Deneysel Sonular.....	22
BÖLÜM 4: ÇİZGİ SEZME ALGORİTMALARI.....	26
4.1. Köşe Noktaları Kullanan Çizgi Sezme.....	26
4.1.1. Algoritma.....	26
4.1.2. Deneysel Sonular.....	28
4.2. Hough Dönüşümü Kullanan Doğru Çizgi Sezici.....	31
4.2.1. Algoritma.....	31
4.2.2. Deneysel Sonular.....	33
4.3. Rotasyon Dönüşümü Kullanan Doğru-Sezici.....	34
4.3.1. Algoritma.....	35
4.3.2. Rotasyon Dönüşümü ve Hough Dönüşümü Arasındaki İlişki.....	37
4.3.3. Deneysel Sonular.....	38
BÖLÜM 5: GEOMETRİK DAGITMA YÖNTEMİ İLE POLİHEDRAL NESNE TANIMA.....	41
5.1. Problem Tanımı.....	41
5.2. 2-Boyutlu Nokta Kümelerin Benzerlik-değişmez Gösterimi	44
5.3. 3-D Nesnelerin Tanınması.....	45
5.3.1. Eğitim Evresi.....	45
5.3.2. Tanıma Evresi.....	47
5.3.3. Eniyi-enküçük Kareler Eşlemesi.....	48
5.3.4. Sağlama.....	52
5.4. Gerçekleştirme.....	54
5.5. Deneysel Sonular.....	58
BÖLÜM 6: SONUÇ.....	69
KAYNAKLAR	73

TABLO LİSTESİ

Tablo 5.1. Oylama sonuçları ve işlem süreleri (Paydalarda gösterilen skorlar en yüksek oy alan yanlış eşlemeler içindir).....	66
---	----

ŞEKİL LİSTESİ

Şekil 2.1. Model-tabanlı bir tanıma sistemi.....	6
Şekil 3.1. İki yumuşatma penceresi (a) Çevre ortalaması, (b) Ağırlıklı çevre ortalaması.....	17
Şekil 3.2. (a) Dikey, (b) Yatay yönde Sobel kernelleri.....	20
Şekil 3.3. Sobel Kernellerinin ayrıştırılması.....	21
Şekil 3.4. Laplacian kenar sezme kerneli.....	22
Şekil 3.5. Kenar sezme ve inceltme (a) Orijinal nesne görüntüsü, (b) Eşiklenmiş kenar noktaları, (c) Inceltilmiş kenar noktaları.....	24
Şekil 3.6. Kenar sezme ve inceltme (a) Orijinal nesne görüntüsü, (b) Eşiklenmiş kenar noktaları, (c) Inceltilmiş kenar noktaları.....	25
Şekil 4.1. Köşeleri kullanan çizgi sezme (Şek. 3.5a için) (a) sezilen köşeler, (b) Her köşedeki çizgi demetleri, (c) Bulunan çizgi çizim.....	29
Şekil 4.2. Köşeleri kullanan çizgi sezme (Şek. 3.6a için) (a) sezilen köşeler, (b) Her köşedeki çizgi demetleri, (c) Bulunan çizgi çizim.....	30
Şekil 4.3. Bir doğrunun (a) görüntü uzayında, (b) parametre uzayında gösterimi.....	32

Şekil 4.4. θ - ρ parametrelerini kullanan doğru gösterimi....	33
Şekil 4.5. Hough dönüşümü ile elde edilen doğru çizgiler: (a) Şekil 3.5 için, (b) Şekil 3.6 için.....	34
Şekil 4.6. Görüntü düzleminin orijin etrafında θ kadar dönderilmesi.....	35
Şekil 4.7. Hough dönüşümü ve rotasyon dönüşümünün parametreleri.....	38
Şekil 4.8. Rotasyon dönüşümü ile bulunan çizgiler. (a) Şekil 3.5 için, (b) 3.6 için.....	39
Şekil 4.9. Sonuç nesne çizimleri (a) Şekil 3.5 için, (b) 3.6 için	40
Şekil 5.1. Bir poligonun bir kenarının belirlediği koordinat sistemi.....	44
Şekil 5.2. Eğitim evresinin akış çizgesi.....	46
Şekil 5.3. Tanıma evresi akış çizgesi.....	49
Şekil 5.4. Saklı çizgilerin yok edilmesini sağlayan programdan örnek çıktılar.....	55
Şekil 5.5. Nesne 1'in orijinal görüntüleri.....	59
Şekil 5.6. Nesne 2'nin orijinal görüntüleri.....	60

Şekil 5.7. Nesne1'in dağıtım tablosu kullanıldığında Nesne1'in birinci görüntüsü için deney sonuçları (a) görüntü, (b) taban vektörleri kullanılarak yapılan eşleme (c) eniyi-enküçük kareler dönüşümünden sonraki eşleme.....	61
Şekil 5.8. A. Nesne1'in dağıtım tablosu kullanıldığında Nesne1'in ikinci görüntüsü için deney sonuçları (a), (b), (c) Şek. 5.7'deki gibi.....	62
Şekil 5.8. B. Nesne1'in dağıtım tablosu kullanıldığında Nesne1'in ikinci görüntüsünün, modelin bir başka görünümü ile eşlenmesi (a), (b), (c) Şek. 5.7'deki gibi.....	63
Şekil 5.9. Nesne2'nin dağıtım tablosu kullanıldığında Nesne2'nin birinci görüntüsü için deney sonuçları (a), (b), (c) Şek. 5.7'deki gibi.....	64
Şekil 5.10. Nesne2'nin dağıtım tablosu kullanıldığında Nesne2'nin ikinci görüntüsü için deney sonuçları (a), (b), (c) Şek. 5.7'deki gibi.....	65

ÖZET

NESNELERİN TANINMASI

Prof.Dr. Mete Severcan ve Hüseyin Uzunaliöđlu
Orta Dođu Teknik Üniversitesi
Elektrik ve Elektronik Mühendisliđi Bölümü

Bu alıřmada, hareketli nesnelere tanınması amacıyla model-tabanlı bir nesne tanıma sistemi geliştirilmiştir. Sistem nitem ıkartma, modelleme ve tanıma aşamalarından oluşmaktadır. Sistem polihedral yapılı nesnelere için tasarlanmıştır. Ancak, kullanılan yöntem daha genel yapılı nesnelere için genelleştirilebilir. Nesne tanımlamaları için doğrusal nitemler kullanılmıştır. Ü adet doğrusal nitem ıkartma yöntemi incelenmiştir: Doğruları bulmak için köşeleri kullanan bir yöntem, klasik Hough dönüşümü yöntemi ve rotasyon dönüşümü yöntemi. Bu yöntemler arasında rotasyon dönüşümü yöntemi kabul edilebilir nitelikte sonuçlar vermiştir. Modelleme ve tanıma işlemleri için geometrik dağıtma (geometric hashing) yöntemi kullanılmıştır. Nesnelere tel-ereve modellerinden başlayarak bir dağıtım tablosu oluşturulmuştur. Aynı yöntem eşleme evresinde nesnenin görüntüsünden elde edilen doğru paralarına uygulanmıştır. Dağıtım tablosundan, bir oylama işlemi sonucunda olası uygun model-nesnelere belirlenmiştir. Daha sonra eniyi-enküçük kareler eşlemesi, ve köşe ve kenar sağlama algoritmaları kullanılarak nesnenin görünümüne uyan model-nesne, konum bilgisi ile birlikte kararlaştırılmıştır. Yöntem gerek nesne görüntüleri ile sınanmış ve başarılı sonuçlar alınmıştır.

Anahtar Sözcükler: Nesne Tanıma, Model-tabanlı Nesne Tanıma, Geometrik Dağıtma.

ABSTRACT
OBJECT RECOGNITION

Prof.Dr. Mete Severcan and Hüseyin Uzunalioglu
Middle East Technical University
Department of Electrical and Electronics Engineering

In this work, a model-based object recognition system is developed to recognize moving objects. The system consists of feature extraction, modeling and matching stages. The system is designed to recognize polyhedral objects. However, the method used can be generalized to other objects. Linear features are used for object descriptions. Three linear features extraction methods are investigated: A method utilizing corner points to detect lines, the classical Hough transform method, and the rotation transform method. Among these methods the rotation transform method gives satisfactory results. Geometric Hashing method is used for modeling and matching processes. Starting with the wire-frame models of the objects a hash table is constructed. The same method is applied to the line segments obtained from the image of the object to be recognized. From the hash table, possible object models are determined using a voting process. Following this, best-least squares matching, and corner and line verification algorithms are used to determine the object model that best matches the view of the object. The method has been tested using realistic images of objects and satisfactory results have been obtained.

Keywords: Object Recognition, Model-based Object Recognition, Geometric Hashing.

BÖLÜM 1

GİRİŞ

Bu çalışmanın amacı, hareketli nesnelerin bir veya birden fazla görüntüsünden elde edilen bilgileri değerlendirerek nesnelerin bilgisayar yardımıyla tanınması, konumlarının ve yönlerinin bulunmasıdır. Üç boyutlu nesnelerin iki boyutlu görüntülerinin kullanılarak tanınması yapay görü' (artificial vision, computer vision) nün önemli konularından birisidir. Üç boyutlu nesne tanıma sistemlerinin, otomatik üretim süreçlerinde, parça muayenesi ve boyut ölçümü gibi, yada robotlarla nesnelerin manipulasyonu gibi uygulamaları vardır.

Bir fabrika ortamında, herhangi bir uygulamada söz konusu nesnelerin türlerinin sayısı genellikle sınırlıdır ve bunlar önceden bilinen nesnelere dir. Bundan dolayı, endüstriyel uygulamalarda, model tabanlı nesne tanıma sistemleri daha kullanışlıdır. Model tabanlı sistemlerde, nesnenin alınan görüntüsünden çıkartılan model, olası tüm nesnelerin modellerini içeren bir veri tabanındaki modellerle eşlenir. Böyle bir sistemin üç temel ögesi vardır. Nitem çıkartma (feature extraction = nitemleme), nesne modelleme, ve eşleme (matching). Sistemin tanıma işleminden önce bir eğitim evresinden geçmesi gerekir. Eğitim evresinde, tanınması istenen her nesneye ilişkin tanımlamalar (descriptions) oluşturulur ve bunlar bir model veri tabanında saklanır. Tanıma evresinde, görüntüdeki nesnenin tanımlamaları bulunur ve veritabanındaki nesne tanımlamalarıyla karşılaştırılır. Sistemin çıktısı hangi

nesnenin tanındığına ilişkin karar bilgisini ve tanınan nesnenin konum ve yön bilgilerini içerir.

Proje başlangıcında, hareketli nesnenin iki veya daha fazla görüntüsünden kenar görüntülerinin elde edilmesi, bu kenar görüntülerinin karşılaştırılması, daha sonra elde edilen nitelemelerle veritabanındaki modellerin eşlenmesi amaçlanmıştır. Kenar sezme işlemi için literatürde insan gözünün kenar sezme işlemini modellediği öne sürülen bir yöntemin (Gupta ve Knopf, 1989) kullanılması düşünülmüştü. Bu yöntem test görüntülere uygulanmış, ancak kabul edilebilir nitelikte kenar görüntüleri elde edilememiştir. Bunun yanında, iki değişik görünümünden sezilen kenar görüntülerinin karşılaştırılması işlemi ile ilgili literatür araştırmasından, bu işlemin endüstriyel nesne tanıma için kullanışsız ve çok zaman alıcı olduğu görülmüştür. Gerçekten de, bu projeye paralel olarak yürütülen bir stereo analiz çalışmasında benzer büyüklükteki kenar görüntülerinin eşleştirilmesi dakikalar mertebesinde olmaktadır. Bu nedenle, projede hareketli nesnenin yalnız bir görüntüsünün tanıma için kullanılmasına karar verilmiştir. Nesne tanıma çalışmalarında iki değişik yaklaşım bulunmaktadır. Bunlardan birisinde eğitim evresinde oldukça basit bir model kullanılmakta ve tanıma evresine ağırlık verilmektedir. İkincisinde ise eğitim evresinde karmaşık bir model kullanılmakta ve tanıma evresi mümkün olduğu kadar kısaltılmaktadır. Bu uygulamada tanıma hızı önemli olduğu için ikinci yaklaşımla ilgili yöntemler araştırılmıştır.

Bu çalışmada, tanınacak nesnelere polihedral yapıları üç boyutlu nesnelere sınırlandırılmıştır. Pek çok uygulama için bu yapı yeterince genel sayılabilir. Ancak yöntem polihedral yapıları olmayan nesnelere de genelleştirilebilir. Polihedral nesnelere için en uygun nitelem nesnenin kenarlarını belirleyen doğru parçaları ve köşeleridir. Bu nedenle bu çalışmada kenarlar ve köşeler nitelem olarak kullanılmıştır. Tanıma işlemi için geometrik dağıtma (geometric hashing) yöntemi (Lamdan ve Wolfson, 1988) kullanılmıştır. Algoritmalar C++ programlama dilinde 80387 yardımcı işlemcili bir 80386 PC (33 MHz) üzerinde geliştirilmiştir. Deneylerde 256x256 piksel büyüklüğünde 256 gri-düzeyle görüntüler kullanılmıştır. Raporda algoritmaların işlem süreleri de verilmiştir. Bu süreler, yazılımlar optimize edilerek düşürülebilir.

Bölüm 2'de, 3- boyutlu nesne tanıma sistemleri tartışılmıştır. Sistemler model tanımlamalarının boyutlarına göre sınıflandırılmıştır. Her grubun özellikleri açıklanmış ve örnekler verilmiştir.

Bölüm 3'de, sistemin ön işleme katı anlatılmaktadır. Ön işleme, görüntü yumuşatma (smoothing) ve kenar sezme işlemlerini içerir. İki değişik kenar sezme yöntemi ele alınmıştır. Ayrıca Sobel kenar sezme işlemi için hızlı bir algoritma da sunulmuştur. Son olarak, kenar piksellerinin oluşturduğu geniş kenar bantları bir kenar inceltme algoritması ile inceltmiştir.

Bölüm 4, nesne kenarlarını belirleyen doğru çizgi parçalarını elde etmek için kullanılan algoritmalara

ayrılmıştır. Bu bölümde, üç değişik çizgi bulma algoritması tartışılmış ve her birisi için uygulama sonuçları verilmiştir.

Bölüm 5'de, 3-boyutlu nesne tanıma amacıyla kullanılan geometrik dağıtma yöntemi tartışılmıştır. Genel kavramlar verildikten sonra, eğitim ve tanıma evreleri açıklanmıştır. Tanıma evresinde, bir oylama işlemi kullanılarak nesneyle eşlenebilecek aday modeller bulunur. Bundan sonra, 3-boyutlu nesne modelleri, daha önce 2-boyutlu nesnelere için kullanılmış olan eniyi-enküçük kareler eşlemesi (Lamdan vd., 1990) ile görüntüdeki nesneye yaklaştırılır. Daha sonra bir sağlama (verification) oylaması ile nesneyle eşlenebilen en uygun aday model, konum ve yönüyle birlikte bulunur. Bu bölüm sonunda değişik nesnelere oluşturulan modeller ve değişik nesne görüntüleri kullanılarak gerçekleştirilen deneyler sunulmuştur.

Bölüm 6, sonuçların tartışmalarını ve ileriki çalışmalar için önerileri içermektedir.

BÖLÜM 2

3-BOYUTLU MODEL TABANLI NESNE TANIMA

Endüstriyel nesnelere tanınması ve konumlarının saptanması yapay görünümün önemli konularından birisidir. Endüstriyel nesne tanıma sistemleri genellikle model tabanlıdır. Bu bölümde model-tabanlı tanıma sistemleri gözden geçirilecek ve bazı örnekler sunulacaktır.

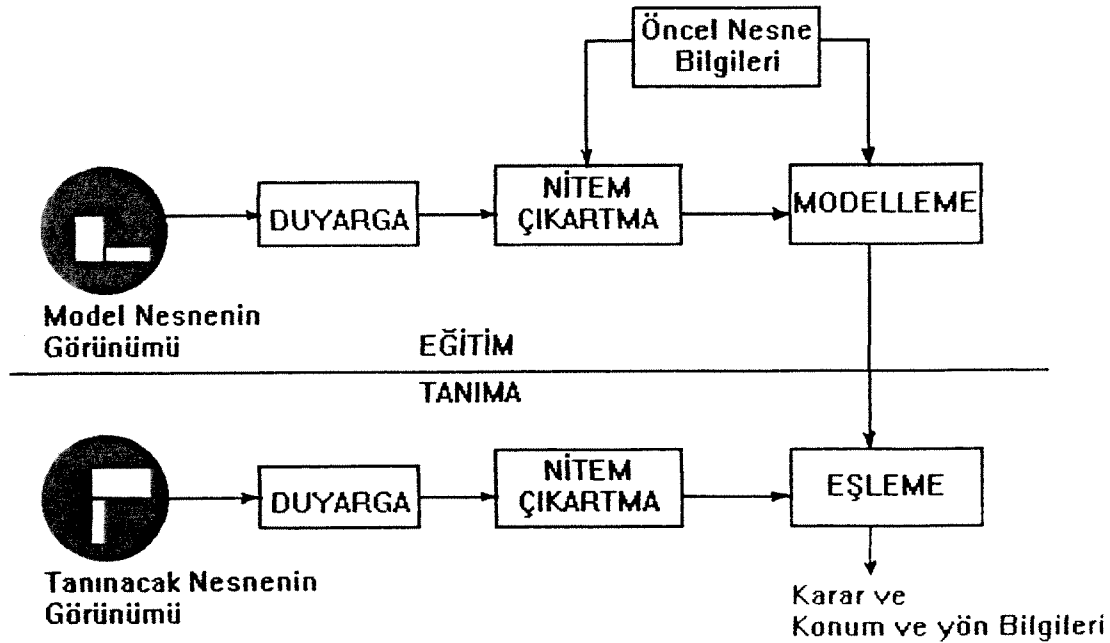
Endüstriyel bir ortamda tanınması istenen nesnelere önceden bilinen nesnelere, bundan dolayı oldukça verimli bir şekilde modellenenlerdir. Model-tabanlı tanıma, bir veri tabanında saklanmış modellerle nesne görüntüsünün eşleştirilmesi işleminden oluşur. Model ve veri tabanı, tanınacak her nesneye ilişkin tüm tanımlamaları içerir ve tanıma işleminden önce hazırlanır. Nesne türüne ve uygulamaya bağlı olarak değişik tanımlama türleri kullanılmaktadır.

Endüstriyel model-tabanlı tanıma sistemleri, nesnelere tanımlama boyutlarına bağlı olarak 2-D, 2½-D ve 3-D olmak üzere üç değişik sınıfa ayrılabilir. Her gruptaki sistemler için farklı nitelendirme (feature extraction), modelleme ve eşleme teknikleri kullanılmaktadır. 2-D nesne tanımlaması izleyici merkezli bir koordinat sistemi kullanılır. Bu yöntem daha çok, nesnenin az sayıda kararlı görüntüsü olduğu durumlar için uygundur. Tanımlamalar ikil veya gri-ton görüntülerden elde edilir. 3-D tanımlamalar nesne-merkezli bir koordinat sistemi kullanır ve izleme noktasından bağımsızdır. Herhangi bir izleme noktasına ait bilgiler bu tanımlamadan hesaplanarak bulunabilir. 2½-D tanımlamalar 2-D ve 3-D tanımlama

özelliklerinin ikisini de kullanır. Tanımlamalar izleyici merkezlidir. Her görünümdeki konum, derinlik, gibi lokal yüzey özellikleri kullanılır. Dolayısıyla her açıdan görünümlerin modellemede kullanılması gerekir.

2.1. Modeller, Nitemler ve Eşleme

Tipik bir model-tabanlı tanıma sistemi Şekil 2.1 de görülmektedir. Sistem eğitim ve tanıma olmak üzere iki evreden oluşur. Sistemin en önemli öğeleri nitemleme, modelleme ve eşleme öğeleridir. Eğitim ve tanıma evrelerindeki sensör ve nitemleme öğeleri değişik olabilir. Aşağıda her öğenin işlevleri daha ayrıntılı tartışılacaktır.



Şekil 2.1. Model-tabanlı bir tanıma sistemi.

2.1.1. Modeller

Modeller genellikle nesnelere geometrik özellikleri kullanılarak kurulur. Her gösterim türü için değişik niteler kullanılmaktadır.

2-D modeller, nesnenin olası 2-D görünümüleri kullanılarak kurulur. Pek çok uygulamada sınırlı sayıda görünüm yeterlidir. Nite olarak nesnenin sınırları veya kenarları kullanılır. Niteleme işlemi, 2-D görünümlemler kullanılarak bulunmaya çalışılan 3-D modellemeye göre daha kolaydır. 2-D modellemenin olumsuz yanı bir boyutun kaybolmasıdır. Modelin yeterliği nesnenin karmaşıklığına ve olası görüş açılarının sayısına bağlıdır. Bazı durumlarda 2-D görüntüler değişik 3-D nesnelere olarak yorumlanabilir.

