

**UZMAN SİSTEM KULLANARAK İKİ BOYUTLU
İZDÜŞÜMLERDEN
KATI MODELLER OLUŞTURMA**

İsmail ŞAHİN

**DOKTORA TEZİ
MAKİNE EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**OCAK 2008
ANKARA**

İsmail ŞAHİN tarafından hazırlanan UZMAN SİSTEM KULLANARAK İKİ BOYUTLU İZDÜŞÜMLERDEN KATI MODELLER OLUŞTURMA adlı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

(Ünvanı, Adı ve Soyadı)

Prof. Dr. H. Rıza BÖRKLÜ

Tez Danışmanı, Makine Eğitimi Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Makine Eğitimi Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

(Ünvanı, Adı ve Soyadı)**

Prof. Dr. Selim ÇETİNKAYA

(Makine Eğitimi, Gazi Üniversitesi)

(Ünvanı, Adı ve Soyadı)***

Prof. Dr. Hüseyin Rıza BÖRKLÜ

(Anabilim Dalı, Üniversite Adı)

(Ünvanı, Adı ve Soyadı)

Prof. Dr. Faruk MENDİ

(Makine Eğitimi, Gazi Üniversitesi)

(Ünvanı, Adı ve Soyadı)

Prof. Dr. Ömer Faruk BAY

(Elektronik ve Bilgisayar Eğitimi, Gazi Üniversitesi)

(Ünvanı, Adı ve Soyadı)

Prof. Dr. Varol AKMAN

(Bilgisayar Mühendisliği, Bilkent Üniversitesi)

Tarih 08 / 01 / 2008

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Doktora derecesini onamıştır.

Prof. Dr. Nermin ERTAN

.....

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

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

İsmail ŞAHİN

UZMAN SİSTEM KULLANARAK İKİ BOYUTLU İZDÜŞÜMLERDEN KATI MODELLER OLUŞTURMA

(Doktora Tezi)

İsmail ŞAHİN

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Ocak 2008

ÖZET

Bu tezde, tek görünüşle ifade edilen dönel simetrik, iki ve üç görünüşle ifade edilen silindirik ve prizmatik parçaların katı modellerini oluşturan bir çalışma tanıtılmaktadır. Çalışmada bu iki farklı işlem bir arada ele alınmaktadır. Dönel simetrik parça oluşturma algoritması dönel simetrik model alanı belirleme ve model alanının kendi eksenini etrafında döndürülmesi ile katı model oluşturma aşamalarından meydana gelir. İki ve üç görünüşle ifade edilen parçalarla ilgili katı model oluşturma algoritması ise prizmatik ve silindirik parçaları işleme alır. İşlem adımları; düzlemsel ve silindirik bölgelerden model alanları oluşturma, model alanlarından ana ve temel katılar oluşturma, görünüş bölgelerinde boolean operasyonları ile ön katılar oluşturma daha sonra ön katıların hacimsel kesiştirilmesiyle katı model oluşturmaktır.

Bu algoritmalar Visual BASIC ve VisualLISP kodunda yazılan ve AutoCAD grafik ortamını kullanan bir bilgisayar programında test edilmiştir. Hazırlanan programda nesne biçimine karar verme, katı temel eleman ve görünüş bölgelerinde ön katılar oluşturma adımlarında uzman sistem destekli bir karar mekanizması kullanılmaktadır. Çalışma katı model oluşturma sürecinde YZ tekniği kullanması, üç farklı görünüş girdi türünü ele alması ve kesit görünüşleri ele alması açısından yenilikler getirmektedir. Geliştirilen bu

yaklaşım, iki veya üç görünüşte açık bir şekilde ifade edilen, silindirik ve prizmatik parçaların katı modelleri oluşturulabilmektedir.

Bilim Kodu : 708.3.028
Anahtar Kelimeler : Dik izdüşümler, 3B nesne yeniden yapılandırma, uzman sistem
Sayfa Adedi : 262
Tez Yöneticisi : Prof. Dr. Hüseyin Rıza BÖRKLÜ

**SOLID MODEL RECONSTRUCTION FROM
2D PROJECTIONS USING
EXPERT SYSTEM
(PhD. Thesis)**

İsmail ŞAHİN

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
January 2008**

ABSTRACT

This thesis reports a research work for the reconstruction of 3D models of single view and multiple views parts that may be either cylindrical or prismatic. The research involves these two 3D modelling processes. The reconstruction of rotational symmetrical parts includes the determination of model areas and application of rotational sweeping to them. The algorithm process steps are the construction of model areas from planar and cylindrical regions, construction of main and basic solids from model areas, construction of pre-solid in the views regions with boolean operations then construction of solid model by volumetric intersections of pre-solid.

These algorithms were tested through a computer programme used in the AutoCAD environment and written in Visual BASIC 6.0 and VisualLISP codes. In this programme, an expert system aided decision mechanism is used for making decision to the type of object, and constructing of basic solid and pre-solid element on the view regions. The work contributes the literature in that the use of AI during reconstruction of 3D models, tackling the problem of multiple views and sectional views. It is also possible to construct 3D models of

cylindrical and prismatic parts together or separately given by their single, two or three views.

Science Code : 708.3.028

Key Words : Orthographic views, 3D solid reconstruction, expert system

Page Number: 262

Adviser : Prof. Dr. Hüseyin Rıza BÖRKLÜ

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla ben yönlendiren danışman hocam Prof. Dr. H. Rıza BÖRKLÜ'ye, yine kıymetli tecrübelerinden faydalandığım hocalarım Prof. Dr. Faruk MENDİ ve Prof. Dr. Ömer Faruk BAY'a, ayrıca Araş. Gör. Dr. İhsan TOKTAŞ'a, Öğr. Gör. Dr. Yunus KAYIR'a, Yrd. Doç. Dr. Hüdayim BAŞAK'a, tüm çalışma arkadaşlarıma ve manevi desteğiyle beni hiçbir zaman yalnız bırakmayan eşime teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ	xv
ŞEKİLLERİN LİSTESİ	xvi
SİMGELER VE KISALTMALAR	xxii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	5
2.1. Katı Model Oluşturma Yaklaşımları	5
2.2. Öncü Çalışmalar	6
2.3. Tel Kafese Dayalı (Aşağıdan-Yukarı) Yaklaşımlar	7
2.4. Hacime Dayalı Yaklaşımlar	20
2.4.1. Klasik YKG yaklaşımını kullanan çalışmalar	20
2.4.2. Süpürme operasyonları kullanan çalışmalar	23
2.4.3. Hacimsel kesiştirme yaklaşımı kullanan çalışmalar	27
2.5 Karma Yaklaşımlar (YKG-BRep)	31
2.6. Değerlendirme	32
3. YAPAY ZEKA ve YAPAY ZEKA TEKNİKLERİ	35
3.1. Yapay Zeka	35
3.2. Yapay Zeka Teknikleri	36
3.2.1. Uzman sistemler (US)	36