2½-D modellerde yüzey bilgisi kullanılır. Dolayısıyla 2-D modellere göre daha fazla bilgi taşırlar. Tanımlamalar izleyici merkezlidir. Bundan dolayı her görüş açısı için farklı bir tanımlama gerektirir. Bu gösterim türü için en önemli sorun yüzey bilgilerinin elde edilmesidir.

3-D gösterimi izleyiciden bağımsız oylumsal bir gösterimdir. En genel ve kapsamlı bilgiyi içerir. Bu modeller CAD sistemleri kullanılarak doğrudan elde edilebilirler. Ancak 2-D görüntülerden elde edilen niteler 3-D modellerle doğrudan karşılaştırılmaz. Dolayısıyla 2-D 3-D eşleştirme (correspondence) sorununun çözülmesi gerekir.

Bölüm 2.2, 2.3, 2.4 üç tür nesne tanımlama ile ilgili daha ayrıntılı tartışmaları içermektedir.

2.1.2. Niteimler

Nesne niteimleri kullanılan modele ve uygulama alanına göre farklılık gösterir. Temel niteimler, kenar, köşe, doğru ve eğri çizgiler, delik ve sınır eğriselliği (curvature) dir. Nesne tanımlamaları bu niteimlerin birinden veya birkaçının birleştirilmesinden elde edilir. Gri-ton görüntü bilgilerine göre üst düzey bilgi taşıdıklarından gürültüye karşı daha az duyarlıdır.

Endüstriyel uygulamalarda çoğu zaman nesne sınırları ve bu sınırlardan türetilmiş ölçümler nitem olarak kullanılır. Bu niteimler global, lokal ve ilişkisel (relational) olmak üzere üç gruba ayrılabilir. Global niteimlere örnek olarak çevre, ağırlık merkezi, sınır noktalarının ağırlık merkezine uzaklığı, eğrisellik, alan, eylemsizlik momenti gibi niteimler verilebilir. Doğru parçaları, sabit eğriselliği olan çember parçaları lokal niteimler için örnek verilebilir. İlişkisel niteimlere örnek olarak nesnenin alt parçalarının birbirlerine göre olan uzaklıkları, açıları gibi parametreler verilebilir.

Bazı uygulamalarda kontrollü aydınlanma ve kontrastlı bir arka plan kullanılarak ikil görüntüler elde edilir. Daha sonra bu ikil görüntülerden 2-D niteimler elde edilir. Nesnenin görüntü sınırları içindeki kenar bilgileri kullanılmadığından özellikle 3-D nesnelere için bu yöntemler yetersiz kalmaktadır.

İkil görüntü yerine gri-ton görüntüler kullanılarak daha karmaşık tanıma işlemleri gerçekleştirilebilir. Gri-ton görüntülerden niteimleme daha zor ve daha zaman alıcıdır.

Eşleme her ne kadar 2-D nitelere dayansa bile daha güvenilir sonuçlar verir.

2.1.3. Eşleme

Eşleme işleminin amacı tanınması istenen nesnenin görüntüsünden elde edilen bir dizi nitemi yaklaşık olarak model bir nesnenin bazı niteleri ile eşleştirmektir. Eşleme işlemi kullanılan nesne gösterim türüne göre değişir. 2-D global niteler için genellikle istatistiksel örüntü tanıma yöntemleri kullanılır (Chin ve Dyer, 1986). Lokal nitelerle gösterilen modeller için sintaktik eşleme yöntemleri, ilişkisel nitelerle gösterilen modeller için çizge (graph) eşleme yöntemleri kullanılır. 2½-D modeller için yüzeyler karşılaştırılır. Bu karşılaştırma model yüzeyleri ile görüntü yüzeyleri arasında en iyi çakışan bölgelerin aranması veya yüzeylerden elde edilen bazı nitelerin karşılaştırılması biçiminde yapılabilir. 3-D modellerin kullanıldığı durumlarda eşleme algoritması, 3-D modelin 2-D izdüşüm (projection) lerinden niteleri görüntüden bulunan nitelere en uygun olanını bulmaya çalışır. Eğer tanınacak nesneden üç boyutlu nitem bilgileri ("shape from shading, range, stereo imaging" gibi yöntemlerle) elde edilmişse eşleme işlemi doğrudan 3-boyutlu uzayda yapılır.

2.2. İki Boyutlu Nesne Gösterimi

Nesneler değişik açılardan alınan iki boyutlu görünüşleri ile modellenabilir. Bu görünüşler doğrudan CAD gösteriminden

hesaplanarak da elde edilebilir. Tanıma sistemi iki boyutlu görünümlerden elde edilen modellerden oluşan bir veri tabanı kullanılarak eğitilir. Böylece 3-D nesne tanıma işlemi bir 2-D nesne tanıma işlemine dönüşür. Kullanılan modele ve eşleme sistemine göre 2-D tanıma yöntemleri üç grupta incelenebilir. Global nitem, yapısal ve ilişkisel çizge yöntemleri.

2.2.1. Global Nitem Yöntemi

Bu yöntem çevre uzunluğu, alan, ağırlık merkezi, eylemsizlik momenti ve Fourier tanımlayanları (descriptors) gibi global nitimleri kullanır. Bu nitimlerin elde edilmesi oldukça kolaydır. Nitimlerin sıralanmasıyla bir nitem vektörü oluşturulur. Tanıma işlemi için, bilinmeyen nesnenin nitem vektörü ile model nesnelerin nitem vektörleri istatistiksel örüntü tanıma yöntemleri kullanarak karşılaştırılır. Karar kuralları paralel (Bayes sınıflandırıcı) veya hiyerarşik/ardışıl olabilir.

Nitem vektörlerinin boyutlarının ve sayılarının sınırlı olması ve bir görüntüden kolayca elde edilebilmeleri nedeniyle bu yöntem oldukça hızlıdır. Ayrıca bazı nitimler dönme ve konumdan bağımsızdır. Ancak bu yöntemin sınırlamaları şunlardır.

1. Her olası görüş açısı için ayrı bir model kullanılmalıdır.
2. Nesneler birbirlerine değmemeli veya birbirini örtmemeli ve önemli kusurları olmamalıdır.
3. Bir görüntüdeki tüm nesneleri ayırmak için tek bir eşik değerinin kullanılması gerekmektedir.

"SRI Vision Module" (Gleason ve Agin, 1979) global nitem yöntemi kullanan tanıma sistemi için bir örnek olarak verilebilir.

2.2.2. Yapısal Nitem Yöntemi

Bu yöntemde doğru çizgi, eğri parçası, köşe gibi lokal nitemler sıralı bir liste biçiminde birleştirilerek yapısal bir model oluşturulur. Sıralama o şekilde yapılır ki ardarda birleştirildiğinde nesnenin sınırları çiziliyormuş etkisini yapar. Bu modelleme türünde nesnelere birbirini örtmesinden dolayı bazı nitemler eksik olsa bile tanıma olasılığı bulunmamaktadır. Eşleme hipotez sınama yoluyla gerçekleştirilir. Nitemler nesne görüntüsünden, model kurulurken kullanılan yöntem ile bulunur. Bulunan nitemler eşlenerek modelden türetilen hipotezleri sağlayıp sağlamadığına bakılır. Tanıma işlemi için sintaktik yöntemler de kullanılabilir. Bu yöntemlerde lokal nitemler ilkelere (ilkel: primitive) dönüştürülür. Gramatik kurallar kullanılarak ilkeller tümcelere dönüştürülür. Tanıma gramatik olarak yapılır.

Yapısal nitem yöntemi global nitem yöntemine göre daha üstündür. Ancak, daha karmaşık eğitim ve tanıma işlemleri gerektirir. Gri-ton görüntülerden elde edilen nitemler ikil görüntülerden elde edilen nitemlerden daha güvenilir ve sayıca çok daha fazladır. Bundan dolayı modelleme ve eşlemenin, bütün durumların sınanmasını gerektirmeyecek biçimde tasarlanması gerekir.

Yapısal nitem yöntemi için örnek olarak (Stockman vd. 1982) tarafından geliştirilen yöntem verilebilir. Bu yöntemde model, sınır parçalarını birleştiren bir vektör ve parçanın deliklerini birleştiren doğruları tanımlayan abstrakt vektörlerden oluşur.

2.2.3. İlişkisel Çizge Yöntemi

Bu yöntemde kenar, köşe gibi lokal niteler arasındaki geometrik ilişkiler önem taşır. İlişkisel yapılar bir çizge ile gösterilebilir. Çizgenin her bir düğümü bir lokal niteme karşı gelir. Her düğüme ilgili lokal niteme ilişkin bir dizi özellik iliştilirilir. Tanıma işlemi bir çizge eşleme işlemidir.

Bolles ve Cain (1982) tarafından geliştirilen bir yöntem ilişkisel çizge yöntemi için örnek olarak verilebilir. Bu yöntemde delikler ve köşeler lokal niteler olarak kullanılmıştır. Her nitem için nesnenin ağırlık merkezine olan uzaklığı ve konumu kaydedilir. Eğitim sürecinde her nesneyi diğerinden farklı yapan niteler öbeklenir. Tanıma sürecinde her nesneyi belirleyen öbekleri oluşturan niteler aranır ve eşlemeye çalışılır.

2.3. 2½-D Yüzey Gösterimi

Bu gösterim izleyici merkezli bir koordinat sisteminde bir gösterim türüdür. İki boyutlu gri-ton bilgiye ek olarak her pixeli tanımlayan bazı yüzey özellikleri, örneğin her pixelin derinliği, açısal konumu, rengi, hızı kaydedilir. Birden fazla görünüm gerekiyorsa her biri ayrı ayrı modellenir.

Yüzey bilgilerinin elde edilmesi gri-ton görüntülerin elde edilmesinden çok daha zordur, ultrasonik yöntemler veya özel aydınlatma teknikleri gerektirir.

Oshima ve Shirai (1983) tarafından geliştirilen bir yöntemde yüzey derinlik bilgileri ilişkisel çizge gösterimi içinde kullanılarak bir modelleme yapılmıştır. Eşleme, gözlenen görüntüden elde edilen ilişkisel çizgenin, nesnenin her görünümü için çıkartılmış ilişkisel çizgeleri ile karşılaştırılarak yapılır.

2.4. Üç Boyutlu Nesne Gösterimi

2-D ve 2½-D gösterimleri izleyici merkezlidir ve dolayısıyla her bir görünüm ayrı ayrı modellenmek zorundadır. Nesne karmaşık oldukça model sayısı artar. Bu sorun izleyiciden bağımsız, oylumsal bir gösterim ile çözülebilir. Her nesne için bir tek model kullanılır ve bu model nesnenin tüm görünümünü içerir. 3-D gösterimler iki grupta toplanabilir:

- 1) Tam gösterim (exact representation)
- 2) Çokgörünümlü gösterim (multiview representation)

2.4.1. Tam Gösterim

Tam gösterim türleri nesneyi tüm olarak yüzeyleri ile veya oylum tanımlamasıyla modellerler. Yüzey modelleri nesnenin kenarlarını ve her yüzünün geometrik tanımlamalarını içerir. Oylum modelleri nesneyi genelleştirilmiş silindirler, küpler, küreler ve dikdörtgen blokların birleştirilmesi ile tanımlarlar. Her ne kadar bu gösterim türleri nesneyi tam

olarak modellese de gerçek zaman uygulamalarında bu modelleri kullanmak çok zordur. Bunun nedeni, 2-D görüntüden bulunan nitelerin 3-D modelleme eşlenebilmesi için 2-D den 3-D ye ve 3-D den 2-D ye izdüşüm hesaplamaları yapılması gerekmesindedir.

Bolles vd. (1984) tarafından geliştirilen 3DPO sistemi (three dimensional part orientation system) tam gösterim kullanan sistemlere örnek olarak gösterilebilir.

2.4.2. Çokgörünümlü Gösterim

Bu gösterim türü, nesnenin 2-D (veya 2½-D) gösterimlerini birleştirerek bir tek model oluşturur. Çok görünümlü gösterim için iki yaklaşım kullanılmaktadır. Bunlardan birisinde nesnenin topolojik olarak farklı kararlı görünümleri modellenmektedir. Bu görünümler izleme noktasındaki küçük değişmelerin nesnenin görünen nitelerinin oluşturduğu topolojik yapısında (kenarlar, köşeler gibi) herhangi bir değişiklik yapmayacak şekilde seçilmektedir.

İkinci yaklaşım ayırık gözlem küresi gösterimidir (discrete view sphere representation). Nesne bu yaklaşımda onu çevreleyen bir kürenin değişik noktalarından izlenir. 2-D izleyici merkezli görünümler birleştirilerek tek bir model elde edilir.

Lamdan ve Wolfson (1988) tarafından önerilen geometrik dağıtma (geometric hashing) yöntemi çokgörünümlü nitem gösterimi kullanmaktadır. Eğitim evresinde gözlem küresi üzerindeki noktalardan alınan 2-D görüntülerden ilgi noktaları

(örneğin köşeler) bulunur. Daha sonra her görüntü için, görüntü içindeki iki nokta kullanılarak bir koordinat sistemi oluşturulur ve diğer noktaların bu koordinat sistemindeki koordinatları hesaplanır. Bu koordinatlar bir tablo için indeks olarak kullanılır ve tabloda bu indekse karşı gelen öğeye nesne adı, görünüm numarası, ve koordinat için seçilen baz (iki nokta) kaydedilir. Daha sonra bu işlem diğer nokta çiftleri için tekrarlanır. Son olarak her görünüm için bu işlemler tekrarlanır. Tanıma evresinde görüntüden bulunan ilgi noktalarından ikisi seçilerek bir koordinat sistemi oluşturulur. Eğitim evresindeki gibi diğer noktaların koordinatları hesaplanır. Bu koordinatlar daha önce oluşturulan tablodan nesne adı, görünüm numarası ve baz çiftini okumak için kullanılır. Bir oylama tablosu oluşturularak model görünümünün hangisinin daha çok oy aldığına bakılır. En çok oy alan model nesne eşleme için aday olarak kabul edilir. Daha sonra model ve görüntünün ilgi noktaları karşılaştırılarak eşlemenin doğruluğu sağlanır.

Bu çalışmada kullanılan tanıma yönteminde temel olarak yukarıda özetlenen geometrik dağıtma yöntemi kullanılmıştır. Bu yöntemin üstünlüğü eğitim evresinin uzun zaman almasına karşın önceden yapılması ve tanıma işleminin kısa sürmesidir. Yöntemin ayrıntıları ve uygulama sonuçları Bölüm 5 de sunulacaktır.

BÖLÜM 3

ÖNİŞLEME

Üç boyutlu bir nesnenin değişik yüzeyleri genellikle gri-ton bir görüntüde farklı bölgeleri oluşturur. Dolayısıyla bölge sınırları görüntü analizinde önemli nitelerdir. Bölgeler global veya lokal yöntemlerle bulunabilirler (Gonzalez ve Wintz, 1987). Eşikleme (thresholding), bölge büyütme, bölge parçalama ve birleştirme gibi global yöntemler bir bölge içindeki gri-ton benzerliklerini kullanır. Lokal yöntemlerde ise gri-ton süreksizlikler kullanılır. Kenarlar ışık yoğunluğunda ani değişimlerin olduğu görüntü noktalarıdır. Kenarları doğru bir şekilde sezme için görüntünün önce yumuşatılması (smoothing) gerekir. Yumuşatma bir bölge içinde küçük yoğunluk değişimlerini ortadan kaldırırken bölge sınırlarındaki yoğunluk değişimlerini yerinde bırakır. Bu işleme ek olarak, üst düzey işlemlerin hızını artırmak için sezilen geniş kenar parçalarının inceltilmesi gerekir.

Bu bölümde projede kullanılan yumuşatma, kenar sezme ve inceltme yöntemleri sunulacak ve deneysel sonuçlar verilecektir.

3.1. Görüntü Yumuşatma

Görüntü yumuşatmanın amacı, çoğu zaman görüntüleme sistemi sonucunda eklenen gürültüden dolayı bozulan görüntünün niteliğini artırmaktır. Bu bozulma ışıklandırma koşullarına da bağlı olabilir. Kenar sezme işleminde pixellerin ışık yoğunlukları kullanıldığından görüntüdeki bozulmalar hatalı

kenarların bulunmasına neden olabilir. Uygun bir yumuşatma kenar sezme işleminin başarısını artırabilir.

Uzamsal alçak geçiren süzme, basit bir yumuşatma tekniğidir. Bu yöntemde, görüntü $f(i,j)$ nin, (i,j) çevresindeki pixellerin gri değerlerinin ortalaması alınarak yumuşatılmış görüntü, $g(i,j)$, bulunur. Bu işlem

$$g(i,j) = \frac{\sum_{k,l} w(k,l) f(i-k, j-k)}{\sum_{k,l} w(k,l)}$$

şeklinde yazılabilir. Burada toplamların sınırları w penceresinin büyüklüğüne bağlıdır. Çoğunlukla 3×3 bir pencere yeterli olmaktadır. Eğer w nun bütün elemanları 1 olarak alınırsa bu yumuşatma işlemine çevre ortalaması (neighborhood averaging) denilmektedir. Eğer w nun elemanları değişik değerler alıyorsa bu işleme ağırlıklı çevre ortalaması denilmektedir. Bu çalışmada her iki tür ortalama da kullanılmıştır. Şekil 3.1 de pencere değerleri görülmektedir.

1	1	1
1	1	1
1	1	1

(a)

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

(b)

Şekil 3.1. İki yumuşatma penceresi (a) Çevre ortalaması, (b) Ağırlıklı çevre ortalaması

3.2. Kenar Sezme

3-D yapay görüde, en önemli nitelmlerden birisi "kenar" yada tekbiçimli bölgelerin sınır çizgileridir. Kenar sezme genellikle doğrusal nitelmede ilk adımdır. Kenar pixelleri ışık yoğunluğunun ani değışiklik gösterdiği pixellerdir. Bundan dolayı genellikle lokal türev işlemleri ile kenar sezilir. Aşağıda önce histogram kullanan global bir kenar sezme yöntemi, daha sonra lokal türev işlemleri kullanan kenar sezme işlemleri anlatılacaktır.

3.2.1. Global Bir kenar Sezme Yöntemi

Bu yöntem (Gupta ve Knopf, 1989), lokal yoğunluk süreksizlikleri yerine, bir bölge içindeki gri-değer benzerliklerini kullanır. Yöntem, kenarları görüntüdeki görünürlüklerine (visibility) göre sınıflandırılır. Örneğin, "birincil kenarlar" nesnenin genel hatlarını belirleyen kenarlar olup ilk bakışta farkedilebilir. "İkincil kenarlar" ve diğerleri daha az farkedilen kenarlardır. Bu yöntemin doğal görü sistemlerinde kenar sezme işlemleri olduğu şekilde bir görüş vardır. Sonuç olarak bu yöntemde her pixele hangi sınıf kenara ait olduğuna dair bir kenar algılama değeri atanır.

Yöntemin uygulamasında, görüntü yumuşatıldıktan sonra yoğunluk histogramı elde edilir. Histogramın yanyana iki minimum noktası arasındaki değerlerinin aynı bölgeleri belirleyen pixellere ait olduğu varsayılır. Yanyana iki bölge, birinin pixel değerlerini 0 diğerinin pixel değerlerini 1 yaparak ayırdedilir. Bunun için histogram, minimum değerlerinde

0.5 deęerini alacak, minimumların ortasında ise 0 veya 1 deęerlerini alacak sinüsoidal parçalarla birleştirilerek yeni bir dönüşüm elde edilir. Bundan sonra her pixele bir kenar algılama deęeri atanır. Bunun için o pixelin dönüşümden sonra bulunan deęeri ile yakın çevresindeki pixellerin minimum deęeri arasındaki fark bulunur. Belli bir eşik deęerini aşan kenar algılama deęerleri kenar pixelleri olarak seçilir.

Bu yöntem bazı görüntüler üzerinde sınıanmış, ancak olumlu sonuçlar alınamamıştır. Oldukça fazla yanlış kenar ortaya çıkmıştır. Ayrıca, işlem süresi birkaç dakika tutmuştur. Bundan dolayı kenar sezme işlemlerinde aşağıda sunulan lokal yöntemler kullanılmıştır.

3.2.2. Kenar Sezmede Türevsel Yöntemler

Kenarlar ışık yoğunluęundaki ani deęişimlerin olduęu noktalar olduęundan, bu noktalardaki lokal türevin büyüklüęü yüksektir. Eğer görüntüye bir lokal türev işlemi uygulanırsa kenarları belirginleştirilmiş bir görüntü bulunur. Daha sonra bu görüntü üzerinde bir eşikleme işlemi ile kenarlar elde edilir.

Bir pixeldeki lokal türev, o pixelin çevresindeki pixellerin aęırlıklı ortalamalarının hesaplanması ile bulunur. Literatürde, deęişik amaçlar için kullanılmak üzere çeşitli kenar sezme kernelleri önerilmiştir. Bu çalışmada Sobel süzgeçleri kullanılmıştır. Sobel kernelleri Şekil 3.2 de görülmektedir. Her görüntü pixeli için yatay ve dikey

türevler, f_x ve f_y , ilgili Sobel kernelini kullanarak bulunur. Bir pixelin kenar gücü (edge strength) ve kenar yönü

$$S(x,y) = \sqrt{f_x^2 + f_y^2}$$

ve

$$\Psi(x,y) = \arctan (f_y/f_x)$$

bağıntıları ile hesaplanır. Burada $S(x,y)$ hesaplamasında iki çarpma, bir toplama ve bir karekök alma işlemleri gerekmektedir. Bunun yerine daha az zaman alıcı

$$S(x,y) = |f_x| + |f_y|$$

işlemi kullanılabilir.

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

Şekil 3.2. (a) Dikey, (b) Yatay yönde Sobel kernelleri

Şekil 3.2 de görülen Sobel kernelleri ile kenar bulma için her pixel için 4 çarpım ve 8 toplama işleme gerekmektedir. Çarpım işlemi daha uzun zaman aldığından, doğrudan Sobel kernellerini kullanmak hesaplama açısından pahalı bir yöntemdir. Bunun yerine Şekil 3.3. de görüldüğü gibi Sobel kerneli daha basit iki P ve Q kernellerine ayrıştırılabilir. P ve Q çarpım gerektirmeyen kerneller olduğundan hesaplama açısından daha verimlidir. Sobel süzgecinin bu biçimine hızlı sobel süzgeci denilmektedir.

$$\begin{array}{ccc}
\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} & = & \begin{array}{|c|c|c|} \hline 1 & 1 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & -1 & -1 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & -1 & 0 \\ \hline \end{array} \\
K1 & = & P + Q \\
\begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} & = & \begin{array}{|c|c|c|} \hline 1 & 1 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & -1 & -1 \\ \hline \end{array} - \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & -1 & 0 \\ \hline \end{array} \\
K2 & = & P - Q
\end{array}$$

Şekil 3.3. Sobel Kernellerinin ayrıştırılması

Kenar gücü görüntüsü $S(x,y)$ elde edildikten sonra basit bir pixel eşiklemesi uygulanır. Belirlenmiş bir eşik değerinin üzerindeki pixeller kenar pixeli ve böyle pixellerden elde edilen görüntüye de kenar görüntüsü denilir.

Bu çalışmada kenar sezme işlemi için Sobel dışında Laplacian işleği ve sıfır-geçiş yöntemi (LoG: laplacian of Gaussian) de denenmiştir. Laplacian

$$L(f(x,y)) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

bir ikinci türev işlemi olup, gürültüye karşı daha duyarlıdır. Ayrıca yalnız kenar büyüklüğü verir, yön bilgisi yoktur. Bu işlem Şekil 3.4 de görülen kernel kullanılarak gerçekleştirilebilir. LoG işleğinde de bir bölge içinde küçük yoğunluk değişimleri nedeniyle istenmeyen sıfır geçişleri olabilmektedir. Bu nedenle, kenar pixellerini tanımlamak için

ayrıca ek işlemlerle sıfır-geçiş pixelleri çevresindeki gri-değer değişikliklerini de kontrol etmek gereklidir.

0	1	0
1	-4	1
0	1	0

Şekil 3.4. Laplacian kenar sezme kerneli

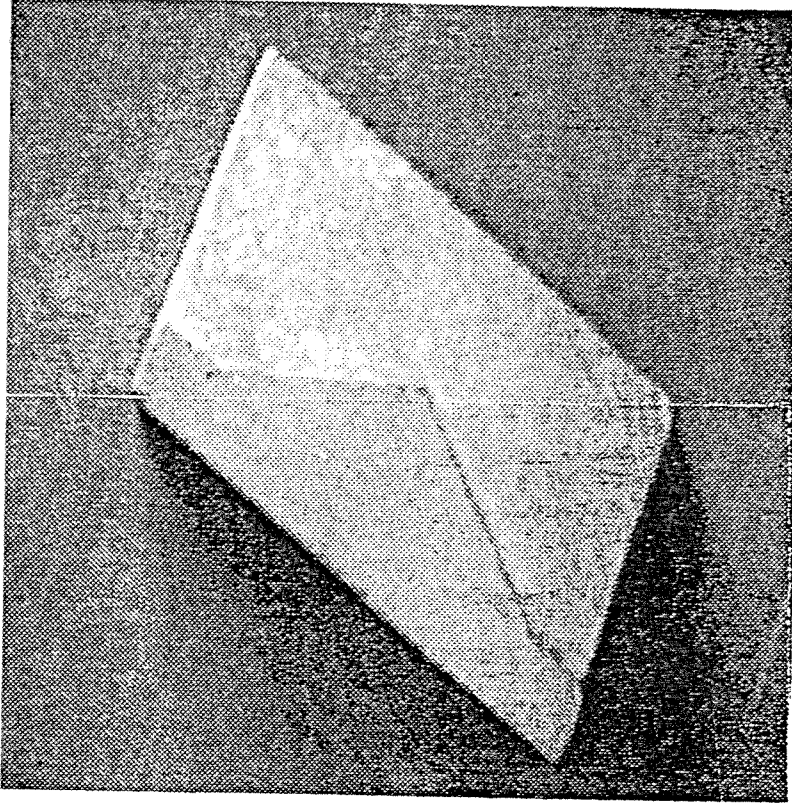
3.3. Kenar inceltme

Sobel işlemleri her pixelde kenar gücünü ve yönünü verir. Basit bir eşikleme işlemi olası kenar pixellerini ortaya çıkartır. Ancak çoğu zaman bu yeterli değildir. Kenar pixellerini içeren geniş bantları inceltmek için maksimum olmayan bir bastırma tekniği kullanılmalıdır. bu teknikte, bir kenar ögesi eğer bir lokal pencere içinde kendi yönüne dik yöndeki komşu elemanların kenar güçlerinden yüksek bir kenar gücüne sahipse kenar ögesi olarak tutulur. Kenar gücü eşik değerinden büyük olsa bile, bu koşulun sağlanmadığı durumda kenar pixeli atılır.

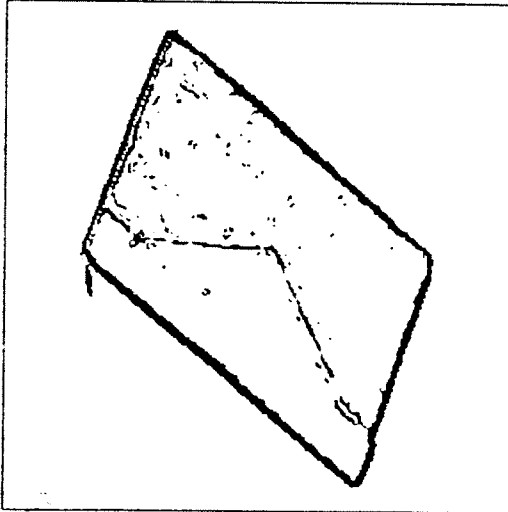
3.4. Deneysel Sonuçlar

Bu çalışmada, önışlemede yumuşatma, Sobel kenar sezme ve kenar inceltme algoritmaları uygulanmıştır. Sobel işleği olarak Bölüm 3.2 de anlatılan hızlı Sobel işleği kullanılmıştır. Yumuşatma için işlem süresi 5 saniye ve kenar sezme ve inceltme için ise bu süre 9 saniyedir. Şekil 3.5 ve 3.6, iki

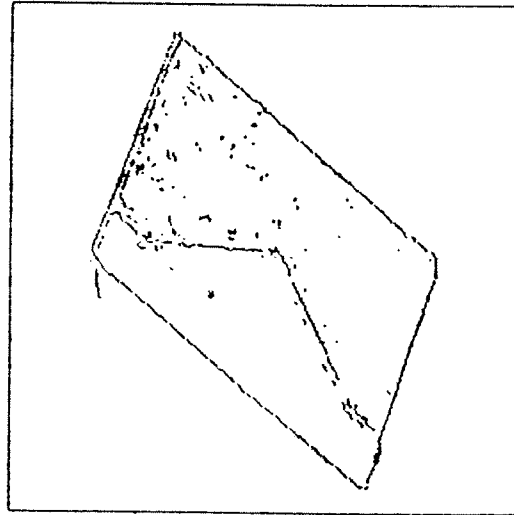
gri-ton görüntü için bu işlemlerin sonucunu göstermektedir. Değişik eşik değerlerine göre sezilen kenarlar değişmektedir. Eğer eşik çok düşük tutulursa bazı hatalı kenarlar görünmektedir. Eğer eşik yüksek tutulursa, bu kez de bazı gerçek kenarlar görünmemektedir. Uygun bir eşik değeri seçimi görüntüye bağlıdır. Aynı koşullar altında alınmış görüntüler için uygun bir eşik değeri deneme ve yanılma yoluyla bulunabilir. Şekil 3.5 ve 3.6 da görülen görüntüler için kenar gücü eşik değeri 100 olarak alınmıştır. Şekil 3.5b ve 3.6b de görüldüğü gibi nesnenin dış sınırlarındaki noktalar geniş kenar alanları oluşturmaktadır. İç kenarlarda ise daha ince kenar alanları bulunmuştur. Bazı gerçek kenarlar kaybolmuş, bunun yanında bazı kenar olmayan noktalar hatalı olarak kenar noktası olarak çıkmıştır. Bu görüntülere maksimum olmayan kenar bastırma tekniği uygulanmıştır. Bundan böyle "kenar görüntüsü" olarak adlandıracağımız Şekil 3.5c ve 3.6c deki sonuç görüntülerde, orijinal nesnede kenar olma olasılığı yüksek bölgelerde kenarlar incetilmiş biçimde görülmektedir. Kenar görüntüleri çizgi sezme algoritmalarının girdileri olarak kullanılacaktır.



(a)

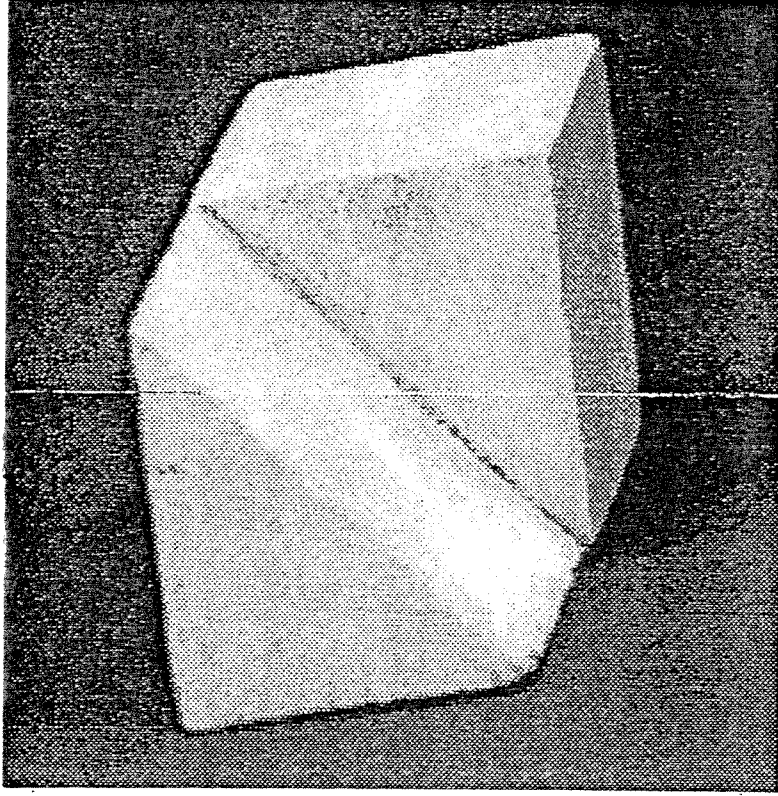


(b)

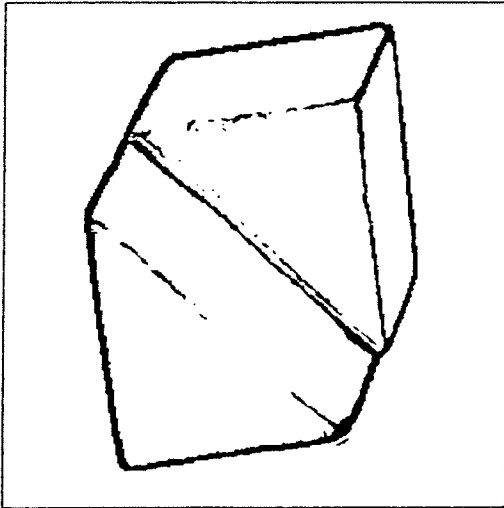


(c)

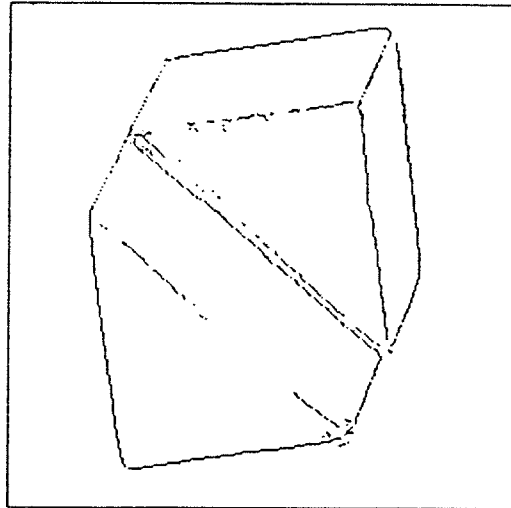
Şekil 3.5. Kenar sezme ve inceltme (a) Orijinal nesne görüntüsü, (b) Eşiklenmiş kenar noktaları, (c) inceltilmiş kenar noktaları



(a)



(b)



(c)

Şekil 3.6. Kenar sezme ve inceltme (a) Orijinal nesne görüntüsü, (b) Eşiklenmiş kenar noktaları, (c) inceltilmiş kenar noktaları

BÖLÜM 4

ÇIZGI SEZME ALGORİTMALARI

Görüntü çözümlemede, özellikle polihedral yapılı nesnelere oluşan görüntüler için doğru çizgi parçaları önemli niteliklerdir. Dolayısıyla doğru çizgi sezimi yapay görüde önemli bir konudur. Doğru parçalarının bulunması yalnız tanıma amacıyla değil, stereo ve dinamik görüntü çözümlemesi gibi alanlarda da kullanılmaktadır. Başarılı bir doğru çizgi sezme algoritması üst-düzyer işlemlerin de basitleştirilmesini sağlar.

Bu bölümde, üç değişik doğru çizgi sezme algoritması tartışılacak ve deneysel sonuçlar verilecektir.

4.1. Köşe Noktaları Kullanan Çizgi Sezme

Bu algoritmada (Gu ve Huang, 1985), doğru çizgi parçaları bir köşe-sezici ile bulunan köşe noktaları kullanılarak bulunur. Köşe noktaları doğruların kesişim noktalarıdır. Dolayısıyla doğru çizgi arama işlemi köşelerin çevresinde yoğunlaştırılabilir. Olası doğru parçaları bulunduktan sonra, bir hatalı doğru ve köşe noktası eliminasyonu algoritması kullanılır. Aşağıda algoritma ve uygulama sonuçları verilmiştir.

4.1.1. Algoritma

Köşeler, kenar yönlerinin büyük değişiklik gösterdiği görüntü noktalarıdır. Fang ve Huang (1982) tarafından geliştirilen bir köşe sezici algoritmada, her noktanın

"köşelik"i (cornerness) hesaplanır ve köşelik değeri bir eşik düzeyini geçen noktalar köşe noktası olarak kabul edilir. Bir noktanın köşeliği, o noktadaki kenar yönü $\Psi(x,y)$ nin gradyanı ile orantılıdır. Bu değer bir köşe noktasında lokal bir maksimuma ulaşır. Köşelik, her görüntü noktasında kenar büyüklüğü $P = (f_x^2 + f_y^2)^{1/2}$ ile kenar yönü $\Psi(x,y)$ nin gradyantınının büyüklüğü $C = (\Psi_x^2 + \Psi_y^2)^{1/2}$ nin çarpımıdır. Burada

$$\Psi(x,y) = \tan^{-1}(f_y/f_x)$$

$$\begin{bmatrix} \Psi_x \\ \Psi_y \end{bmatrix} = \frac{1}{P} \begin{bmatrix} f_{xx} & f_{xy} \\ -f_{xy} & f_{yy} \end{bmatrix} \begin{bmatrix} -f_y \\ f_x \end{bmatrix}$$

ve f_x , f_y , $f(x,y)$ gri-değer görüntüsünün yatay ve dikey gradyantları, f_{xx} , f_{xy} ve f_{yy} ise $f(x,y)$ nin ikinci mertebeden gradyanlarıdır. Bu gradyanların hesaplanmasında Sobel işleçleri kullanılır. P C nin lokal maksimumları bulunur ve belli bir eşik seviyesini geçen değerler köşe noktası olarak kabul edilir.

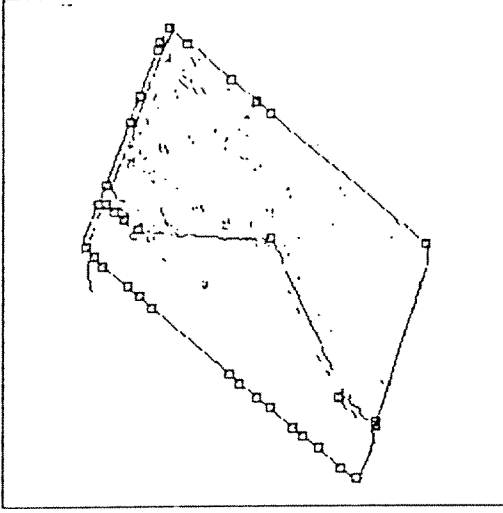
Köşe noktaları bulunduktan sonra, bu köşelerden çıkan doğru parçaları bulunmalıdır. Bu amaçla her köşe noktasının çevresinde lokal bir pencere içinde belli yönlerde yeterli sayıda kenar noktası olup olmadığına bakılır. Her köşe noktasının etrafındaki yönler nicemlenir ve belli açısal aralıklar içine düşen kenar noktaları sayılır. Her köşe çiftini birbirine bağlayan bir çizgi olup olmadığına karar vermek için bir arama algoritması uygulanır. Verilen bir köşe çifti için

eğer köşeler arası uzaklık belli bir eşik değerini aşıyorsa şu koşul uygulanır: Her iki köşeyi bağlayan doğrunun yaptığı açı ile uyumlu her iki köşeden çıkan çizgi varsa bu iki noktayı birbirine bağlayan bir çizgi vardır denir. Eğer köşeler arası uzaklık eşik değerinden küçükse şu koşul uygulanır. Eğer köşeleri bağlayan doğrunun lokal çevresindeki kenar noktaları, köşeler arası uzaklığın belli bir yüzdesini aşıyorsa, iki köşe arasında bir çizgi vardır denir.

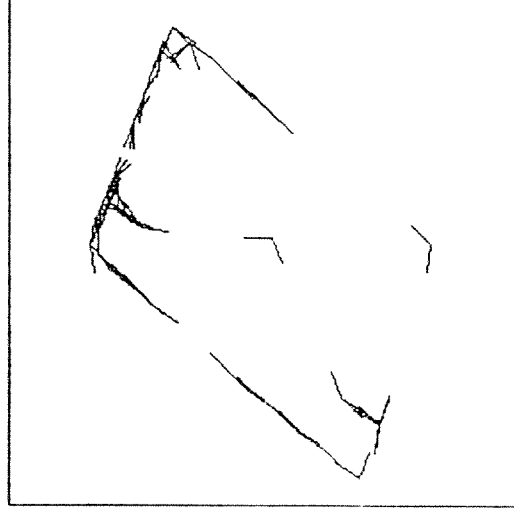
Doğru çizgi arama algoritması sonucunda bir doğrular seti elde edilir. Hatalı köşe ve çizgilerin bu set içinden ayıklanması gerekir. Üç tür köşenin atılması gereklidir: 1) İzole noktalar: Bu noktalardan yayılan çizgi yoktur. 2) Uç noktaları: Yalnız bir doğrunun ulaştığı noktalar. 3) Köşe olmayan noktalar: Köşeden iki çizgi yayılmasına karşın çizgiler arası açı farkı 180° dolayında olan köşeler. Bu köşe ve çizgi ayıklama algoritması, ayıklama tamamlanıncaya kadar yinelemeli olarak uygulanır.