Sayfa

3.2.2. Genetik algoritmalar (GA)	46
3.2.3. Yapay sinir ağıları (YSA).....	50
3.2.4. Bulanık mantık (BM)	55
3.3. Sonuç.....	59
4. 3B NESNE OLUŞTURMADA KULLANILAN NESNE TEMSİL TEKNİKLERİ	62
4.1. Tel kafes modelleme	63
4.2. Yüzey Modelleme	64
4.3. Katı Modelleme.....	66
4.3.1. Sınır temsili (Boundary Representation-BRep) modelleme.....	67
4.3.2. Yapısal katı geometri (Constructive Solid Geometry - CSG).....	69
4.3.3. Diğer temsil teknikleri.....	71
4.4. Tasarımda Unsur Kavramı	72
4.4.1. Unsur oluşturma yöntemleri.....	74
4.5. Değerlendirme.....	79
5. PROGRAM GELİŞTİRME VE PROGRAMLAMA TEKNİKLERİ.....	81
5.1. Programlama Teknikleri	82
5.1.1. Nesne tabanlı programlama.....	82
5.1.2. Fonksiyonel programlama.....	85
5.1.3. Yordamsal (Prosedürel) programlama	87
5.1.4. Modüler programlama.....	89
5.2. Sonuç.....	91

Sayfa

6. ARAŞTIRMANIN KAPSAMI	93
6.1. Giriş.....	93
6.2. Amaç	94
6.3. Kapsam.....	94
6.3.1. Girdi çeşitleri.....	95
6.3.2. Çalışmada ele alınan parça türleri	99
7. 2B GİRDİ VE VERİ TABANI OLUŞTURMA	101
7.1. Giriş.....	101
7.2. 2B Girdi.....	102
7.3. Veri Tabanı Oluşturma.....	103
7.3.1. DXF'ten veri aktarma.....	104
8. DÖNEL SİMETRİK PARÇALARIN KATI MODELLERİNİ OLUŞTURMA	110
8.2. Kesit Alınmış Dönel Simetrik Parçaların Katı Modellerini Oluşturma	112
8.2. Kesit Alınmamış Parça Katı Modellerini Oluşturma	114
8.2.1. Parçanın iç ayrıntı durumunu saptama	114
8.2.2. Döndürme eksenini altında kalan kenarları silme	115
8.2.3. Eksen üstündeki doğru ve yay köşe noktalarını düzenleme.....	116
8.2.4. Eksik çizimleri tamamlama	117
8.2.5. Tekrar çizim ve model alanı oluşturma.....	122
8.2.6. Katı model	123
9. UZMAN SİSTEM YARDIMIYLA KATI MODELLER OLUŞTURMA	124
9.1. Temel Kavram ve Tanımlamalar	126

Sayfa

9.2. Düzlemsel ve Silindirik Bölgeleri Saptama	128
9.3. İlişkili Kenar Kümesi Oluşturma (ik_s^k)	128
9.4. Açık Kenar Kümesi Kapatma Algoritması (AKKA)	131
9.4.1. Açık kenar kümesinin açık köşe noktalarının tespiti	132
9.4.2. Temas kenarlarının (t_k) bulunması	133
9.4.3. Açık kenar kümelerinin kapatılması	134
9.5. Düzlemsel Model Alanı (dma) Oluşturma	135
9.6. Silindirik Model Alanlarının (sma) Tespiti	136
9.7. Ana Katı Profilini Belirleme	137
9.7.1. Düzlemsel model alanlarından düzlemsel ana katı profili belirleme .	137
9.7.2. Silindirik model alanlarından silindirik ana katı profilini seçme	138
9.7.3. Ana katı profiline karar verme	138
9.8. Görünüş İzdüşümlerini Belirleme	139
9.8.1. Düzlemsel model alanlarının izdüşümlerinin bulunması	140
9.8.2. Silindirik model alanlarının (sma) izdüşümlerinin bulunması	144
9.9. Süpürme Miktar ve Açılarını Hesaplama	145
9.10. Ana Katı Oluşturma	146
9.11. Uzman Sistem Modülü	150
9.11.1. US veri tabanı	152
9.11.2. Kural tabanı	157
9.11.3. Çıkarım mekanizması	164
9.12. Katı Temel Elemanlar Oluşturma	167

Sayfa

9.12.1. Prizmatik katı temel elemanların oluşturulması.....	168
9.12.2. Silindirik katı temel elemanların oluşturulması	169
9.13. Görünüş Bölgelerinde Ön Katılar Oluşturma	172
9.14. Hacimsel Kesiştirme ve Katı Model	174
9.14.1. Ön katıları döndürme	174
9.14.2. Ön katıların taşınması.....	175
9.14.3. Hacimsel kesiştirme	177
9.15. Katı Model	178
10. AUTOMODEL PROGRAMININ KULLANIMI.....	179
10.1. Programa Giriş	179
10.2. Ana Menü.....	180
10.2.1. Dosya menüsü	181
10.2.2. Veri tabanı menüsü.....	185
10.2.3. 3B yorumlama menüsü.....	188
10.2.4. Uzman menü	192
10.2.5. 3B model menüsü.....	197
10.3. Program Performansının Test Edilmesi	198
11. SONUÇ VE ÖNERİLER	202
KAYNAKLAR	210
EKLER.....	222
EK - 1 Örnek Parçalar.....	223
EK – 2 Örnek Kurallar	237

Sayfa

ÖZGEÇMİŞ	259
----------------	-----

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Katı modelleme tekniklerinin karşılaştırılması	71
Çizelge 9.1. Uzman sistem bilgi tabanı ve örnek çıkarımlar	164
Çizelge 10.1. Bazı örnek parçalarda programın performansı	199
Çizelge 10.2. Bazı örnek parçalarda programın performansı	200