4.1.2. Deneysel Sonuçlar

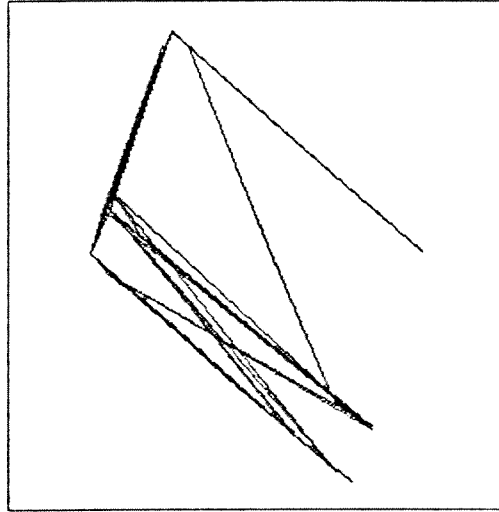
Algoritma Şekil 3.5a ve 3.6a da görülen görüntülere uygulanmıştır. 4 saniye işlem süresi elde edilmiştir. Sonuçlar Şekil 4.1 ve 4.2 de verilmiştir. Şekil 4.1a da küçük kareler, köşe-sezici ile bulunan köşe noktalarını göstermektedir. Eşik seviyesi olarak 6×10^4 alınmıştır. Görüldüğü gibi köşe-sezici pek çok hatalı köşe bulmaktadır. Bunun yanında bazı gerçek köşe noktaları bulunamamıştır. Ayrıca bulunan bazı köşe noktalarının yerleri biraz sapmıştır. Dolayısıyla, köşe-sezicinin başarımı oldukça zayıf kalmaktadır. Şekil 4.1b ve 4.2.b de çizgi derleme



(a)

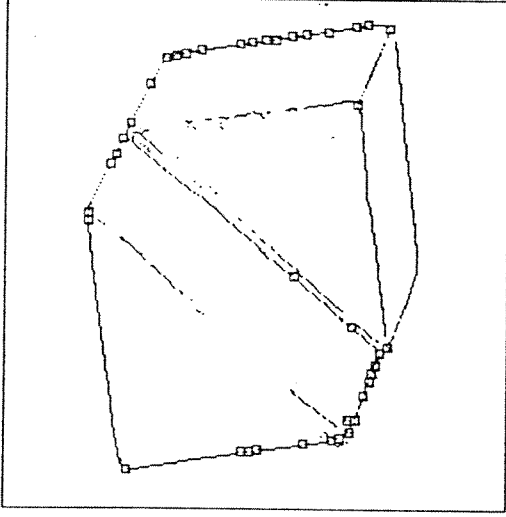


(b)

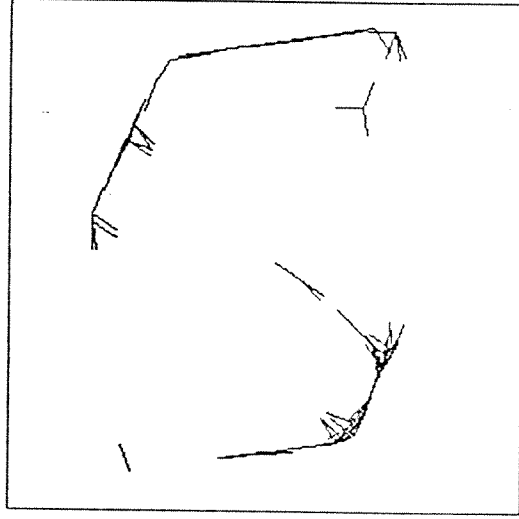


(c)

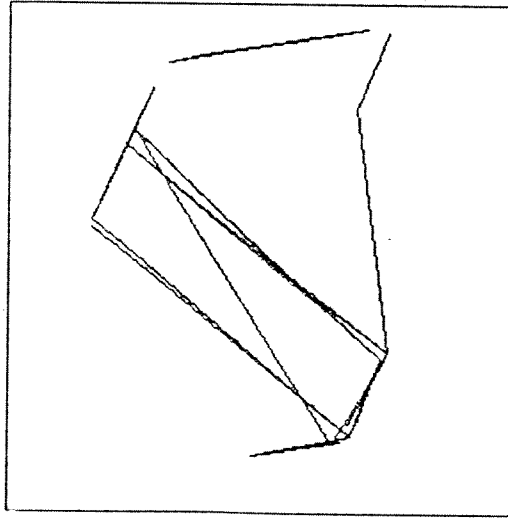
Şekil 4.1. Köşeleri kullanan çizgi sezme (Şek. 3.5a için) (a) Sezilen köşeler, (b) Her köşedeki çizgi demetleri, (c) Bulunan çizgi çizim.



(a)



(b)



(c)

Şekil 4.2. Köşeleri kullanan çizgi sezme (Şek. 3.6a için) (a) Sezilen köşeler, (b) Her köşedeki çizgi demetleri, (c) Bulunan çizgi çizim.

algoritmasının sonuçları görülüyor. Eğer köşe-sezici, köşe noktalarının yerlerini kaydırıyorsa, yön-bulucu yanlış sonuçlar çıkarmaktadır. Şekil 4.1c ve 4.2c de görülen sonuç doğru çizgiler algoritmanın pek başarılı olmadığını göstermektedir. Hatalı köşelerin ve çizgilerin çokluğundan dolayı, hatalı köşe ve doğru eliminasyonu uygulanmamıştır. Bu yöntemin oldukça kontrollü bir biçimde aydınlatılmış ortamlarda kullanılabileceği düşünülmektedir.

4.2. Hough Dönüşümü Kullanan Doğru Çizgi Sezici

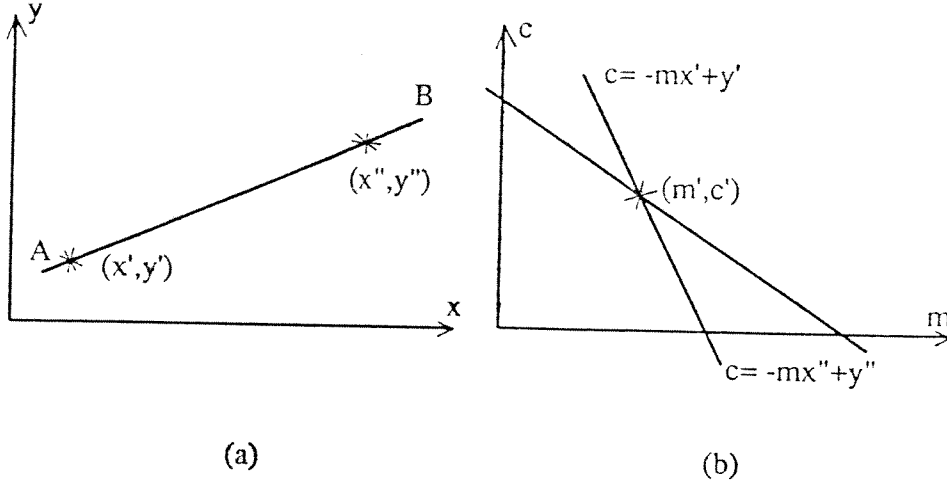
Hough Dönüşümü, verilen bir dizi noktadan geçen bir eğrinin parametrelerini verir. Eğer bu dönüşüm doğru çizgi sezme amacıyla kullanılırsa, dönüşüm doğrunun parametrelerini verir. Bunun için bir oylama işlemi uygulanır.

Şekil 4.3a da herhangi bir (x',y') noktasını ele alalım. Bu noktadan $y'=mx'+c$ bağıntısını sağlayan sonsuz sayıda doğru geçer. Bu koşul (m,c) parametre uzayında $c=-mx'+y'$ gibi bir doğruya karşı gelir. Başka bir (x'',y'') noktası $c=-mx''+y''$ doğrusu ile belirlenir. Bu iki doğru bir (m',c') noktasında kesişir ki bu değerler (x',y') ve (x'',y'') noktalarını birbirine bağlayan doğruyu tanımlayan parametrelerdir. Hough dönüşümü algoritması (Shirai, 1987) görüntü ve parametre uzayı arasındaki bu ilişkiyi doğru-sezici olarak kullanılır.

4.2.1. Algoritma

Önce, parametre uzayı nicemlenir. Nicemlenen uzay bir birikeç matrisi oluşturur. Başlangıçta, matrisin bütün öğeleri

sıfırlanır. Her kenar noktası için o noktadan geçen bütün doğruların parametreleri bulunur ve birikeç matrisinde buna



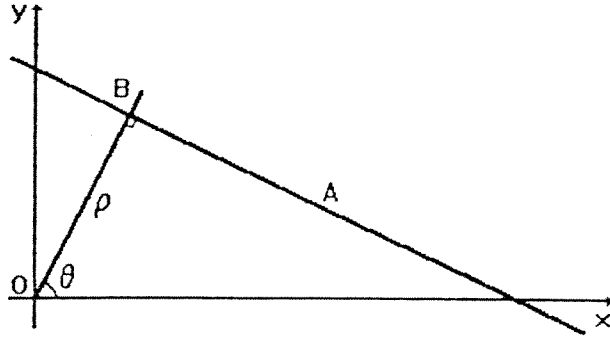
Şekil 4.3. Bir doğrunun (a) görüntü uzayında, (b) parametre uzayında gösterimi

karşı gelen doğru üzerine düşen bütün öğeler 1 artırılır. Aynı doğru üzerindeki noktalar birikeç matrisinde belli bir öğenin değerini sürekli artırır. Birikeç matrisinde oluşan lokal maksimumlar doğru parçalarını belirler. Bu algoritmanın bir dezavantajı m eğiminin veya c kesim noktasının sonsuz olabilmesidir. Bu sorun polar gösterim (açı-yarıçap parametre uzayı) kullanılarak çözülür.

Şekil 4.4 de A doğrusu üzerindeki her nokta

$$x \cos \theta + y \sin \theta = \rho \quad , \quad \theta \in [0, \pi]$$

bağıntısını sağlar. Dolayısıyla A doğrusu üzerindeki bütün noktalar parametre uzayında (θ, ρ) koordinatlarından geçer. Bu

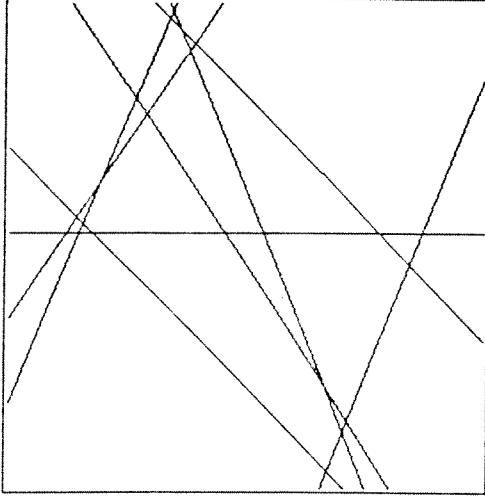


Şekil 4.4. θ - ρ parametrelerini kullanan doğru gösterimi

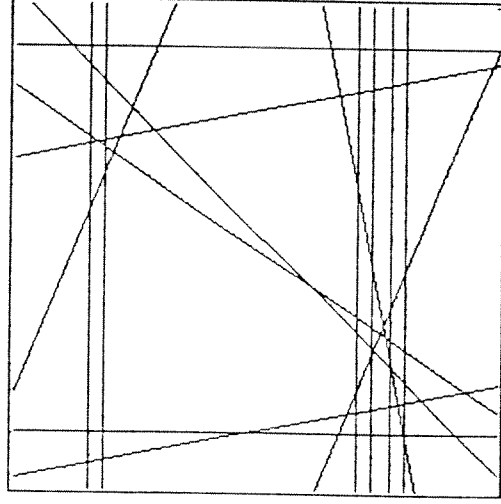
algoritmanın uygulanmasında θ ve ρ değerleri nicemlenir. Hesaplama süresini azaltmak için kenar yön verileri, doğrunun yönünü veren yaklaşık değerler olarak alınabilir (Davies, 1990). Her kenar pixeli (x,y) için bütün (θ, ρ) değerlerini hesaplamak yerine, yalnızca θ nin $(\theta_e - \phi, \theta_e + \phi)$ aralığındaki değerleri için hesaplama yapılır. Burada θ_e kenar yönü, ϕ ise hatalı yön değerlerini göz önünde tutan bir tolerans açısıdır.

4.2.2. Deneysel Sonuçlar

Yukarda anlatılan Hough dönüşüm algoritması doğru çizgi bulma amacıyla gerçekleştirilip sınanmıştır. İşlem süresi kullanılan görüntüler için 8 saniye sürmüştür. Şekil 4.5 de elde edilen doğrular görülmektedir. Lokal maksimumları bulmak için uygun bir eşik değeri bulmak zor olmaktadır. Eşik yüksek seçildiğinde kısa doğru parçaları kaybolmakta, düşük seçildiğinde ise gerçek doğru parçasının etrafında hatalı doğrular ortaya çıkmaktadır.



(a)



(b)

Şekil 4.5. Hough dönüşümü ile elde edilen doğru çizgiler: (a) Şekil 3.5 için, (b) Şekil 3.6 için

Algoritmanın bir başka olumsuz yanı, parametrelerin her doğru parçasını sonsuz bir doğru olarak göstermesidir. Bu durumda, ikinci bir işlemle hatalı doğru parçalarının elimine edilip kalan doğru parçalarının uç noktalarının bulunması gerekmektedir. Bunun dışında, aynı doğru üzerinde fakat farklı doğru parçaları üzerinde olan noktalar hatalı yeni bir doğru oluşturabilmektedir. Bu tür hataları bulmak zor olduğundan daha iyi bir doğru sezici algoritması gerekmektedir.

4.3. Rotasyon Dönüşümü Kullanan Doğru-Sezici

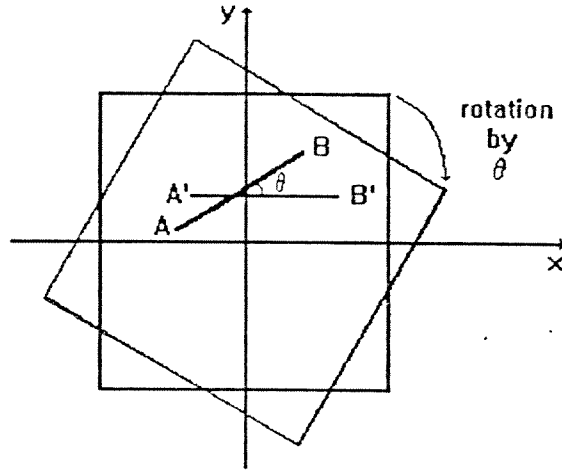
Bu algoritmada (Kang vd. 1991), kenar görüntüsü bir merkez etrafında belli açısal adımlarla döndürülür. Her adımda dikey ve yatay doğru parçaları aranır. Bulunan doğru parçaları, uç noktaları ile birlikte elde edildikten sonra aynı miktarda geri

dönderilerek orijinal görüntüdeki doğru parçaları bulunur. Bu işlem görüntü 90° dödürülünceye kadar devam eder.

4.3.1. Algoritma

Şekil 4.6 da, x-ekseni ile θ açısı yapan bir AB doğru parçası görülmektedir. Eğer görüntü θ açısı kadar çevrilirse AB doğru parçası A'B' doğru parçasına dönüşür ve yatay bir arama işlemi ile bu doğru parçasını ve uç noktalarını belirlemek oldukça kolay olur. AB nin uç noktaları, A'B' nün uç noktalarını $-\theta$ kadar çevirerek bulunur.

Her (x,y) pixelinin θ kadar döndürülmesi ile bulunacak koordinatlar



Şekil 4.6. Görüntü düzleminin orijin etrafında θ kadar dönderilmesi

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$

bağıntıları ile bulunur. Ters dönüşüm için

$$x = x' \cos \theta - y' \sin \theta$$

$$y = x' \sin \theta + y' \cos \theta$$

bağıntıları kullanılır.

Doğru çizgi parçalarını sezme için kullanılan rotasyon dönüşümü algoritması şöyle açıklanabilir.

1) $M \times N$ büyüklüğünde bir görüntü için (bu uygulamada $M=N=256$),
 $(\sqrt{2M}/E) \times (\sqrt{2N}/E)$ büyüklüğünde bir birikeç matrisi oluşturulur.

E burada nicemleme aralığıdır.

2) Her kenar ögesi $e_i(x_i, y_i)$, θ_j kadar çevrilir. Burada $i=1,2,\dots,K$, kenar ögesi sayısı; $j=1,2,\dots,L$ θ nın nicemleme düzeyi sayısıdır. Bulunan (x', y') koordinatları E nin katları olarak nicemlenir. Birikeç matrisinde ilgili ögenin değeri 1 artırılır. Her kenar ögesi için bu işlem tekrarlanır.

3) Bir doğru parçasının uçlarını belirlemek için birikeç matrisi yatay yönde aranır. Arama işlemi için kurallar şöyledir: Eğer matris ögelerinden birisi belli bir eşik değerini (LE: line element) geçiyorsa doğru parçası ögesi olarak kabul edilir. Birbirine SG (separation gap) kadar bir aralıktan daha yakın ögelerin sayısı ML (maximum line length) gibi minimum doğru parçası uzunluğundan daha fazla ise bu ögeler bir doğru parçasını oluşturur. Bu doğrunun ilk ve son ögeleri doğrunun başlangıç ve bitiş uçlarını belirler.

4) Bulunan uç noktaları geri çevrilerek orijinal kenar görüntüsündeki koordinatlar bulunur.

5) 3. ve 4. adımlar dikey yönde yinelenir.

6) Dönme açısı θ bir nicemleme adımı artırılır ve 1 den 5 inci adıma kadar işlemler yinelenir.

Dönme açısı adım büyüklüğü Hough dönüşümündeki gibidir ve dikkatli seçilmelidir.

4.3.2. Rotasyon Dönüşümü ve Hough Dönüşümü Arasındaki İlişki

Rotasyon dönüşümü Hough dönüşümünün genelleştirilmiş biçimidir. İki dönüşüm karşılaştırma amacıyla aşağıda yeniden yazılmıştır.

$$\begin{aligned}\text{Rotasyon dönüşümü: } x' &= x \cos \theta + y \sin \theta \\ y' &= -x \sin \theta + y \cos \theta\end{aligned}$$

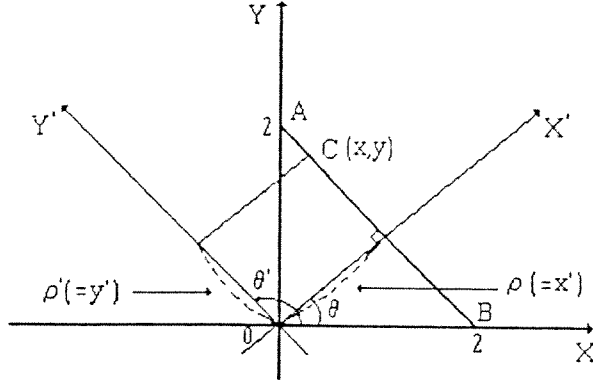
$$\text{Hough dönüşümü: } \rho = x \cos \theta + y \sin \theta$$

x' ve ρ özdeştir. Öte yandan y' , açının $\theta+90^\circ$ olduğu zamanki uzaklığa karşı gelmektedir. Bu uzaklık ρ' olsun

$$\begin{aligned}\rho' &= x \cos(\theta+90^\circ) + y \sin(\theta+90^\circ) \\ &= -x \sin \theta + y \cos \theta \\ &= y'\end{aligned}$$

olduğu kolayca görülür. Şekil 4.7 de görüldüğü gibi ρ' , AB üzerindeki herhangi bir kenar ögesinin yer bilgisini verir. Bu bilgi AB nin uç noktalarını bulmaya yarar. ρ , ρ' ve θ birbirine dik (orthogonal) olup 3 boyutlu genelleştirilmiş Hough uzayını oluşturur. Görüntü düzleminde her nokta bu 3 boyutlu uzayda bir eğriye dönüşür. Hough dönüşümünün, Rotasyon

dönüşümünün $\rho' = 0$ düzlemindeki izdüşümü olduğunu göstermek mümkündür.



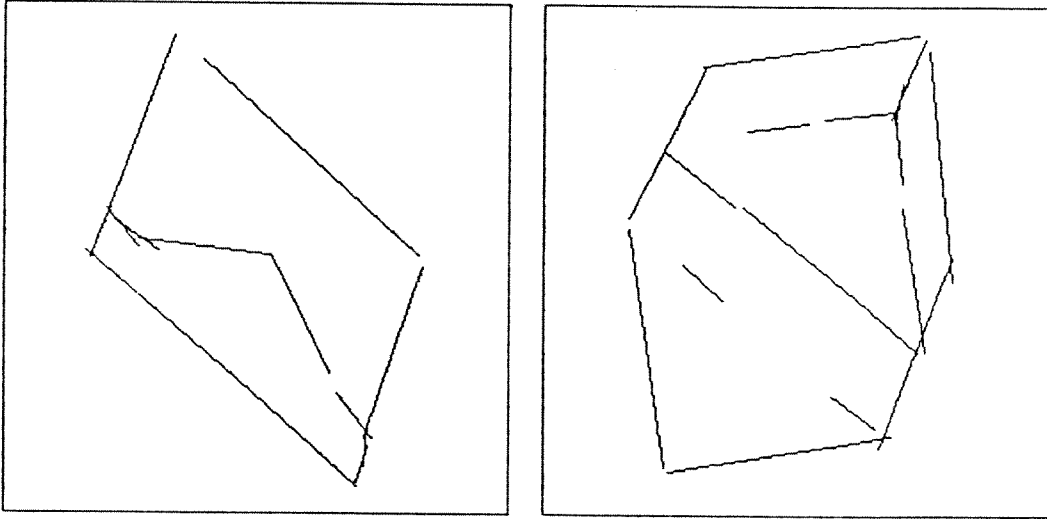
Şekil 4.7. Hough dönüşümü ve rotasyon dönüşümünün parametreleri

4.3.3. Deneysel Sonuçlar

Rotasyon dönüşümü kullanarak doğru çizgi sezme algoritması test görüntüler üzerinde denenmiştir. İşlem süresi 17 saniye sürmektedir. Sonuç görüntüler Şekil 4.8 de görülmektedir. Seçilen parametre değerlerine göre sezilen doğrular değişebilmektedir. LE (minimum çizgi uzunluğu) 7 ile 15 pixel arasında seçilebilir. LE çok küçük olursa yanlış çizgilerin belirlenmesine, büyük olursa kısa çizgelerin kaybolmasına neden olur. Burada LE=10 seçilmiştir. Doğru parçaları elde edildikten sonra ikinci bir işlem ile birbirine çok yakın iki doğru parçası kaynaştırılmaktadır. Aynı çizgi üzerindeki iki doğru parçasının minimum aralığı SG, 1 ile 3 arasında seçilebilir. ρ nun nicemleme adımı $E = 2$, nicemleme adımı $\theta=7.5^\circ$ olarak seçilmiştir.