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Huffman'ın kullandığı etiketleme şeması.....	6
Şekil 2.2. Aşağıdan–yukarı (tel kafes) yaklaşım işlem adımları.....	8
Şekil 2.3. Tel – kafes (aşağıdan – yukarı) yaklaşımda işlem sırası.....	9
Şekil 2.4. Sakurai'nin 2B köşe tip sınıflandırması.....	12
Şekil 2.5. Yan ve arkadaşlarının keskin kenar eklemi işlemi	14
Şekil 2.6. 2B montaj resimlerinden 3B katı modellerin oluşturulması	16
Şekil 2.7. 3B konik kenar çeşitleri	18
Şekil 2.8. Klasik YKG yaklaşımı ile katı model oluşturma.....	21
Şekil 2.9. Wang'ın YKG ağacı	22
Şekil 2.10. Süpürme operasyonu ile katı model oluşturma.....	23
Şekil 2.11. VRT'de süpürme operasyonu	26
Şekil 2.12. Hacimsel kesiştirme yaklaşımı ile katı model oluşturma	28
Şekil 2.13. Silindirik elemanların hacimsel kesiştirme yaklaşımı ile katı modellerinin oluşturulması.....	30
Şekil 3.1. Uzman sistemin genel yapısı	41
Şekil 3.2. İleriye zincirleme	42
Şekil 3.3. Geriye zincirleme.....	42
Şekil 3.4. US'in ayrıntılı yapısı	43
Şekil 3.5. Genetik Algoritma işlemleri	47
Şekil 3.6. Yapay bir sinirin: (a) katmanları (b) temel bileşenleri	52
Şekil 3.7. Bulanık kümelerde üyelik fonksiyonu	57

Şekil	Sayfa
Şekil 3.8. Bulanık bir denetleyicinin yapısı	58
Şekil 4.1. Örnek tel kafes model	64
Şekil 4.2. Örnek yüzey modeller.....	65
Şekil 4.3. AutoCAD’de hazır 3B yüzeyler.....	66
Şekil 4.4. Bir katı modelin sınır temsili	67
Şekil 4.5. Bir katı modelin sınır temsili	68
Şekil 4.6. Bir YKG ağacı örneği	70
Şekil 4.7. BDT Sisteminin yapısı.....	74
Şekil 4.8. Etkileşimli unsur tanımlama	77
Şekil 4.9. Otomatik unsur tanıma.....	78
Şekil 4.10. Unsurlar ile tasarım.....	79
Şekil 5.1. Nesne tabanlı programlama yazılımı	83
Şekil 5.2. Yordamsal programlamada ana program –yordam ilişkisi.....	88
Şekil 5.3. Modüler programlamada ana program – modül ilişkisi.....	90
Şekil 5.4. Geliştirilen algorithmada kullanılan programlar.....	92
Şekil 6.1. Birinci izdüşüm yöntemi.....	97
Şekil 6.2. Bir doğru parçasının iki farklı çizgi türüne sahip olma durumu.....	97
Şekil 6.3. İzdüşümlerin çakışması.....	98
Şekil 6.4. Ana parçanın arkasında kalan temel elemanlar	98
Şekil 6.5. Çalışmada ele alınan çeşitli silindirik delik unsurları	99
Şekil 6.6. Çalışmada ele alınan çeşitli prizmatik delik unsurları	99

Şekil	Sayfa
Şekil 6.7. Köşe, kenar cepler ve kenar unsurları.....	100
Şekil 6.8. Çalışmada ele alınan prizmatik ve silindirik çıkıntılar	100
Şekil 7.1. Program modüllerine geçiş	102
Şekil 7.2. Microsoft Access ortamında oluşturulan veri tabanı	104
Şekil 7.3. Veri aktarım işlemi	105
Şekil 7.4. Bir doğruya ait köşe noktalarının birbirlerine göre konumları	107
Şekil 7.5. Daire ve ilgili veriler.....	108
Şekil 7.6. Yay ve ilgili veriler	109
Şekil 8.1. Dönel simetrik parça oluşturma işlem akış şeması	110
Şekil 8.2. Kesit alınmış bir parçanın ifadesi	111
Şekil 8.3. 2B görüşlerin tam ve yarım kesitle ifadesi	112
Şekil 8.4. Kesit alınmış dönel simetrik parçanın katı modelini oluşturma	113
Şekil 8.5. Eksen altında kalan kenarları silme	116
Şekil 8.6. Ara sanal kenarlar	118
Şekil 8.7. İçi dolu parçalarda son sanal kenarlar.....	119
Şekil 8.8. Görünmez kenar içeren parçalarda son sanal kenarlar	120
Şekil 8.9. Tekrar çizim ve model alanı oluşturma	122
Şekil 8.10. Katı model oluşturma.....	123
Şekil 9.1. İki ve üç görüş program modülü ana işlem adımları	125
Şekil 9.2. 2B izdüşümlerle ilgili temel kavramlar.....	127
Şekil 9.3. İlişkili kenarlar	129
Şekil 9.4. İlişkili kenar oluşturma algoritması	130

Şekil	Sayfa
Şekil 9.5. Açık kenar kümesi kapatma işlemi	132
Şekil 9.6. Açık köşe noktalarının tespiti	132
Şekil 9.7. Temas kenarlarını bulma	133
Şekil 9.8. Sanal köşe oluşturma	134
Şekil 9.9. Açık kenar kümelerini kapatma	135
Şekil 9.10. Düzlemsel ve silindirik model alanları	136
Şekil 9.11. Düzlemsel model alanlarının (<i>dma</i>) en küçük ve en büyük köşe noktaları	138
Şekil 9.12. Düzlemsel, silindirik ve ana katı profillerine karar verme	139
Şekil 9.13. Ortak koordinat eksenleri	140
Şekil 9.14. <i>dma</i> 'ların izdüşümleri	141
Şekil 9.15. <i>dma</i> izdüşümlerini bulma algoritması	142
Şekil 9.16. <i>dma</i> 'ların tek köşe noktası ile eşleşme durumu	143
Şekil 9.17. Veri tabanında kayıtlı <i>dma</i> verileri	143
Şekil 9.18. Silindirik model alanlarının izdüşümleri	145
Şekil 9.19. Süpürme miktar ve açıları	146
Şekil 9.20. Ön ve yan görünüşte süpürme yönü belirleme	147
Şekil 9.21. Üst görünüşte süpürme yönü belirleme	148
Şekil 9.22. Görünüş bölgelerinde oluşturulan ana katılar	148
Şekil 9.23. <i>dma</i> 'ların tekrar çizilmesi	149
Şekil 9.24. US modülünün genel yapısı	151
Şekil 9.25. Model alanı z koordinat değerlerini belirleme	154

Şekil	Sayfa
Şekil 9.26. Model alanı z koordinat değerlerini belirleme.....	155
Şekil 9.27. US veri tabanında kayıtlı veriler.....	157
Şekil 9.28. Kural tabanından örnek kurallar.....	161
Şekil 9.29. Kural tabanından örnek kurallar.....	163
Şekil 9.30. İleriye zincirleme metoduyla katı temel eleman biçimi belirleme.....	164
Şekil 9.31. US çıkarım mekanizması sonuç verileri.....	167
Şekil 9.32. Görünüş düzlemlerde prizmatik katı temel elemanlar.....	169
Şekil 9.33. <i>sma</i> 'ların tekrar çizimi ($z = 0$ hatalı çizim).....	170
Şekil 9.34. 3B konumlarında oluşturulmuş silindirik model alanları.....	171
Şekil 9.35. Görünüş bölgelerinde oluşturulan katı temel elemanlar.....	172
Şekil 9.36. Görünüş bölgelerinde ön katılar oluşturma.....	173
Şekil 9.37. Ön katılar.....	175
Şekil 9.38. Ön katıların en büyük ve en küçük sınır noktaları.....	176
Şekil 9.39. Ön katıların taşınması.....	177
Şekil 9.40. Hacimsel kesiştirme sonrası katı model.....	177
Şekil 9.41. Döndürme sonrası katı model.....	178
Şekil 10.1. Programa giriş ekranı.....	180
Şekil 10.2. AutoModel programı ana menüsü.....	181
Şekil 10.3. Dosya menüsü.....	181
Şekil 10.4. İki boyut ön işlem penceresi.....	182
Şekil 10.5. <i>Aç</i> menüsü ile AutoCAD'le bağlantı kurma.....	183
Şekil 10.6. 2B çizimin seçimi ve ilgili katmana dahil edilmesi.....	184

Şekil	Sayfa
Şekil 10.7. <i>Kaydet</i> menüsü.....	185
Şekil 10.8. Veri tabanı menüsü	185
Şekil 10.9. Ön görünüş için Access ortamında oluşturulan veri tabanı	186
Şekil 10.10. Veri aktarma penceresi	186
Şekil 10.11. Access veri tabanında düzenlenmiş çizim verileri.....	187
Şekil 10.12. Veri görüntüleme ekranı ve anlık yardım penceresi	188
Şekil 10.13. 3B veri yorumlama menüsü	188
Şekil 10.14. Veri yorumlama penceresi	189
Şekil 10.15. Model alanı veri tabanı	190
Şekil 10.16. Model alanı ve izdüşüm verileri	191
Şekil 10.17. 2B görünüşlerin tekrar çizimi	191
Şekil 10.18. Uzman menü	192
Şekil 10.19. Yönetici şifre giriş ekranı.....	193
Şekil 10.20. Mevcut kuralları görüntüleme birimi.....	194
Şekil 10.21. Kural ekleme menüsü	195
Şekil 10.22. Kural ekleme onay uyarısı	195
Şekil 10.23. Kural düzenleme işlemi	196
Şekil 10.24. 3B model menüsü	197
Şekil 10.25. US açıklama birimi	198

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılan kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
2B	İki boyutlu
3B	Üç boyutlu
BDT	Bilgisayar destekli tasarım
CAD	Computer Aided Design
BDÇ	Bilgisayar destekli çizim
BDİ	Bilgisayar destekli imalat
CAM	Computer Aided Manufacturing
STM	Sınır temsili modelleme
BREP	Boundary Representation
AKM	Analitik katı modelleme
ASM	Analytic Solid Modelling
YKG	Yapısal katı geometri
CSG	Constructive Solid Geometry
NTP	Nesne Tabanlı Programlama
SS_n^k	Son Sanal Kenar
ik_s^k	İlişkili Kenar Kümesi
ak_s^k	Açık Kenar Kümesi
ik_s^k	Kapalı Kenar Kümesi
<i>dma</i>	Düzlemsel Model Alanı
<i>sma</i>	Silindirik Model Alanı
AKKA	Açık Kenar Kümesi Kapatma Algoritması
US	Uzman Sistem

1. GİRİŞ

“Fonksiyonel uzayda tanımlanan ihtiyaçları en uygun karşılayacak ürünün, fiziki çözüm uzayında tam bir tanıtımını yapabilmek için yürütülen tüm faaliyetler” olarak tanımlanan tasarım işlemi; amacın netleştirilmesi, kavramsal tasarım, şekillendirme tasarımı ve ayrıntılı tasarım olarak dört ana safhada incelenmektedir [1, 2].

Bilgisayar destekli tasarım (BDT), tasarım işleminde bilgisayarın etkin kullanımını içerir. Bilgisayar, problem tanımı ile başlayıp çözüm aşamasına kadar olan süreçte tasarımcıya yardımcı olmaktadır. Tasarımcı bu süreci her aşamada kontrol edip yönlendirirken; bilgisayar, kolay değiştirilebilir hassas grafik oluşturma, karmaşık tasarımları çok yüksek hızda analiz etme, verileri uygun bir şekilde depolama veya istenildiğinde geri getirme gibi yollarla tasarım işlemi verimliliği ve etkinliğinin artırılmasına yardımcı olmaktadır.

İlk Bilgisayar Destekli Çizim (BDÇ) programları, kâğıt üzerinde yapılan 2B’lu teknik resim çizimlerini bilgisayar ortamında gerçekleştirilmesi ile sınırlıdır. Ancak gelişen teknoloji ve artan ihtiyaçlar BDT sistemlerinin daha karmaşık problemleri çözmesi gerekliliğini ortaya çıkarmıştır.