Bu algoritma kullanılarak kabul edilebilir sonuçlar elde edilmiştir. Parametre değerlerinin seçilebilmesi algoritmaya büyük bir esneklik sağlamaktadır. Bulunan nesne çizimleri hala ideal değildir. Bunun nedeni kenar sezme işleminden dolayıdır.

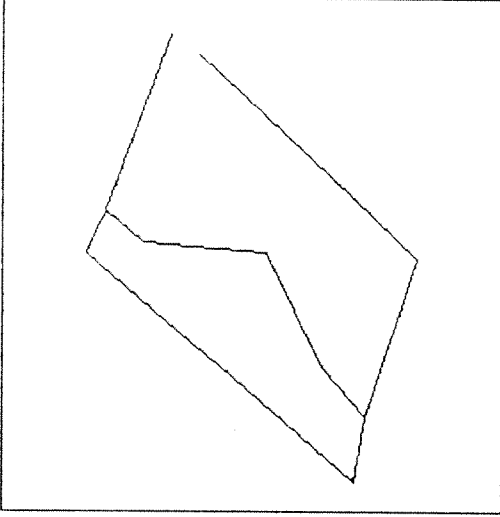
Bundan sonra bir artışem (postprocessing) ile çift çizgilerin kaldırılması, kırık çizgilerin birleştirilmesi ve köşelerin bulunması gerekmektedir. Şekil 4.9 test görüntülerinin bu işlemlerden sonraki durumunu göstermektedir. Bulunan nesne çizimleri tel çerçeve modeline benzemesine rağmen gerçek durumdan farklıdır. Bazı doğru parçaları kaybolmuş, bazı köşelerin koordinatları sapmıştır. Üst düzey işlemlerin bu tür durumlardan etkilenmeyecek biçimde olması gerekmektedir. 5 nci bölümde anlatılan geometrik dağıtma yöntemi bu eksiklikler olmasına rağmen başarılı bir tanıma yapabilmektedir.



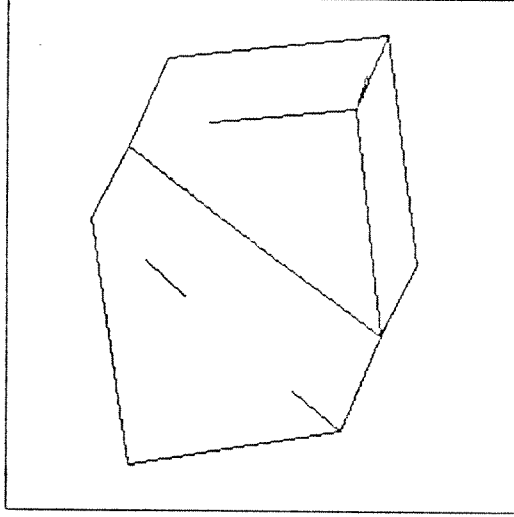
(a)

(b)

Şekil 4.8. Rotasyon dönüşümü ile bulunan çizgiler. (a) Şekil 3.5 için, (b) 3.6 için



(a)



(b)

Şekil 4.9. Sonuç nesne çizimleri (a) Şekil 3.5 için, (b) 3.6 için.

BÖLÜM 5

GEOMETRİK DAGITMA YÖNTEMİ İLE POLİHEDRAL NESNE TANIMA

Geometrik dağıtma (hashing), 3-boyutlu konumu ve yönü bilinmeyen 3-boyutlu nesnelerin 2-boyutlu görüntülerinden tanınmasını sağlayan bir yöntemdir. Bu yöntemde, köşeler ve doğru parçaları gibi lokal niteliler kullanılır. Seçilen nitelilerin geometrik ilişkileri kullanılarak modeller oluşturulur. Eğitim evresinde nesne tanımlamalarından bir dağıtım tablosu (hash table) oluşturulur. Nesneyi çevreleyen bir küre üzerindeki noktalardan alınan 2-D görünümlemler kullanılarak 3-D nesne modellenir. Tanıma evresinde bilinmeyen nesnenin bir tek 2-D görüntüsü ve daha önce hazırlanan dağıtım tablosu kullanılarak, bir oylama işlemi sonucunda nesne tanınması yapılır.

5.1. Problem Tanımı

Amacımız, 3-D bir nesnenin bilinmeyen bir açıdan alınmış 2-D görüntüsünden nesnenin tanınmasını sağlamaktır. Bu sorunu çözmek için önce görüntüleme sisteminin modellenmesi gerekmektedir.

Görüntüleme sistemleri 3-D den 2-D ye dönüşüm sağlayan perspektif izdüşümü gerçekleştirir. Perspektif izdüşümünün parametreleri model ve görüntünün 6 çift noktası kullanılarak hesaplanabilir. Ancak, perspektif izdüşüm parametrelerinin çözümü çok zordur. Çoğunlukla, perspektif izdüşüm yerine paralel (orthographic) izdüşüm kullanılmaktadır. Bu yöntemde izdüşüm yönü görüntü düzlemine diktir. Böylece nesne kameradan

uzaklaştıkça büyüklüğü değişmez. Nesnenin kameraya en yakın noktasının uzaklığı z_{min} , en uzak noktasının uzaklığı z_{max} olursa, perspektif izdüşüm yerine paralel izdüşüm kullanılabilmesi için gerekli koşul

$$z_{max} - z_{min} \ll (z_{max} + z_{min})/2$$

olmalıdır.

Bu uygulamada, görüntüleme dönüşümü bir 2-D yerdeğiştirme (bakış yönüne dik bir düzlemde), bir 3-D dönme ve ölçeklemeden meydana gelir. Nesne tanıma sisteminin amacı bu dönüşümün parametrelerinin bulunmasıdır.

Geometrik dağıtma yöntemi bu sorunu çokgörünümlü bir yaklaşımla çözer. Eğer görüş açısı sabit tutulursa, bir nesnenin iki farklı görüntüsü karşılıklı benzerlik (similarity correspondance) özelliğini taşır. (Biri diğerinden 2-D dönme, yerdeğiştirme ve ölçekleme ile bulunur). Böylece 3-D tanıma problemi 2-D eşleme (matching) problemine dönüşür. Ancak, görüş açısı rastgele olduğundan, eğitim evresi birden çok görünüm kullanır. Görüş açısı bilgisi modele parametre olarak eklenir. Nesneyi çevreleyen bir gözlem küresi üzerinde eşit aralıklarla alınmış noktaların tanımladığı sınırlı sayıda açı, görüş açılarını belirler.

Benzerlik dönüşümü, bir 2-D A matrisi ve bir 2-D yerdeğiştirme vektörü b ile gösterilebilir. Birinci görüntüdeki her x noktası ikinci görüntüde Ax+b noktasına kayar. Böylece sorun model noktalarının görüntü noktaları ile eşleştirilmesi sorununa dönüşür. Bu işlem için bazı özel noktaların bulunması

gerekir. Bu noktaların 2-D dönme, yerdeğiştirme ve ölçekleme altında deęişmez olması gerekir. Bu noktalara ilgi noktaları denir. ilgi noktalarının seçimi model veritabanına baęlıdır. Örneęin, polihedral nesnelere için doğruların uç noktaları ilgi noktaları olarak kullanılabilir. Öte yandan genel nesnelere için keskin dışbükey, derin içbükey veya eğrilięin sıfır olduęu noktalar kullanılabilir.

Benzerlik dönüşümünün parametreleri model üzerindeki bir çift nokta ile nesnedeki bir çift noktanın karşılaştırılması ile bulunur. Her karşılaştırma benzerlik dönüşümü için bir adaydır. Model ve görünümün tüm tanımlamaları kullanılarak adayların birer birer doğrulanması gerekir.

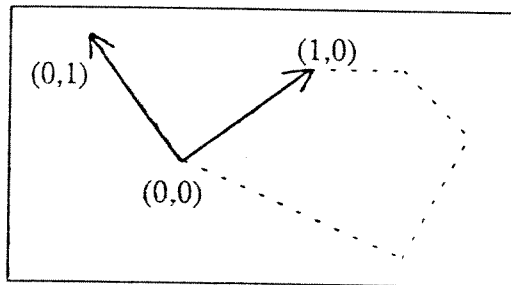
Bu yöntem nesne tanıma için hesaplama açısından çok pahalıdır. Model ilgi noktaları sayısı m , görüntü ilgi noktaları sayısı n ise, en kötü durumda doğrulanması gerekli hipotez sayısı $2\binom{m}{2}\binom{n}{2}$ 'dir. Böylece, tanıma işleminin, en kötü durumda karmaşıklığı $O(m^2n^2t)$ dir. Burada, t modelin görünüme uygunluęunun doğrulanmasının karmaşıklığını ifade etmektedir. Eğer m ve n aynı büyüklükte olursa t de en az n nin büyüklüğü kadar olursa, en kötü durum karmaşıklığı $O(n^5)$ kadar olur. Nesnenin birden fazla görünümü ve birden fazla nesnenin modeli olması gereklilięi göz önüne alınırsa karmaşıklık da o oranda artar.

Geometrik dağıtma yöntemi, algoritmayı bağımsız iki evreye ayırarak tanıma işleminin karmaşıklığını azaltmayı amaçlar. Birinci evre, model noktalarına uygulanan eğitim evresini belirler. Burada görünümle ilgili hiçbir bilgi kullanılmaz.

Dolayısıyla, tanıma işleminden önce "off-line" olarak yapılabilir. İkinci evre ise tanıma evresidir ki burada görüntüden alınan bilgiler ve eğitim evresinde hazırlanan veriler kullanılır. Asıl tanıma süresini bu evrenin işlem süresi belirler.

5.2. 2-Boyutlu Nokta Kümelerinin Benzerlik-değişmez Gösterimi

Tanıma karmaşıklığını azaltmak için, model ve görünüm noktalarının birbirinden bağımsız olarak gösterilmesi ve bu gösterimin nokta çiftlerini karşılaştırmaya uygun olması gerekir. Bu amaçla, sıralı bir nokta çiftinin tanımladığı yeni bir koordinat sistemi kullanılır. Bu nokta çifti x-yönünde birim vektörü tanımlar. Şekil 5.1 deki gibi noktaların birisi $(0,0)$ koordinatlarını, diğeri de $(1,0)$ koordinatlarını belirler. y yönü, bu vektöre saat yönüne ters yönde 90° deki birim vektörle tanımlanır. Bu iki vektör 2-D uzayını tanımlar. Nokta çiftine "taban çifti" (basis pair) denir. Bu yeni koordinat sistemi tanımlandıktan sonra, diğer noktaların koordinatları yeni taban vektörlerini kullanarak bulunur.



Şekil 5.1. Bir poligonun bir kenarının belirlediği koordinat sistemi.

Eğer aynı sıralı nokta çifti, taban çifti olarak kullanılırsa, nokta setlerine uygulanan bir benzerlik dönüşümü noktaların koordinatlarını değiştirmez. Bu şöyle kanıtlanabilir. u ve v seçilen taban çiftinin belirlediği taban vektörleri olsun. $u-v$ düzleminde herhangi bir p noktası

$$p = \alpha.u + \beta.v$$

bağıntısı ile yazılabilir. (α, β) p nin koordinatlarıdır. Eğer nokta kümesine bir T benzerlik dönüşümü uygulanırsa

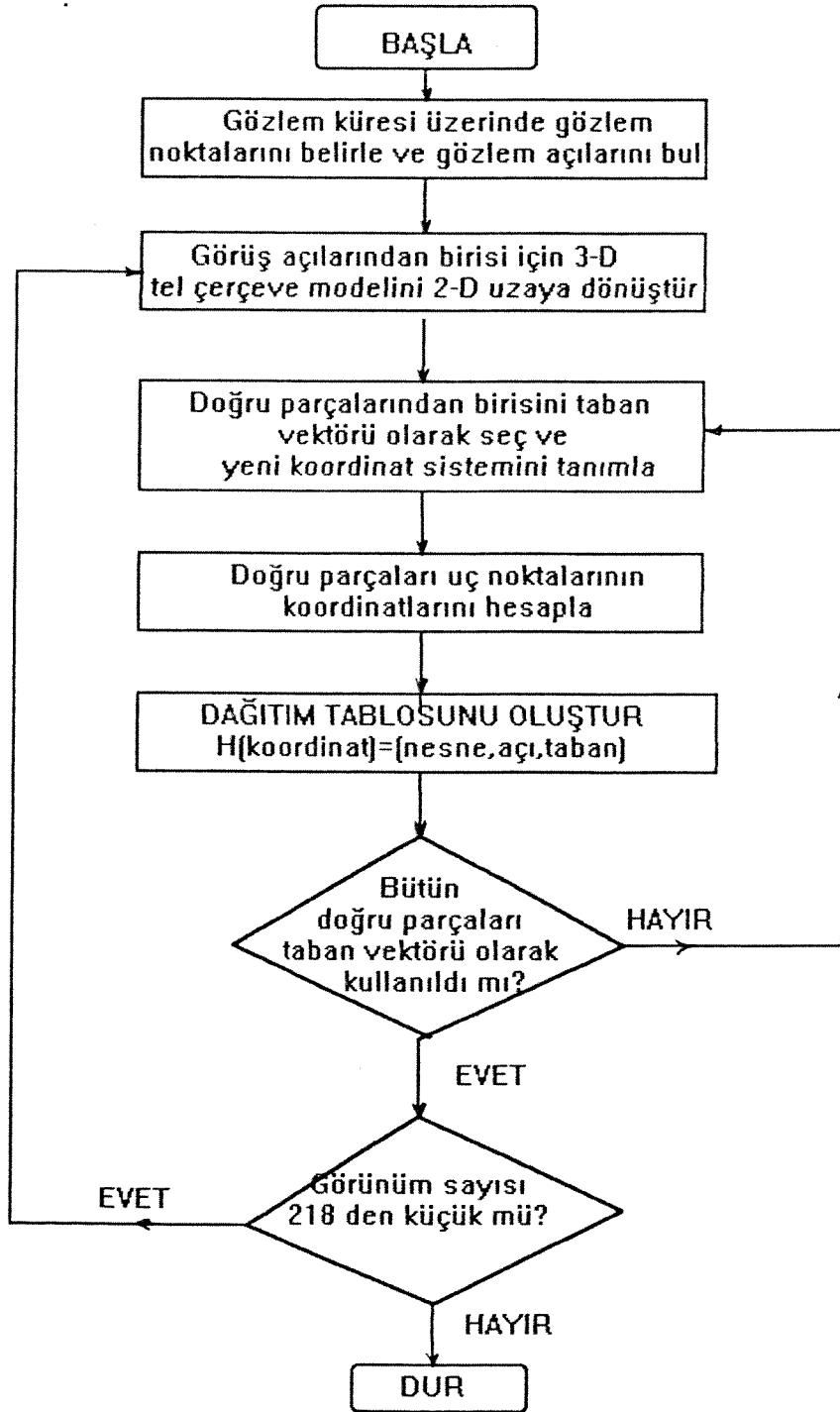
$$T.p = \alpha.Tu + \beta.Tv$$

elde edilir. Eğer aynı sıralı nokta çifti taban çifti olarak alınırsa, yeni taban vektörleri Tu ve Tv olacaktır. Böylece $T.p$ nin yeni koordinat sistemindeki koordinatları yine (α, β) olur.

5.3. 3-D Nesnelerin Tanınması

5.3.1. Eğitim Evresi

Modelin bir görünümünün m ilgi noktası olduğunu varsayalım. Bu noktaların her sıralı çifti için geriye kalan $m-2$ noktanın koordinatları, bu çiftin tanımladığı koordinat sistemi içinde bulunur. Nicemlendikten sonra bu koordinatlar bir dağıtım tablosunun öğelerinin indisini belirler. Her koordinat için (model, görünüm açısı, taban çifti) üçlüsü dağıtım tablosunun belirlenen yerine kaydedilir. Bu evre'nin akış çizgesi Şekil 5.2 de verilmiştir.



Şekil 5.2. Eğitim evresinin akış çizgesi

Bu aşdalı (redundant) gösterim, algoritmaya bir tür güçlülük kazandırır ve tanıma süresini azaltır. Bir tek taban çifti için hesaplanan koordinatlar tanıma için yeterli olmasına rağmen, nesne görüntüsünde bazı ilgi noktalarının örtülü olduğu durumlarda tanıma başarılı olmayabilir. Bu sorunun üstesinden gelmek için nesnedeki olası tüm ilgi noktası çiftleri eğitim evresinde kullanılır.

Eğitim evresinin karmaşıklığı, bir modelin bir görünümü için $O(m^3)$ dür (Lamdan ve Wolfson, 1988). Değişik görünümler ve modeller sabit bir çarpanla bu karmaşıklığı artırır. Veri tabanına eklenecek her yeni model nesne veritabanında daha önce kaydedilmiş modellerden bağımsız olarak işlenir. Eğitim evresi "off-line" olduğundan tanıma süresi etkilenmez.

5.3.2. Tanıma Evresi

Nesne görüntüsünden, ilk olarak, ilgi noktaları bulunur. n ilgi noktası çıkartıldığını varsayalım. Bu noktalardan sıralı bir çifti, taban çifti olarak seçilir ve diğer $n-2$ noktanın koordinatları bu taban kullanılarak bulunur. Böyle hesaplanan her koordinat için (nicemlendikten sonra) dağıtım tablosundaki gözeye bakılır ve burada bulunan her üçlü (model, açı, taban) için o üçlünün oyu bir artırılır.

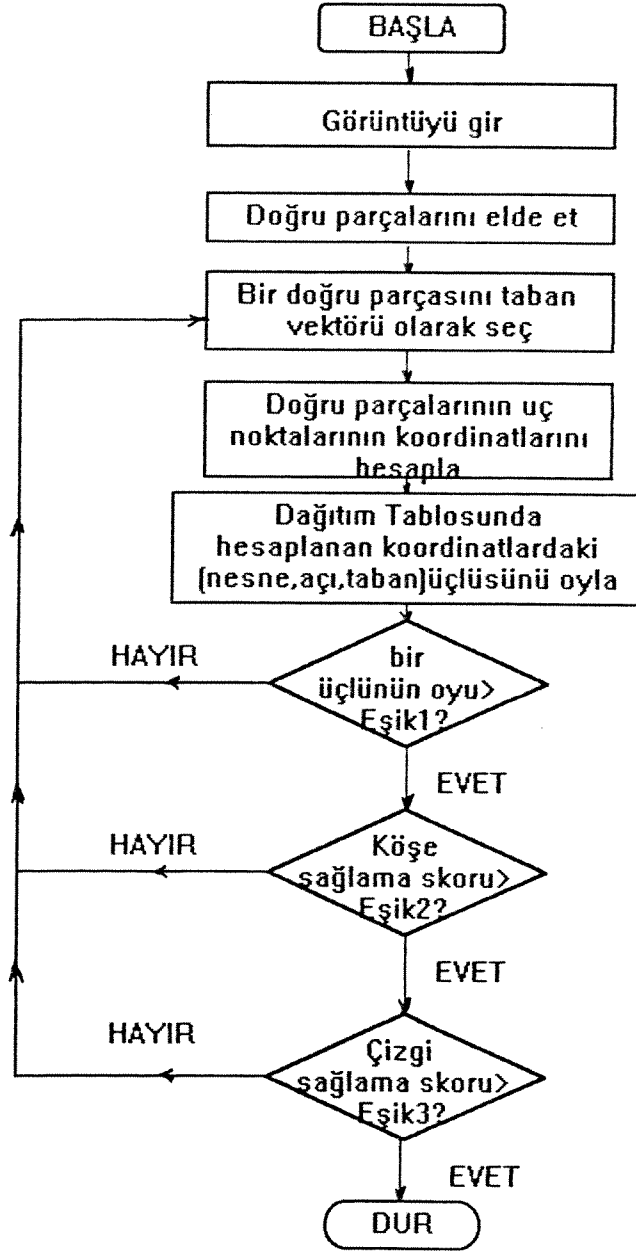
Bu işlemden sonra, eğer üçlülerden birisi yüksek bir oy alırsa, o üçlü eşleme adayı olarak seçilir. Nesnenin kısmen örtülmesi yada hatalı ilgi noktası çıkartılması sonucunda hiç eşleme adayı bulunmayabilir. Eğer bu durum olursa, aynı işlem farklı taban çiftleri kullanılarak tekrarlanır.