BDÇ sistemleri 2,5 ve 3B’lu tel kafes sistemler olarak ortaya çıkmıştır. Fakat bu sistemlerde nesne modeli hacim içermediği için hacimsel işlemler yapılamamaktadır. Ayrıca bu sistemlerde parça biçimi ile ilgili anlam zorlukları vardır. Bu probleme paralel olarak, tasarımda göz ardı edilemeyecek yüksek tasarım ihtiyaçlarını karşılamakta güçlüklerle karşılaşmaktadır. Bu ihtiyaçlara analiz, benzetim, 3B tasarım vb. gibi hacimsel işlemler örnek gösterilebilir.

Bu ve benzeri sebeplerden dolayı nesneleri gerçek görüntülerine yakın gösteren sistemlere ihtiyaç doğmuştur. Bu ihtiyaç, bugünkü BDT ve Bilgisayar Destekli İmalat (BDİ) sistemlerini ortaya çıkarmıştır.

Yeni nesil BDT sistemlerinde en önemli özellik çizimlerin gerçek görüntüsüne yakın olarak oluşturulmasıdır. Bu da katı modelleme teknikleri ile mümkün olmaktadır. Bu sistemler ile katı temel elemanlara (küp, silindir, takoz vb.) boolean operasyonları uygulayarak (çıkartma, birleştirme vb.) veya 2B izdüşümlere bir takım işlemler uygulanarak (Süpürme, Boolean operasyonları vb.) 3B'lu modeller elde edilmektedir. Yeni nesil katı modelleme programlarında ise unsur tabanlı teknikler kullanılarak parçaların katı modelleri oluşturulmaktadır.

Son yıllarda BDT alanında çok büyük yenilik ve değişimler yaşanması ve birçok ticari BDT / BDİ sistemi geliştirilmesine rağmen bu alanda hala bazı eksiklikler bulunmaktadır. Bunlar şu şekilde sıralanabilir:

- 1) Ticari BDT / BDİ sistemlerinden hiçbiri, teknik resim veya temel görünüşleri ile ifade edilen bir parçanın 3B katı model resmini etkin bir şekilde oluşturarak kullanıcıya sunan bir modül içermemektedir.
- 2) Günümüz katı modelleme programlarında ekrandaki görüntüyü yorumlayıp sonuç üretme özelliği yoktur, bu tür işlemler kullanıcı tarafından yapılmaktadır.
- 3) Katı modelleme programları karmaşık ve fazlasıyla kullanıcı katkısına ihtiyaç duymaktadır. Bu durum, parça karmaşıklık düzeyi arttıkça yanlış yorumlara ve modelleme zamanının artmasına yol açmaktadır.
- 4) Katı modelleme programlarının etkinliği arttıkça karmaşıklığı da artmaktadır. Karmaşık programları kullanmak kullanıcının bilgi ve yeteneği ile doğrudan ilişkilidir. Doğru katı model oluşturma, kullanıcının parçayı doğru yorumlamasının yanında programı kullanma yeteneğine bağlıdır.

2B izdüşümleri yorumlayıp otomatik olarak 3B katı model oluşturma çalışmaları yukarıda sayılan sebeplerden ötürü yaklaşık kırk yıldır ciddi bir araştırma konusudur. Burada amaç, katı model oluşturma zamanını azaltırken kullanıcı yorumunu en aza indirmektir. Kullanıcının yerine ekrandaki parçayı yorumlayıp katı modeli oluşturan bir programın yukarıda bahsedilen sıkıntıları gidermesi mümkündür. Böyle bir programla aşağıdaki avantajlar sağlanabilir:

- 1) Kullanıcı yorumundan kaynaklanan hatalar azaltılabilir,
- 2) Modelleme zamanı kısaltılabilir,
- 3) Birden fazla yazılım ihtiyacı ortadan kalkar.

Bu konu üzerinde yürütülen çalışmalar hemen bütün disiplinlere yayılmıştır. Konu makine tasarımı alanında yeni olmasına karşın, tıp (özellikle iç organlarının görüntülenmesi, plastik cerrahi ve kemiklerin tekrar modellenmesi), savunma veya jeoloji (uzaydan fotoğrafları çekilen bölge veya hedeflerin modellenmesi) gibi alanlarda ciddi bir araştırma alanına sahiptir.

Bu tezde otomatik katı model oluşturma problemine önceki çalışmaların ışığında bir çözüm getirilmeye çalışılmaktadır. Çalışmada daha güncel olması ve yaygın kullanılan BDT paketlerinde kolay uygulanması sebebiyle hacimsel bir yaklaşım kullanılmıştır. Geliştirilen hacimsel yaklaşımla, tek görünüşle ifade edilen dönel simetrik, iki ve üç görünüşle ifade edilen prizmatik ve silindirik parçaların katı modelleri oluşturulabilmektedir.

Bu tez kapsamında yürütülen çalışmanın üç temel hedefi vardır. Bunlar, düzenleme, yorumlama ve yeniden yapılandırma olarak belirtilebilir. Düzenleme, AutoCAD ekranındaki 2B'li verileri düzenleme, girdi verilerinden yeni veriler oluşturma ve bunlardan ayrıntılı veri tabanı oluşturmaya kapsar.

Yorumlama, düzenlenmiş 2B'li veriler arasında ilişkiler kurarak 3B'li nesne yorumları oluşturmaya içerir. Bu yorumları geliştirilen bir uzman sistem modeli içerisinde kullanarak katı temel eleman biçimi ve operasyonlarına karar vermeyi ele alır.

Yeniden yapılandırma, geliştirilen uzman sistem destekli karar mekanizmasıyla belirlenen katı temel elemanların bilgisayar grafik ortamında çizimini ve bu katı temel elemanlar arasında boolean (birleştirme, çıkartma ve kesiştirme) operasyonları ile katı model oluşturmaya içerir.

Çalışmanın özellikleri ve diğer çalışmalardan farkı beş ana başlık altında özetlenebilir:

1. *Görünüşlerin girdi sayısı:* Bugüne kadar yapılan çalışmalarda girdi sayısı tek, iki, üç ve altı görüş olarak seçilmiştir. Önceki çalışmalarda bir, iki, üç veya altı görüşten herhangi biri kullanılmıştır. Burada yapılan çalışmada bir, iki ve üç görüş girdisi bir arada ele alınmaktadır.
2. *Aynı Programda İki Farklı Algoritma:* Dönel simetrik parça yapım algoritması ile iki ve üç görüşle ifade edilen parçaların katı modellerini oluşturan algoritma farklıdır. Çalışma, iki farklı nesne oluşturma algoritmasını aynı program içinde kullanması yönü ile alana yenilik katmaktadır.
3. *Ele aldığı nesne türleri:* Çalışmada, standart makine unsurlarının (delik ve çıkıntı unsurları) büyük bölümü oluşturulabilmektedir. Ayrıca, görüş bölgelerinde açık bir şekilde ifade edilen, iç çıkıntı içermeyen, tek çizgi türü ile ifade edilmiş, teknik resim çizim kurallarına uygun oluşturulmuş iki veya üç görüşle ifade edilen silindirik, prizmatik veya karma parçaların katı modelleri oluşturulmaktadır.
4. *Kesit görüşlerle ifade edilen nesneleri yorumlayabilme:* Geliştirilen algorithmada tam ve yarım kesitle ifade edilen nesnelerin de katı modelleri oluşturulabilmektedir. Kaynaklarda kesit görüşlerle ilgili bir çalışmaya rastlanmıştır. Ancak, tez çalışması tek, iki ve üç görüş girdisinin üçüne birden kesit görüşleri uygulayan ilk çalışmadır. Burada geliştirilen yöntemle tek kalınlıklı nesnelerin tam ve yarım kesitleri yorumlanmaktadır.
5. *3B yorum elde etmede uzman sistem karar desteği:* Çalışmada 3B yorum elde etme aşamasında yapay zeka tekniklerinden uzman sistem kullanılmıştır. Oluşturulacak katı temel elemanların biçimine ve buna bağlı olarak birbirleri arasında uygulanacak boolean operasyonlarına hazırlanan uzman sistem aracılığı ile karar verilmektedir.

2. KAYNAK ARAŞTIRMASI

2.1. Katı Model Oluşturma Yaklaşımları

2B’lu teknik resim girdilerini yorumlayıp otomatik katı model oluşturma düşüncesi, bilgisayarların mühendislik çizimlerinde kullanılmaya başlanmasına kadar uzanır. İlk Bilgisayar Destekli Çizim programı olan Scatchpad'den bu güne araştırmacılar hızlı ve otomatik bir şekilde katı model oluşturma sorunuyla ilgilenmişler ve bu sorunu çözmek için yoğun çalışmalar yürütmüşlerdir.

Günümüzde daha kolay ve etkin tasarım, analiz ve simülasyon gibi işlemleri yapmak için çok sayıda kapsamlı BDT paket programları mevcuttur. Bu programları kullanarak teknik resim çizimleri yanında tasarım, analiz, simülasyon, işlem planlama gibi karmaşık pek çok işlem bir arada gerçekleştirilebilmektedir.

Bu tür programlarda katı modelin doğru oluşturması kullanıcının programı kullanabilme yeteneğine ve parçayı doğru yorumlamasına bağlıdır. Eğer kullanıcı programı etkin kullanamıyorsa, parçayı doğru yorumlasa bile modeli oluşturamamaktadır. Kullanıcının parçayı doğru yorumlayamaması ise katı modelin yanlış oluşturulmasına sebep olmaktadır. Bunların yanında kullanıcının parça yorumlama ve program kullanabilmek kabiliyeti katı model oluşturma zamanına doğrudan tesir etmektedir.

Bu nedenlerden dolayı katı model oluşturma sürecini otomatikleştirmeye yönelik olarak yaklaşık kırk yıldır pek çok bilimsel araştırma yapılmıştır. 2B görüşlerden 3B katı model oluşturma çalışmaları kaynaklarda tel kafese dayalı (Aşağıdan Yukarı – Buttom Up) ve hacime dayalı (Yapısal Katı Geometri – YKG) yaklaşımlar olarak iki ana başlık altında incelenmektedir [3 – 6]. Son yıllarda bu iki yöntemi bir arada kullanan bazı çalışmalara rastlanmaktadır. Bu çalışmada, araştırma konusu ile ilgili çalışmaların sınıflandırılması yeni bir yaklaşımla üç ana başlık altında incelenmiştir:

- a. Tel kafese dayalı yaklaşımlar
- b. Hacim tabanlı (YKG) yaklaşımlar
- c. Karma yaklaşımlar

Bunların yanında, 2B izdüşüm görüntülerini yorumlayarak 3B model bilgisi elde etmeye yönelik olarak nesnelerin kenar oluşumlarını inceleyen bazı öncü çalışmalardan da bahsedilebilir. Bu çalışmalar “Öncü Çalışmalar” olarak ayrı bir başlık altında ele alınmıştır.

2.2. Öncü Çalışmalar

Bu alanda, izdüşümleri köşe ve kenar bağlantı türlerine göre etiketleyip yorumlayan bir yöntem geliştiren Huffman ve Clowes öncü kabul edilebilir [7, 8]. Huffman ve Clowes’in etiketleme yaklaşımı, üç yüzlü köşelerden yansıtılan çizgi türlerini sistematik bir şekilde bulmakta idi ve çakışmayan (üst üste gelmeyen) dört köşe türü ve bir köşe oluşumunu karşılayacak çizgi etiketlerinden oluşmaktadır [4, 7–9] (Şekil 2.1).

L Tipi	
T Tipi	
W Tipi	
Y Tipi	

Şekil 2.1. Huffman’ın kullandığı etiketleme şeması - [7].

Bu çalışmada girdi alanı üç köşe ve görünüşle sınırlıdır. Ayrıca burada, hacimsel açıdan farklı yorumlar belirlenememiştir. Çalışma daha sonra Falk ve Turner tarafından eksik çizgi ve eğri yüzlerin yorumlanmasını da içerecek şekilde genişletilmiştir [10, 11]. Araştırmacılar çizgi bağlantı yerlerindeki muhtemel oluşumları üç grupta sınıflandırılmaktadır [12];

- İç bükey yüzey bağlantıları
- Dış bükey yüzey bağlantıları
- Kademeli yüzey kenar.

Çizgi etiketleme yöntemi daha sonra Lee v.d. tarafından, hem düzlemsel hem de eğri yüzeyli nesneleri yorumlayacak şekilde geliştirilmiştir [13]. Lee'nin elde ettiği çizgi etiketleri yüzey gruplarını geliştirmek için kullanılmıştır. Etiketleme yaklaşımını ele alan bir diğer araştırmacı olan Malik ise, etiketleme şemasını, kavisli ve saydam olmayan yüzeyleri içine alacak şekilde genişletmiştir [14].