Eğer uygun bir eşleme adayı bulunursa, model ve görünümdeki diğer noktalar kullanılarak doğruluğu sağlanır. Model ve görünümdeki birer nokta çiftinin eşleşmesi, modelden görünüme dönüşümü genellikle doğru bir şekilde vermez. Ancak bu, ek bazı nokta ilişkilerini verir. Bunlar kullanılarak, model noktaları görünüm noktalarına eniyi-enküçük kareler kriteri kullanılarak yaklaştırılır. Bundan sonra model köşeleri ve çizgileri görünüm köşeleri ve çizgileri kullanılarak sınanır. Eğer bu sına sonu yeterli bulunursa, varsayımın doğru bir eşleme olduğu kabul edilir. Eğer yeterli bulunmazsa, tüm işlem başka bir taban çifti, kullanılarak yinelenir. Tanıma evresinin akış çizgesi Şekil 5.3 de verilmiştir.

Eğer görünümdeki doğru taban seçilmişse, doğru model yüksek sayıda oy alır. Oylama işleminin karmaşıklığı görünümdeki ilgi noktalarının sayısı ile doğru orantılıdır. Eğer görüntüde bir nesne varsa ve bu nesnenin modeli veritabanında ise, ilk denemede nesne $O(n)$ karmaşıklığı ile tanınacaktır. Eğer veritabanında nesnenin modeli yoksa $O(n^2)$ denemeden sonra nesne reddedilecektir. Böylece, tanıma işleminin en kötü durumda karmaşıklığı $O(n^3)$ olur (Lamdan and Wolfson, 1988). Çoğu durumda, tanıma karmaşıklığı $O(n)$ ile $O(n^3)$ arasındadır.

5.3.3. Eniyi-enküçük Kareler Eşlemesi

Model ve görünüm nokta çiftlerinin ilişkilendirilmesi model-görünüm dönüşümü için bir başlangıç kestirimini verir. Eğitim evresinde CAD benzeri sistemler kullanılmışsa modeller



Şekil 5.3. Tanıma evresi akış çizgesi

gürültüsüdür. Ancak tanıma evresinde elde edilen ilgi noktaları çoğu zaman çok gürültülüdür. Bunun sonucunda model nokta çiftinin görünüm nokta çiftine karşılaştırılması için yapılan dönüşüm çok hatalı sonuç verebilir. Ancak bu dönüşüm ile daha başka nokta ilişkileri bulunabilir. Bu bilgi kullanılarak, eniyi-enküçük kareler hata kriteri ile daha doğru bir dönüşüm elde edilebilir. Aşağıda anlatılan yaklaşım kaynak (15) de düzlemsel nesnelere tanınması amacıyla kullanılmıştır. Bu çalışmada ise bu yöntem üçboyutlu nesnelere için uygulanmıştır.

$(u_i)_{i=1}^n$ ve $(v_i)_{i=1}^n$ birbirleri ile eşlenecek iki nokta dizisi olsun. $(Tu_i)_{i=1}^n$ ve $(v_i)_{i=1}^n$ dizilerinin aralarındaki uzaklığı enküçük yapacak bir T dönüşümünün bulunması amaçlanmaktadır. T dönüşümü

$$Tu_i = Au_i + b \quad i = 1, 2, \dots, n \quad (5.1)$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

bağıntısıyla tanımlanmıştır.

A ve b matrislerinin öğelerinin

$$\delta = \min_T \sum_{i=1}^n |Tu_i - v_i|^2 \quad (5.2)$$

hatasını enküçük yapacak şekilde bulunması amaçlanmaktadır.

Problemin genelliği bozulmadan $(u_i)_{i=1}^n$ noktaları $\sum_{i=1}^n u_i = 0$ koşulunu sağlayacak şekilde kaydırılabilir. Bu durumda

$$\delta = \min_{A,b} \sum_{i=1}^n |Au_i + b - v_i|^2 \quad (5.3)$$

$$= \min_{A,b} \sum_{i=1}^n \left[|b-v_i|^2 + \sum_{i=1}^n |Au_i|^2 + 2 \sum_{i=1}^n b \cdot Au_i - 2 \sum_{i=1}^n Au_i \cdot v_i \right]$$

açılımında

$$\sum_{i=1}^n b \cdot Au_i = b \cdot A \left(\sum_{i=1}^n u_i \right) = 0 \quad (5.4)$$

konulursa, A ve b nin birbirinden bağımsız olarak bulunabileceği görülür. b yi bulmak için

$$\frac{\partial}{\partial b} \left[\sum_{i=1}^n |b-v_i|^2 \right] = \sum_{i=1}^n 2(b-v_i) = 0 \quad (5.5)$$

bağıntısından

$$b = \frac{1}{n} \sum_{i=1}^n v_i \quad (5.6)$$

elde edilir. A yı bulmak için

$$g(A) = \sum_{i=1}^n |Au_i|^2 - 2 \sum_{i=1}^n Au_i \cdot v_i \quad (5.7)$$

tanımını yapalım. g(A) nın minimumunu bulmak için aşağıdaki 4 denklemin çözülmesi gerekir.

$$\frac{\partial g(A)}{\partial a_{ij}} = 0 \quad i=1,2; j=1,2 \quad (5.8)$$

Bu denklem seti, aslında herbiri iki bilinmeyenli iki denklemden oluşan bağımsız iki denklem setidir.

$u_i = (u_i^1, u_i^2)$ ve $v_i = (v_i^1, v_i^2)$, $i=1,2,\dots,n$ için iki boyutlu noktalar olsun. n boyutlu

$$U^j = (u_i^j)_{i=1}^n \quad j=1,2 \quad (5.9)$$

$$V^j = (v_i^j)_{i=1}^n \quad j=1,2$$

gibi 4 yeni vektör tanımlayalım. (5.8) denkleminin çözümü

$$\begin{aligned} a_{11} &= \frac{1}{\Delta} ((U^1 \cdot V^1)(U^2 \cdot V^2) - (U^2 \cdot V^1)(U^1 \cdot U^2)) \\ a_{12} &= \frac{1}{\Delta} ((U^1 \cdot U^1)(U^2 \cdot V^1) - (U^1 \cdot V^1)(U^1 \cdot U^2)) \\ a_{21} &= \frac{1}{\Delta} ((U^1 \cdot V^2)(U^2 \cdot U^2) - (U^1 \cdot U^2)(U^2 \cdot V^2)) \\ a_{22} &= \frac{1}{\Delta} ((U^1 \cdot U^1)(U^2 \cdot V^2) - (U^1 \cdot V^2)(U^1 \cdot U^2)) \end{aligned} \quad (5.10)$$

denklemleri ile bulunur. Burada

$$\Delta = (U^1 \cdot U^1)(U^2 \cdot U^2) - (U^1 \cdot U^2)(U^1 \cdot U^2) \quad (5.11)$$

olup (.) skalar çarpımını göstermektedir.

5.3.4. Sağlama

Tanıma evresinde, oylama işlemi sonucunda belli bir eşik düzeyinden fazla oy alan (model, görünüm açısı, taban çifti) üçlüsü olası bir eşleme için aday seçilir. Eniyi-enküçük kareler algoritması dönüşümdeki hatayı azaltır. Son olarak, model ve görünüm arasındaki aday dönüşümün doğruluğunun sağlanması gereklidir. Bu sağlama, modelin köşe ve kenarları

ile görünümün köşe ve kenarlarının karşılaştırılması ile sağlanır. Gerçekleştirme iki aşamada yapılmıştır. Birinci aşama göreceli olarak basit ve hızlıdır. Pekçok hatalı eşleme bu aşamada elimine edilir. İkinci aşama daha hassas ve daha yavaştır. Birinci aşamada görünümde elde edilen doğru parçalarının uç noktaları model köşeleri ile karşılaştırılır. Eğer yeterince çok sayıda doğru eşleme olursa, model bu sağlama adımını geçer. Bir model köşe noktasının, görünümdeki bir doğru parçasının uç noktasına eşlenmesi için aralarındaki uzaklığın yeterince küçük olması gereklidir. Kullanılan eşik değerleri bundan sonraki bölümde verilmiştir.

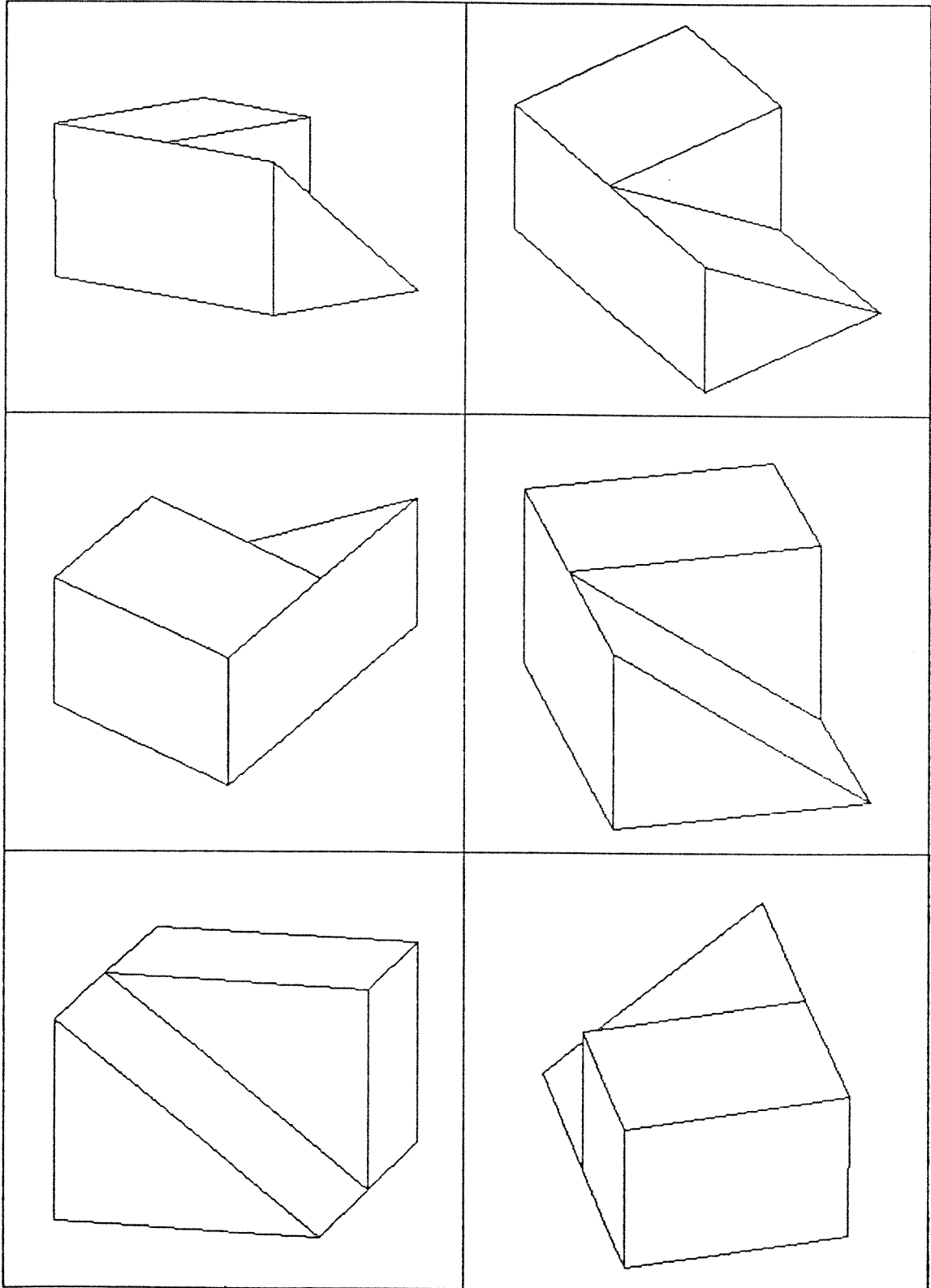
İkinci sağlama aşamasında görünümdeki doğru parçaları model doğru parçaları ile karşılaştırılır. Bu aşama daha çok zaman alıcıdır, ancak yanlış bir eşleme yapılması olasılığını azaltır. Bunun için model ve görünüm çizgilerinin uzamsal ilişkilerine bağlı olarak bir sağlama skoru artırılır veya eksiltir. Eğer dönüşümü yapılmış bir model çizgisi bir görünüm çizgisine yeterince yakın ve paralelse, ve bu iki çizginin uç noktaları birbirine yeterince yakınsa, sağlama skoru 2 artırılır. Eğer bir model çizgisinin yalnızca bir uç noktası, görünüm çizgisinin bir uç noktasına yakınsa skor 1 artırılır. Bütün model çizgileri bu biçimde sınanır. Daha sonra eşlenmemiş her görünüm çizgisi için skor 2 eksiltir ve eşlenmemiş her model çizgisi için skor 1 eksiltir.

Eğer sağlama skoru belli bir eşik düzeyini aşarsa model ve görünümün doğru biçimde eşlendiğine karar verilir ve tanıma işlemi bitirilir. Eğer skor küçük olursa, algoritma oylama işlemine yeniden döner.

5.4. Gerçekleştirme

Polihedral nesnelere tanınması amacıyla dağıtım tablosu yönetimini kullanan bir veritabanı oluşturulmuş ve gerçek nesnelere alınan görüntülerle tanıma algoritması sınanmıştır. Yönlendirilmiş doğru parçaları taban vektörleri olarak kullanılmıştır. Böylece, eğer kenar çizgisi sayısı n ise, $2n$ taban vektörü dağıtım tablosu oluşturulmasında ve tanıma işinde kullanılmıştır. Eğer köşe sayısı n dolayında olursa, eğitim evresinin karmaşıklığı $O(n^2)$ dir. En kötü durum tanıma işlemi karmaşıklığı da $O(n^2)$ dir. Bu ise, doğru parçası bilgisi kullanmak ile eğitim ve tanıma evresi karmaşıklığının n gibi bir çarpanla azalması demektir.

Eğitim evresinde, nesnenin tel çerçeve modelinin değişik açılardan görünümü gerekmektedir. Verilen bir görüş açısı için 3-D den 2-D ye dönüşüm parametreleri bulunmuş, daha sonra saklı çizgilerin yok edilmesi (hidden line elimination) için bir algoritma uygulanmıştır. Bunun için kaynak (Ammeraal, 1986) da verilen bir algoritma bazı değişikliklerle kullanılmıştır. Şekil 5.4 de bazı örnek program çıktıları görülmektedir. Görüş açılarını bulmak için nesneyi içine alan bir küre, bir küp ile kaplanmıştır. Küpün her yüzeyine 6x6 grid çizilmiş ve gridin düğüm noktaları kürenin yüzeyine radyal olarak aktarılmıştır. Bunun sonucunda 218 görüş açısı elde edilmiştir. Görüş açısı sayısı artırılırsa daha başarılı tanıma yapmak mümkündür. Ancak bu, dağıtım tablosu için gereken bellek büyüklüğünü artırır. Daha hassas konum bulunması istendiğinde, bu yöntem kaba bir



Şekil 5.4. Saklı çizgilerin yok edilmesini sağlayan programdan örnek çıktılar.

başlangıç kestirimini verir. Bundan sonra eniyi-enküçük kareler hata kriteri kullanılarak daha hassas konum bulunur.

Deney için iki model nesne kullanılmıştır. Her birisi için ayrı dağıtım tabloları oluşturulmuştur. Aslında bu istenen bir durum değildir ve tanıma süresini uzatır. Ancak, dağıtım tablosu büyüklüğü tek bir nesne için bile çok büyük olmaktadır. IBM uyumlu bir PC deki bellek sınırlamalarından dolayı, birden çok nesne için tek bir dağıtım tablosu sorun olmaktadır. Hatta, nesnenin köşe ve kenar sayısı fazla olduğu durumlarda bile tek bir dağıtım tablosunu kullanmak mümkün olmamaktadır. Deneylerde, nesnelere birisi için 2 dağıtım tablosu kullanılmıştır. Her bir tablo nesnenin değişik açılardan görünümünü tutmaktadır.

Bir görünümdeki nesneyle ilişkin doğru parçaları önceki bölümlerde anlatılan görüntü işleme yöntemleriyle bulunmuştur. Önce görüntü yumuşatılmıştır. Sonra, Sobel işleci kullanılarak kenarlar sezilmiştir. Doğru parçaları rotasyon dönüşümü kullanılarak sezilmiştir. Çizgilerden oluşan sonuç çizim gerçek durumdan biraz farklıdır. Bazı çizgiler kaybolmuşken, bazı hatalı çizgiler sezilmiştir. Ayrıca, çizgi uç noktalarının yeri çok doğru bulunmamıştır. Bu bölümde, bu durum, bundan sonra "görüntü gürültüsü" olarak adlandırılacaktır.

Dağıtım tablosu gözelerinin büyüklüğü için doğrusal olmayan bir niceme yöntemi kullanılmıştır. Koordinat değerleri arttıkça görüntü gürültüsünün etkisinin daha büyük olacağı varsayılmıştır. Bundan dolayı dağıtım tablosu gözelerinin büyüklüğü koordinat değerleri büyüdükçe doğrusal

olarak artırılmıştır. Dağıtım tablosunun oluşturulmasında x ve y yönlerindeki nicemleme sınırları

$$X(0) = Y(0) = 0$$

$$X(i) - X(i-1) = Y(i) - Y(i-1) = i\delta \quad i=1,2,\dots$$

$$X(i) - X(i+1) = Y(i) - Y(i+1) = i\delta \quad i=-1,-2,\dots$$

bağıntıları ile hesaplanmış ve (x,y) koordinatlarındaki bir nokta

$$X(i-1) \leq x \leq X(i) \text{ ve } Y(j-i) \leq y \leq Y(j)$$

sınırları içinde ise (i,j) gözesine atanmıştır. Deneylerde $\delta=0.005$ kullanılmıştır.

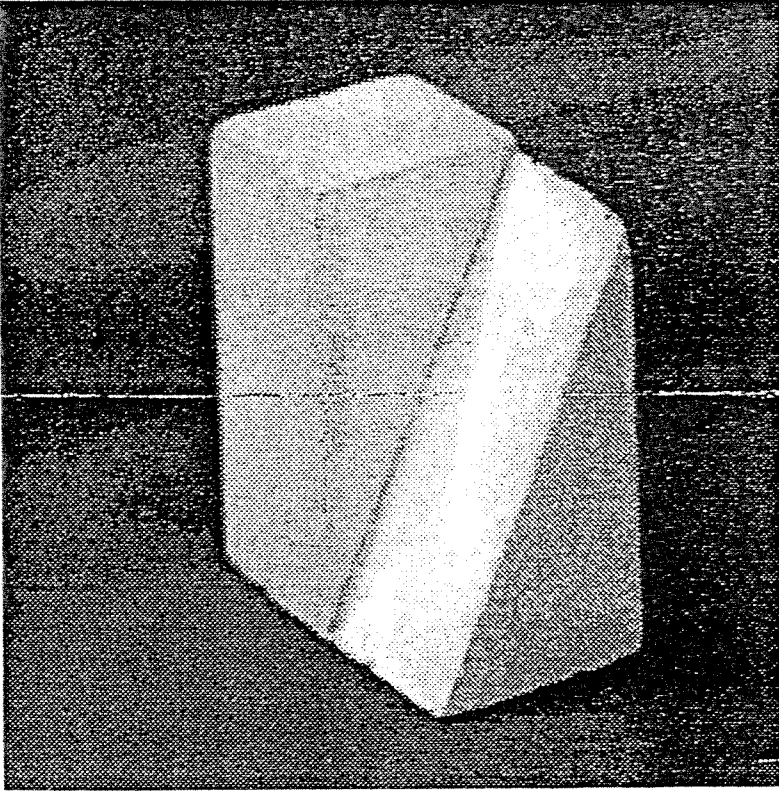
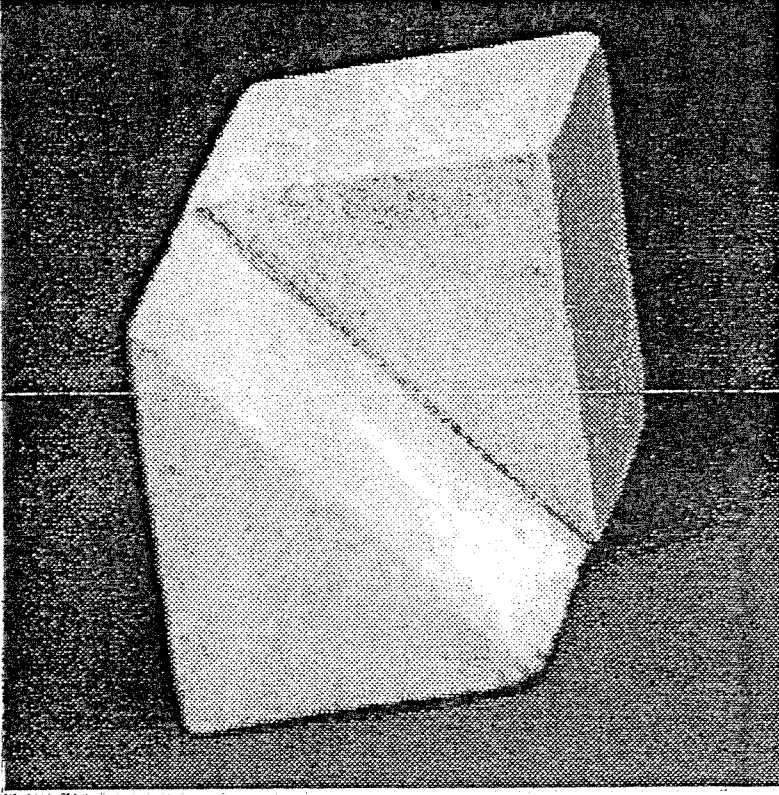
Oylama, yönlendirilmiş bir doğru parçasını taban olarak seçmek ile başlar. Daha sonra, görüntü köşe noktalarının koordinatlarına karşı gelen dağıtım tablosu gözeleri bulunur. Seçilen dağıtım tablosu gözesindeki her üçlünün oyu 1 artırılır. Ayrıca, seçilen gözenin komşuları da oylama için kullanılır. Bu işlem görüntü gürültüsünden dolayı bazı oyların kaybını önler. Görüntüde doğru parçası uç noktalarının sayısının 0.3 ile çarpımından daha büyük oy olan üçlüler eşleme adayı sayılıp köşe ve çizgi uç noktaları bilgileri ile sınırlanır. Köşe sınaması eşik değeri, çizgi uç değerlerinin sayısı çarpı 0.6 ve çizgi sınaması eşik değeri, doğru parçalarının sayısı çarpı 0.8 olarak seçilmiştir. Bu değerler deneme-yanılma yoluyla elde edilmiştir.

5.5. Deneysel Sonular

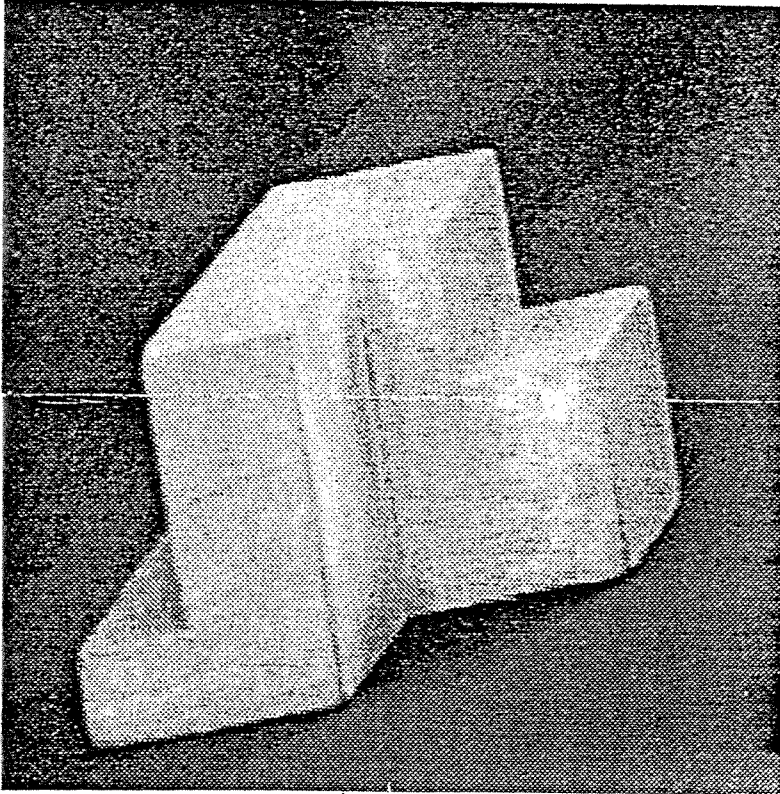
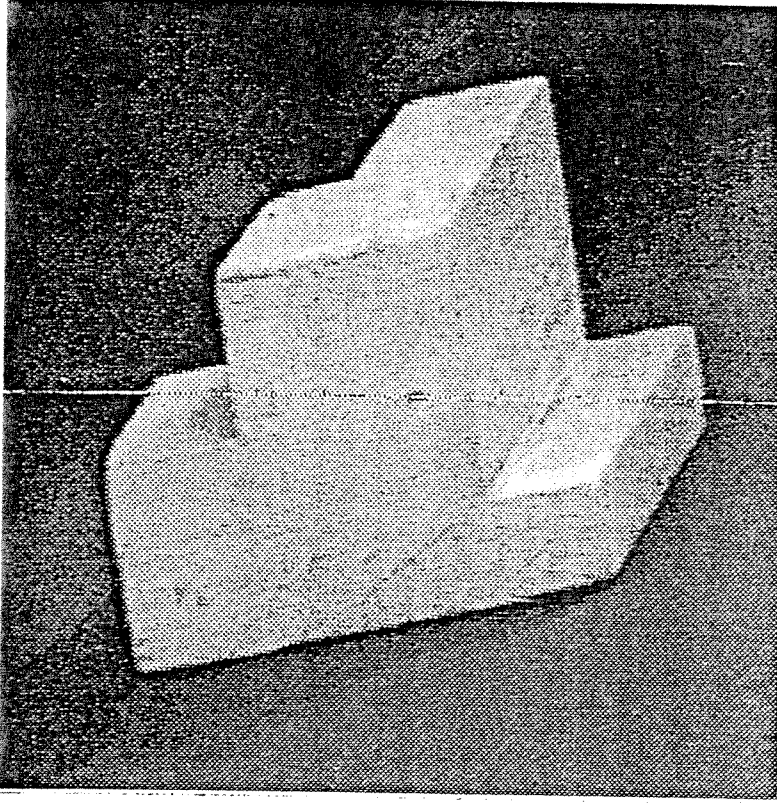
Deneylelerde iki nesne (ki bu b6l6m iinde Nesne1 ve Nesne2 diye belirtilecektir) ve bu iki nesnenin herbirinden alınan ikişer g6r6nt6 kullanılmıřtır. G6r6nt6ler Őekil 5.5 ve 5.6 da g6r6lmektedir. Her model iin dađıtım tabloları oluřturulmuřtur. Tanıma algoritması g6r6nt6lerden bulunan izgi izimleri ve dađıtım tabloları kullanılarak sınanmıřtır.

Nesne1'in dađıtım tablosu model olarak alındıđında Nesne1'in iki g6r6nt6s6 bařarıyla tanınmıřtır. Model, g6r6nt6yle iki adımda akıřtırılmıřtır. İlk6nce dađıtım tablosu oylaması sonucunda, g6r6nt6n6n ve modelin taban vekt6rleri akıřtırılmıř, daha sonra eniyi-enk66k kareler kriteri ile bulunan d6n6ř6m modele uygulanmıřtır. G6r6nt6lerden birisi iin Őekil 5.7 ve 5.8 de g6r6ld6đ6 gibi modelin birbirine ok yakın iki aıdan g6r6n6m6 bulunmuřtur. 6te yandan Nesne2 nin iki g6r6nt6s6 t6m taban vekt6rleri kullanılarak sınanmıřtır ancak Nesne1 in dađıtım tablosu tarafından tanınmamıřtır. Tablo 5.1.a da bu dađıtım tablosu ile ilgili bazı istatistiksel sonular g6r6lmektedir.

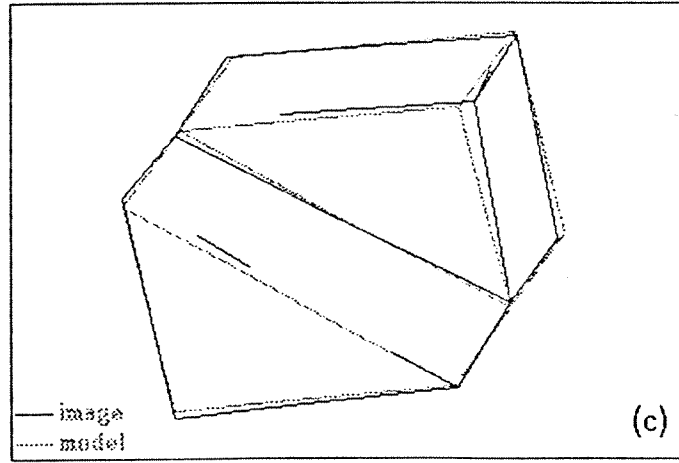
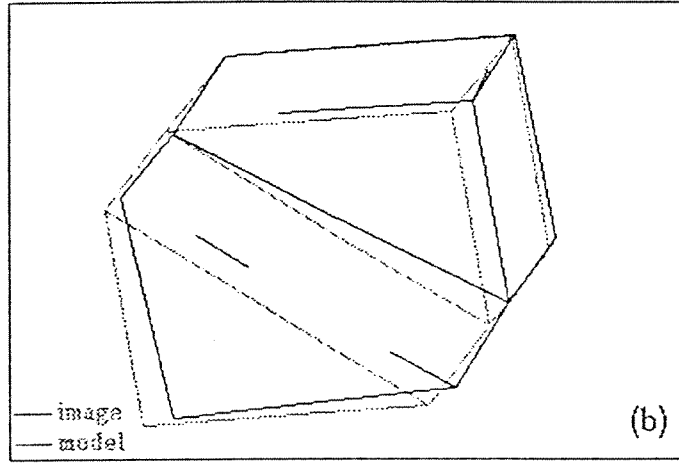
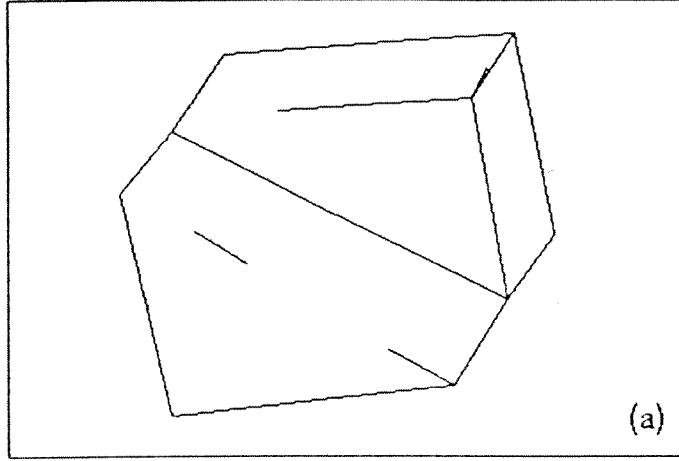
Nesne2 iin dađıtım tablosu ařırı b6y6k olduđundan model g6r6n6mlerinin bir b6l6m6 bir dađıtım tablosuna kalanı diđer bir dađıtım tablosuna kaydedilmıřtir. Nesne2'nin iki g6r6nt6s6 bu tablolarla bařarı ile tanınmıřtır. Nesne1'in g6r6nt6leri ise bilinmeyen nesne olarak kabul edilmiřtir. Tanıma iřlemi sonuları Őekil 5.9 ve 5.10'da, istatistiksel sonular ise Tablo 5.1.b de verilmiřtir.



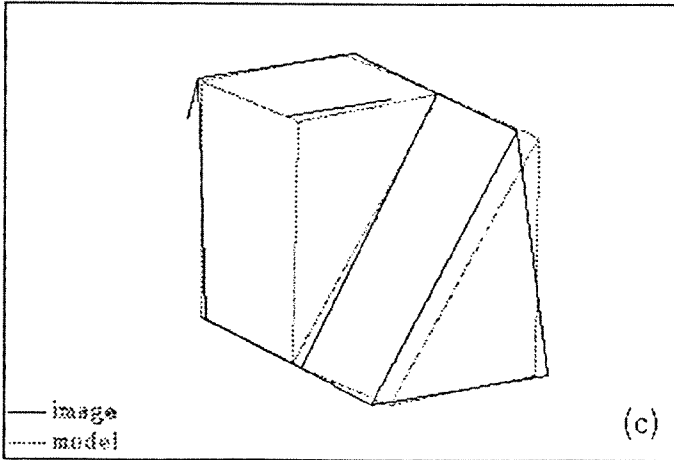
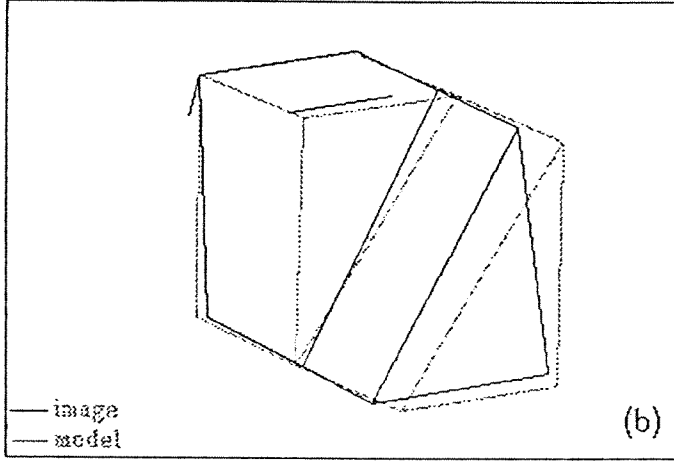
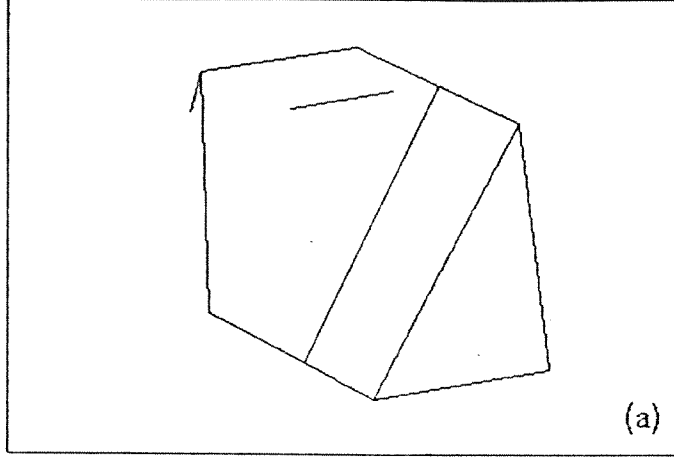
Şekil 5.5. Nesne 1'in orijinal görüntüleri



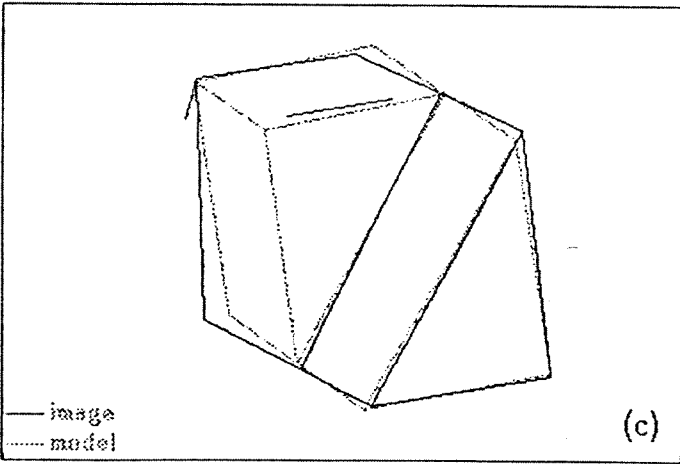
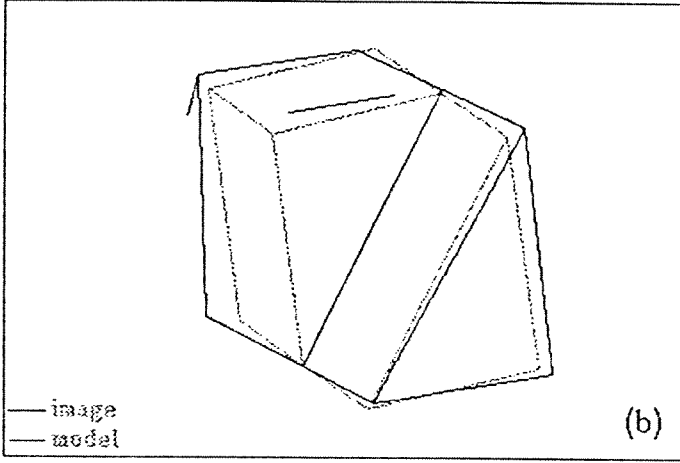
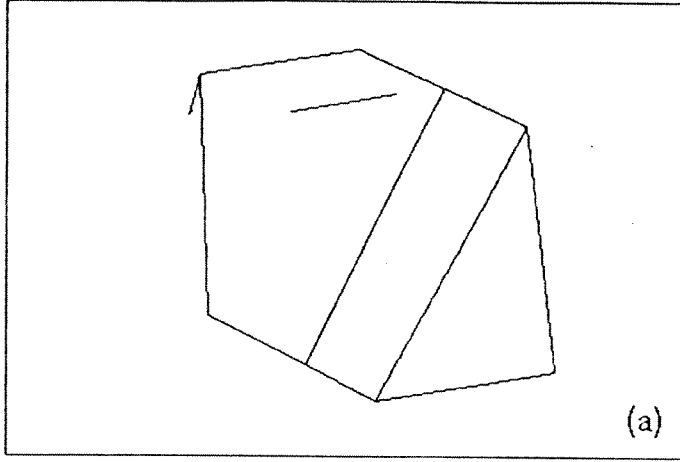
Şekil 5.6. Nesne 2'nin orijinal görüntüleri



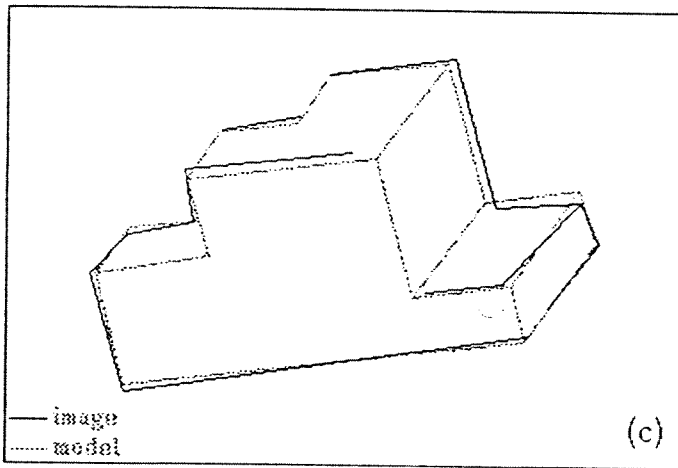
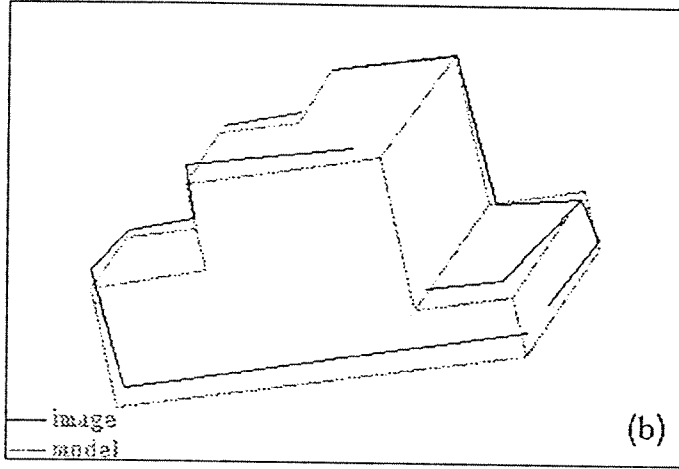
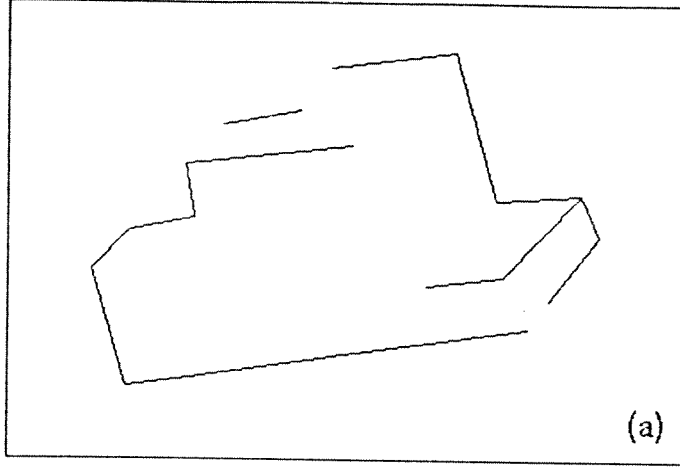
Şekil 5.7. Nesnel'in dağıtım tablosu kullanıldığında Nesnel'in birinci görüntüsü için deney sonuçları (a) görüntü, (b) taban vektörleri kullanılarak yapılan eşleme (c) eniyi-enküçük kareler dönüşümünden sonraki eşleme.