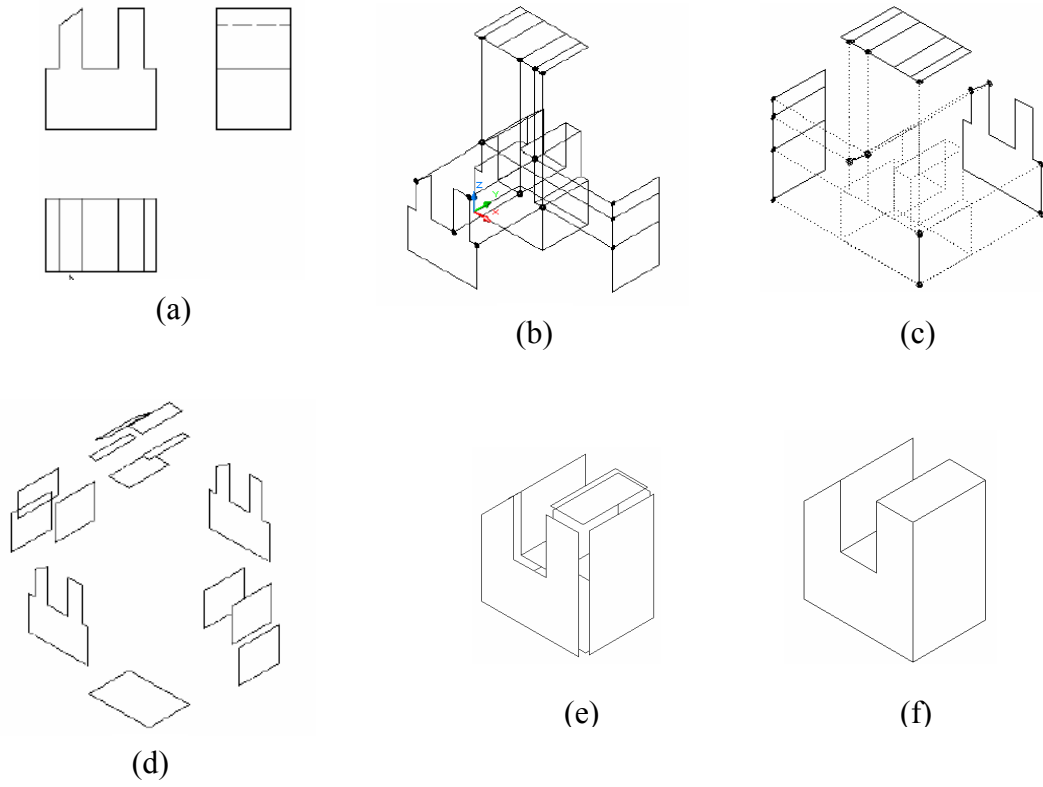
Çizgi etiketleme yaklaşımları, üç yüzlüler için ilk etiketleme şemalarını ortaya koymuştur. Yaklaşımla 2B çizimleri yorumlayarak 3B nesne yorumları oluşturulmakta, fakat bunlarda hiçbir zaman 3B nesne oluşturulamamaktadır [6].

2.3. Tel Kafese Dayalı (Aşağıdan-Yukarı) Yaklaşımlar

Tel kafese dayalı veya aşağıdan yukarı yaklaşımlar, nesne temsil tekniği olarak tel kafes veya sınır temsili tekniklerini kullanırlar. 3B'lu tel kafes model köşeler ve köşeleri birbirine bağlayan kenarlar ile gerçekleştirilir. Sınır temsilleri ise nesnelerin köşe, kenar ve yüzey sınırları ile temsil edilmesidir [15].

Her iki temsil tekniğini kullanan araştırmacılar da Idesawa tarafından aşağıdan – yukarı (Bottom-up) olarak tanımlanan yaklaşımı temel almışlardır [16, 17] (Şekil 2.2). Aşağıdan yukarı yaklaşımı dört aşamalı hiyerarşik bir metoda dayanır ve aşağıdaki işlem basamaklarından oluşur [3, 4, 9, 18, 19];

- İzdüşümlerdeki 2B noktalardan 3B köşeler oluşturma
- 3B kenarları oluşturmak için 3B noktaları kullanma
- 3B kenarları kullanarak 3B yüzeyler elde etme ve
- Yüzeylerle nesneleri sınırlandırma



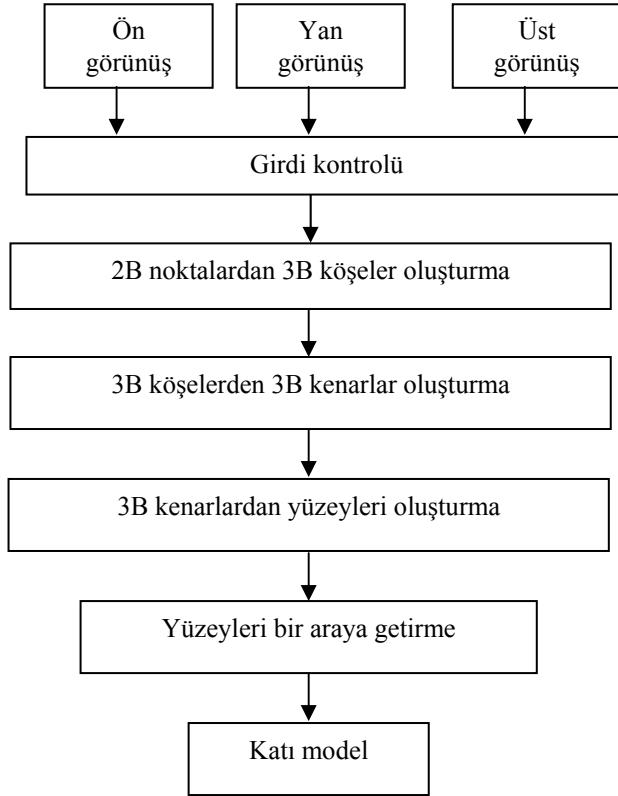
Şekil 2.2. Aşağıdan–yukarı (tel kafes) yaklaşım işlem adımları:

- a) 2B izdüşüm
- b) 3B noktalar oluşturma
- c) 3B kenarlar oluşturma
- d) yüzey Oluşturma
- e) yüzeyleri bir araya getirme
- f) katı model

Idesawa bu yaklaşımla tel çerçeve modeller oluşturmuştur (Şekil 2.3). Araştırmacı, muhtemel hatalı elemanları ayırmak için iki temel kriter kullanmıştır:

1. Bir 3B nokta en az iki 3B kenara ve iki yüzeye sahip olmalıdır
2. Bir 3B nokta sadece bir yüzeye sahip olmalıdır

Idesawa'nın yaklaşımı sadece çok yüzlü düzlemlere uygulanmıştır ve çoklu çözümler içermemektedir.