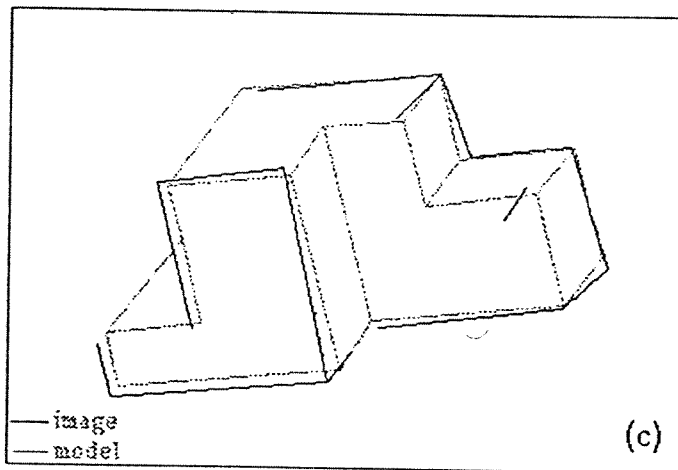
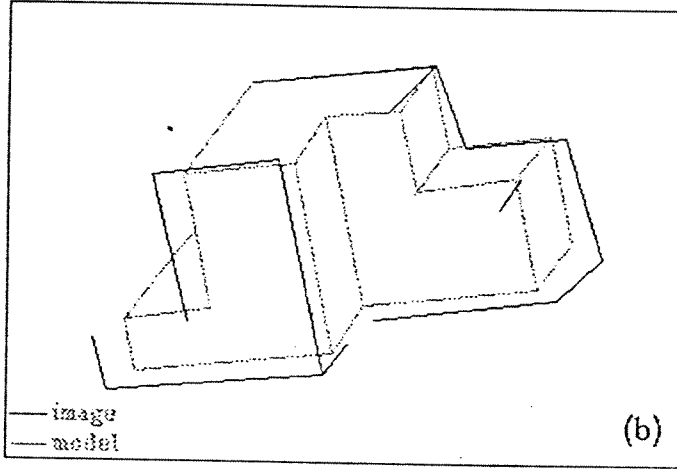
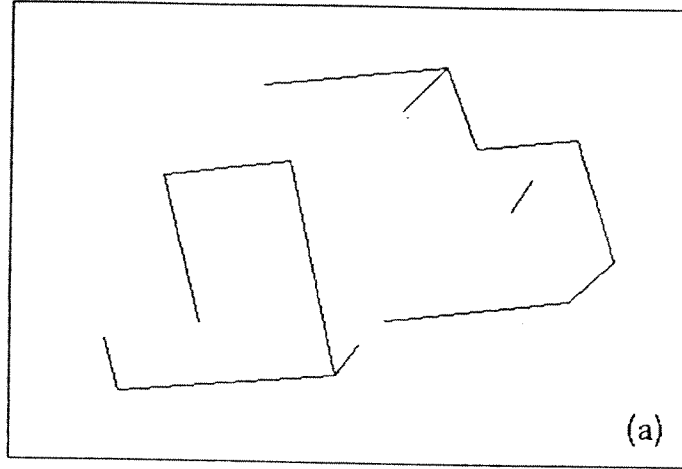
Şekil 5.8. A. Nesnel'in dağıtım tablosu kullanıldığında Nesnel'in ikinci görüntüsü için deney sonuçları (a), (b), (c) Şek. 5.7'deki gibi



Şekil 5.8. B. Nesnel'in dağıtım tablosu kullanıldığında Nesnel'in ikinci görüntüsünün, modelin bir başka görünümü ile eşlenmesi (a), (b), (c) Şek. 5.7'deki gibi



Şekil 5.9. Nesne2'nin dağıtım tablosu kullanıldığında Nesne2'nin birinci görüntüsü için deney sonuçları (a), (b), (c) Şek. 5.7'deki gibi



Şekil 5.10. Nesne2'nin dağıtım tablosu kullanıldığında Nesne2'nin ikinci görüntüsü için deney sonuçları (a), (b), (c) Şek. 5.7'deki gibi

Tablo 5.1. Oylama Sonuçları ve işlem süreleri (Paydalarda gösterilen skorlar en yüksek oy alan yanlış eşlemeler için.)

(a)

Nesne1'in Dağıtım Tablosu	Dağıtım Tablosu Oyu	Köşe Sağlama Skoru	Çizgi Sağlama Skoru	İşlem Süresi (saniye)
Nesne1'in 1nci görüntüsü	5/5	9/8	20/2	1
Nesne1'in 2nci Görüntüsü	4/4	7/6	17/8	1
Nesne2'nin 1nci Görüntüsü	6	5	x	44
Nesne2'nin 2nci Görüntüsü	6	5	x	45

(b)

Nesne2'nin Dağıtım Tablosu	Dağıtım Tablosu Oyu	Köşe Sağlama Skoru	Çizgi Sağlama Skoru	İşlem Süresi (saniye)
Nesne1'in 1nci görüntüsü	5	9	-15	92
Nesne1'in 2nci Görüntüsü	4	7	-4	100
Nesne2'nin 1nci Görüntüsü	10/9	15/9	21	4
Nesne2'nin 2nci Görüntüsü	12/9	15/11	16/2	39

Tablo 5.1 de deęişik görüntüler için oylama skorları verilmiştir. Tanınan nesnelere için oylama sonunda yanlış eşleme veren en yüksek skorlar da ayrıca gösterilmiştir. Nesne reddedilmişse en yüksek skor verilmiştir. Bunların yanında tanıma ve reddetme süreleri de gösterilmiştir.

Tablo 5.1 den görüldüğü gibi yanlış bir model görünümü, bazan doğru görünümün aldığı oy dolayında oy alabilmektedir. Bu durum köşe sinama skorları için de geçerlidir. Ancak, doğru ve yanlış görüntülerin çizgi sağlama skorları arasında büyük bir fark görülmektedir. Yöntemin güvenilirliğini bu durum artırmaktadır. Yanlış bir görünümü doğru olarak kabul etme olasılığı oldukça azdır. Bundan dolayı eşik değerleri, doğru bir görünümü reddetmeyecek kadar düşük seçilmelidir. Eğer eşik değerleri çok düşük seçilirse, sağlaması yapılacak model görüntüleri sayısı yüksek olabilir. En uygun eşik değerleri modellere ve çizgi-çizim elde etme aşamasındaki başarıma bağlıdır.

Tablolardan açıkça görüldüğü gibi, görüntüdeki nesnenin dağıtım tablosunda modeli bulunuyorsa, bir-iki saniyede nesne tanınmakta ve model çakıştırılabilmektedir. Bu süre nitem çıkartma süresini kapsamamaktadır. Nesne2'nin ikinci görüntüsünün tanıma süresi, yanıltıcı olabilecek şekilde, çok uzun görülmektedir. Bunun nedeni, iki ayrı dağıtım tablosu kullanılması ve alınan görüntünün modelinin ikinci dağıtım tablosunda olması nedeniyle birinci dağıtım tablosunda tüm taban vektörlerinin birer birer sinanması zorunda kalındığındandır. Tek bir dağıtım tablosu ile bu süre bir-iki

saniye dolayındadır. Gerçekten de, ikinci dağıtım tablosu ile başlandığında süre 2 saniye olmaktadır. Görüntülerin reddedilme sürelerinin, beklenildiği üzere, tanıma süresine göre uzun olduğu görülmektedir. Bunun nedeni kenar görüntüsündeki tüm kenarların taban vektörü olarak birer birer denenmesidir. Bu süre dağıtım tablosu büyüklüğüne göre değişmektedir.

Bu uygulamada, alınan bir tek görüntü için, birden çok model nesne, sağlama aşamasını geçebilir. Şu andaki gerçekleştirme eşlenen ilk modeli doğru model olarak seçmektedir. Eğer tüm model nesneler için tek bir dağıtım tablosu kullanılırsa, en çok oy alan model köşe ve çizgi sağlama aşamalarına girdi kabul edilebilir. Ayrı dağıtım tabloları kullanılırsa oylama her dağıtım tablosu için yinelenebilir. Daha sonra en yüksek oy alan model daha ileri sağlama aşamalarına kabul edilir.

Bazı durumlarda birden fazla nesnenin bazı görünüşleri aynı olabilir. Bu durum yanlış karar vermeye neden olabilir. Dağıtım tablosuna bu tür bilgiler konularak bu durumlar tanıma işlemi dışına çıkartılabilir.

BÖLÜM 6

SONUÇ

Hareketli nesnelere tanınması amacıyla bir model-tabanlı, üç boyutlu nesne tanıma sistemi geliştirildi ve sınılandı. Sistem önışleme, nitem çıkartma, modelleme ve eşleme kısımlarından oluşmaktadır. Her kısım için bazı algoritmalar gerçekleştirilmiştir.

Proje başlangıcında birden fazla görüntüyü kullanarak daha az hatalı bir kenar görüntüsü elde etmek ve bu görüntüyü tanıma işleminde kullanmak planlanmıştı. Aynı nesnenin iki ayrı görüş açısından elde edilen kenar görüntülerinde kenarların eşlenmesi stereo çözümleme gibi konularda oldukça önemlidir. Bu çalışmaya paralel olarak yürütölen bir başka projede stereo eşleştirme konusu ele alınmıştır. İnsan gözünün kolayca yapabileceği eşleştirme işlemi bilgisayarda oldukça karmaşık yöntemlerle ve uzun süre alan işlemlere gerçekleştirilebilmektedir. Örneğin kaynak [Tan, Gelfald ve Delp, 1991] de anlatılan ve bir stokastik gevşetme (relaxation) yöntemi olan tavlama benzeşiminin (simulated annealing) uygulandığı SunSparc II iş istasyonlarında gerçekleştirilen bir çalışma 4.5 dakika zaman almıştır. Bu yöntemlerin kişisel bilgisayarlarda ve robotik uygulamalarında kullanılması şu anda pratik olmadığından tanıma algoritması süresini mümkün olduğunca kısaltacak bir yöntem kullanılmıştır. Bunun için nesnenin değişik görüş açılarından alınan görüntüler, veya tel çerçeve modeller, sistemin eğitim evresinde kullanılmış; tanıma

evresinde ise nesnenin bir tek görüntüsü değerlendirilerek tanıma işlemi gerçekleştirilmiştir.

Önişleme, yumuşatma ve kenar sezme işlemlerini kapsamaktadır, Yumuşatma için komşuluk ortalaması yaklaşımı kullanılmıştır. Kenarlar hızlı Sobel operatörü kullanılarak sezilmiştir. Sobel işleği kenarları belirginleştirilmiş bir görüntü verir. Bu görüntüden deneme-yanılma yolu ile bulunan bir eşik değeri kullanılarak kenar pixelleri bulunmuştur. Bulunan eşik değeri benzer aydınlanma koşulları için geçerlidir. Koşullar değiştiğinde eşik değerinin yeniden bulunması veya uyarlamalı eşik değeri bulma tekniklerinin uygulanması gereklidir.

Polihedral nesnelere tanınması için en doğal niteliler nesneyi tanımlayan doğru parçalarıdır. Üç doğru parçası elde etme algoritması denenmiştir. Köşe noktalarını kullanan bir yöntem, klasik Hough dönüşümü ve rotasyon dönüşümü. İlk iki yöntem istenen başarıyı sağlayamamıştır, üçüncü yöntem ise bazı dezavantajlarına karşın yeterli bulunmuştur.

Son olarak, nesne modelleme ve eşleme işleri için geometrik dağıtım yöntemi kullanılmıştır. Deneyler yöntemin, bilinmeyen bir nesnenin bilinmeyen bir yönden alınan görüntüsünden başarıyla tanınabileceğini ve konumunun bulunabileceğini göstermiştir. Yöntemin dezavantajlı yanı dağıtım tablosunun büyüklüğüdür. Eğer nesnelere her açıdan görünüşleri dağıtım tablosunda modellenecekse, tüm nesnelere için tek dağıtım tablosu kullanımı mümkün değildir. Hatta, nesnenin çok sayıda niteli olduğu zamanda bile bir dağıtım

tablosu yetmeyebilmektedir. İki ayrı tablonun varlığı ise bazen tanıma süresini artırmaktadır. Görüntüdeki nesnenin modeli, denenen dağıtım tablosunda varsa, tanıma süresi bir-iki saniye sürmektedir. Öte yandan nesne modeli olmadığı durumda reddetme süresi bir dakika dolayındadır. Bu değerlere nitem çıkartma süresi dahil edilmemiştir. (Uygulamada nitem çıkartma işleminin donanımla çözülebileceği ve bu işlemin çok kısa zaman alacağı varsayılabılır). Reddetme süresini azaltmak için dağıtım tablosu arama işlemi birkaç taban vektörü kullanımından sonra durdurulabilir. Red işleminden sonra başka modelin dağıtım tablosu kullanılması gerekir. Eğer hiçbir model bilinmeyen nesne ile eşlenemiyorsa, nesne reddedilir. Dolayısıyla, toplam reddetme süresi herbir dağıtım tablosunun reddetme süresinin toplamıdır. Eğer nesne tanınmışsa, tanıma süresi, tanınincaya kadar sınıanan dağıtım tablolarınının red süresi artı doğru modelin bulunduğu dağıtım tablosunun tanıma süresidir. Bu değerler birden fazla dağıtım tablosu kullanma zorunluluğundan dolayı çok yüksek olabilir.

Uygulamada kabul edilebilir bir tanıma süresi elde etmek için tüm nesnelere ait modellerin tek bir dağıtım tablosunda olması gerekir. Pek çok uygulamada bu sorun rahatlıkla çözülebilir. Nesnelere sınırlı sayıda kararlı konumu olduğu ve kameranın görüş açısı içinde nesnelere sınırlı sayıda görünüşleri olacağı düşünülürse, tüm nesnelere olası tüm görünüşleri bir tek dağıtım tablosunda toplanabilir. Bunun yanında nitelerin gruplandırılması ile dağıtım tablosundaki nitem sayısı azaltılarak tablo küçültülebilir. Bu durumda tanıma süresi bir-iki saniye dolayında olur.

Gerçekleştirilen sistem, birden fazla nesne arasında istenen birisinin seçilmesini ve manipulasyonunu gerektiren endüstriyel işlemlerde kullanılabilir. Bunun yanında, bir üretim hattında üretilen nesnelere büyüklük ölçümleri yapılabilir.

Bu çalışmanın devamı olarak gelecekte yapılabilecek çalışmalar arasında şunlar sayılabilir: Nesnelere birbiriyle kısmen örttüğü durumlarda tanıma işlemi; Dağıtım tablosu skoru düşük olduğu durumlarda tanıma işlemi; Sağlama aşamasında iyileştirme çalışmaları; Eğri kenarlı nesnelere için en uygun niteliklerin çıkartılması; Nitelik gruplandırma yöntemleri geliştirilerek dağıtım tablosunu küçültme; Aynı 2-D görünümü olan farklı nesnelere ayırmak için, hareket bilgisini kullanarak ikinci bir görüntüyü tanıma işlemi içine alma.

KAYNAKLAR

- Ammeraal, L., Programming Principles in Computer Graphics, John Wiley and Sons. Great Britain, (1986).
- Bolles R.C., Chain R.A., Recognizing and Locating Partially Visible Objects: The Local Feature Focus Method, Int. of Robotics Res., 1(3), 57, (1982).
- Bolles R.C., vd., 3DPO: A Three-Dimensional Part Orientation System, Robotics Research: The 1st International Symposium, ed: Brady M., Paul R., 413, (1984).
- Chin R.T., Dyer C.R., Model-Based recognition in Robot Vision, Computing Surveys, 18(1), (1986).
- Davies E.R., Machine Vision: Theory, Algorithms, Praticalities, Academic Press, London, (1990).
- Fang J.Q., Huang T.S., A Corner Finding Algorithm for Image Analysis and Recognition, Amer. Ass. Artiff. Intel. Nat. Conf., Pittsburgh, (1982).
- Gleason G.J., Agin G.J., A Modular System for Sensor-Controlled Manipulation and Inspection, Proc. of the 9th International Symposium on Industrial Robots, Society of Manufacturing Engineers, (1979).
- Gonzalez R.C., Wintz P., Digital Image Processing, Addison-Wesley, Massachusetts, (1987).

- Gu W.K., Huang T.S., Connected Line Drawing Extraction from a Perspective View of a Polyhedron, IEEE Trans. Pattern Anal. Machine Intell. 7, 422, (1985).
- Gupta M.M., Knopf G.K., Theory of Edge Perception for Computer Vision Feedback Control, Journal of Intelligent and Robotics Systems, 2, 123, (1989).
- Kang, C.W., vd., Extraction of Straight Line Segments Using Rotation Transform: Generalized Hough Transform, Pattern Recognition, 24(7), 633, (1991).
- Lamdan, Y., Wolfson H.J., Geometric Hashing: A General and Efficient Model-Based Recognition Scheme, 2nd International Conference on Computer Vision, ed: Bajcsy R., Ullman S., 238, Florida, (1988).
- Lamdan, Y., vd., Affine Invariant Model-Based Object Recognition, IEEE Trans. Robotics and Automation, 6(5), (1990).
- Oshima M., Shirai Y., Object Recognition Using Three-Dimensional Information, IEEE Trans Pattern Anal. Mach. Intell., 5(4), 353, (1983).
- Shirai Y., Three-Dimensional Computer Vision, Springer-Verlag, Berlin, (1987).
- Stockman G.C., vd., Matching Images to Models for Registration and Object Detection via Clustering, IEEE Trans Pattern Anal. Mach. Intell., 4(3), (1982).

Tan H.L., Gelfand S.B., Delp E.J., A Cost Minimization Approach
to Edge Detection Using Simulated Annealing, IEEE Trans
Pattern Anal. Mach. Intell., 14(1), 3-18, (1981).

BİBLİYOGRAFİK BİLGİ FORMU	
1- Proje No: EEEAG-9	2- Rapor Tarihi: 22.2.1994
3- Projenin Başlangıç ve Bitiş Tarihleri: 1.10.1991/30.9.1993	
4- Projenin Adı: NESNELERİN TANINMASI	
5- Proje Yürütücüsü ve Yardımcı Araştırmacılar: Prof.Dr. Mete Severcan Huseyin Uzunalioglu	
6- Projenin Yürütüldüğü Kuruluş ve Adresi: OrtaDoğu Teknik Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü Ankara	
7- Destekleyen Kuruluş(ların) Adı ve Adresi: 1) Orta Doğu Teknik Üniversitesi . Ankara 2) Third World Academy of Sciences. ICTP. 34100.Trieste. Italya	
8- Öz (Abstract): Bu çalışmada, hareketli nesnelerin tanınması amacıyla model-tabanlı bir nesne tanıma sistemi geliştirilmiştir. Sistem nitem çıkartma, modelleme ve tanıma aşamalarından oluşmaktadır. Sistem polihedral yapılı nesneler için tasarlanmıştır. Ancak, kullanılan yöntem daha genel yapılı nesneler için genelleştirilebilir. Nesne tanımlamaları için doğrusal nitemler kullanılmıştır. Üç adet doğrusal nitem çıkartma yöntemi incelenmiştir. Doğruları bulmak için köşeleri kullanan bir yöntem, klasik Hough dönüşümü yöntemi ve rotasyon dönüşümü yöntemi. Bu yöntemler arasında rotasyon dönüşümü yöntemi kabul edilebilir nitelikte sonuçlar vermiştir. Modelleme ve tanıma işlemleri için geometrik dağıtma (geometric hashing) yöntemi kullanılmıştır. Nesnelerin tel-çerçeve modellerinden başlayarak bir dağıtım tablosu oluşturulmuştur. Aynı yöntem eşleme evresinde nesnenin görüntüsünden elde edilen doğru parçalarına uygulanmıştır. Dağıtım tablosundan, bir oylama işlemi sonucunda olası uygun model-nesneler belirlenmiştir. Daha sonra eniyi-enküçük kareler eşlemesi, ve köşe ve kenar sağlama algoritmaları kullanılarak nesnenin görünümüne uyan model-nesne, konum bilgisi ile birlikte karşılaştırılmıştır. Yöntem gerçek nesne görüntüleri ile sınanmış ve başarılı sonuçlar alınmıştır. Anahtar Kelimeler: Nesne Tanıma, Model-tabanlı Nesne Tanıma, Geometrik Dağıtma.	
9- Proje ile ilgili Yayın/Tebliğlerle ilgili Bilgiler 1) Konferans Bildirisi: "IV Int. Conf. on Appl. of Physics..." 21-25/9/1992 Trieste B 2) Internal Report: "Model-based Object Recognition ..." ICTP Trieste Eylül 1992	
10- Bilim Dalı: Doçentlik B. Dalı Kodu: 619.02.05 ISIC Kodu: Uzmanlık Alanı Kodu:	
11- Dağıtım (*): <input type="checkbox"/> Sınırlı <input checked="" type="checkbox"/> Sınırsız	
12- Raporun Gizlilik Durumu : <input type="checkbox"/> Gizli <input checked="" type="checkbox"/> Gizli Değil	

(*) Projenizin Sonuç Raporunun ulaştırılmasını istediğiniz kurum ve kuruluşları ayrıca belirtiniz