Şekil 2.3. Tel – kafes (aşağıdan – yukarı) yaklaşımda işlem sırası

Lafue, Idesawa'nın metodunda ortaya çıkan hatalı elemanları ayırmak ve doğru elemanları bulmak için bir deneme yanılma programı geliştirmiştir [20]. Program, çoklu çözüm hallerinde çoğunlukla hatalı elemanları ayıramamaktadır. Ayrıca, girdi verileri önceden tasarlanmış bir formata uygun olması gerekmektedir. Bu durum büyük zaman kayıplarına yol açmıştır [18]. Lafue'nin yaklaşımı çoklu çözümleri içermemektedir.

Sutherland'ın yaptığı çalışma 3B noktaların elde edilmesini içerir [21]. Burada 2B girdiler kullanılarak katı model elde edilmesi amaçlanmıştır. Yöntemde 2B görüşlerdeki nokta koordinatları (başlangıç ve bitiş noktaları gibi) kullanılarak 3B noktalar belirlenmektedir.

Idesawa'nın metodunu düzlemsel ve silindirik yüzeylere uygulayan Woo ve Hammer'ın yaklaşımı, girdi görünüşlerinden 3B kenarlar oluşturma, 3B kenarlardan yüzeyler yapılandırma, girdiyi karşılayan ne kadar yüzey varsa bunları birleştirmeyi içeren üç ana adımdan oluşur [22]. Algoritma çoklu çözümlerde kullanılsa da, tüm algoritmanın uygulandığı belirsizdir.

Markowsky ve Wesley iki ayrı çalışma ile aşağıdan yukarı yaklaşımı ile ilgili somut sonuçlar elde etmişlerdir [23, 24]. Daha önceki çalışmalarda bir takım sonuçlardan söz edilse de katı model oluşturma aşaması ile ilgili sağlam sonuçlar yoktur. Markowsky ve Wesley bu çalışmalarda düzgün çokgen nesnelerin 3B modellerini elde etmiştir. Bu algoritma, çeşitli temel elemanların birleştirilmesi ile katı modeller oluşturmaya temel almaktadır. Çalışmada, yüzeylerin belirlenmesi sonrası, yüzlerle çevrili bölgeler araştırılmakta ve bu bölgelerden temel katı elemanlar elde edilmektedir. Daha sonra temel elemanlar birleştirilerek muhtemel katılar oluşturulmakta ve girdi görünüşleri ile karşılaştırılmaktadır [23].

Markowsky ve Wesley daha sonra, metodu çoklu çözüm üretmek için geliştirmişlerdir [24]. Metod, yedi aşamadan oluşmaktadır. Bunlar;

- girdiyi kontrol etme,
- hayali noktalı yapıyı oluşturma,
- hayali döngü oluşturma,
- gerçek yüzeyleri oluşturma,
- kenarları oluşturma,
- gerçek blokları oluşturma,
- karar verme

aşamalarıdır.

Preiss ile birlikte girdiler önceden hazırlanmış format halinden kurtarılmıştır [25 – 27]. Preiss'in metodu, düzlemsel ve silindirik parçaları kapsar. Preiss, konu üzerinde

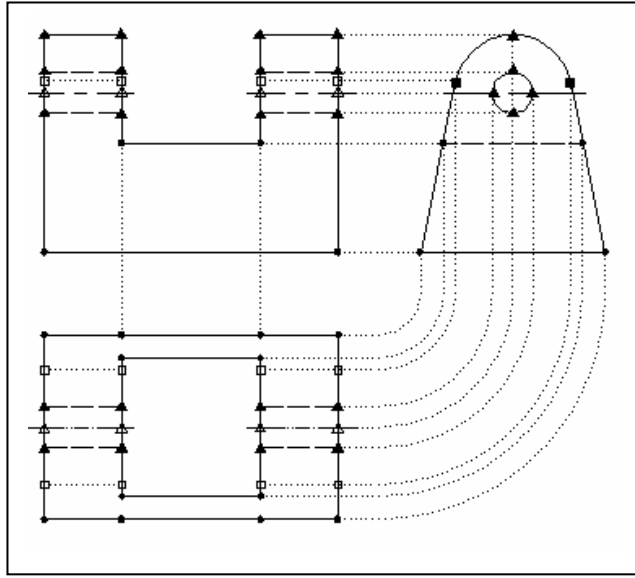
daha önce çalışan araştırmacılardan farklı olarak çokgen izdüşümlerinden görünür yüzeyler elde etmiştir [25]

Preiss başka bir çalışmada 2B görünüşleri x , y ve z koordinatlarına göre düzenleyen bir çalışma yapmıştır [26]. Yaklaşım, kullanıcının yüzey ve kenar biçimindeki girdileri koymasına izin vermemektedir. Burada tüm kullanıcı girdileri, 2B çizgi sınırlayıcılarından oluşmaktadır.

Preiss daha sonra sınırlandırmaya dayalı bir yaklaşım geliştirmiştir [27]. Burada ilk olarak aday noktalar bir takım sınırlandırmalara başvurularak belirlenir, ikinci olarak sınırlamalar her bir kenara uygulanarak aday kenarlar belirlenir, daha sonra sınırlandırmalara uymayan hatalı kenarlar yok edilir. Son olarak kenarlardan döngüler oluşturularak yüzeyler ve yüzeylerden nesne elde edilmektedir.

Sakurai ve Gossard [18], Wesley'in metodunu eğrisel yüzeyleri de kapsayacak şekilde genişleterek, dönele simetrik eğri nesnelerin katı modellerini oluşturmuşlardır. Araştırmacılar asal eksenlerden birine paralel olan, silindirik, konik ve küresel yüzeylerden oluşan nesneleri incelemişlerdir.

Sakurai ve Gossard'ın yaklaşımı, köşe-tipi sınıflandırması kullanılmaktadır. Bu yaklaşımda 2B köşe noktaları: teğetsel, standart ve hayali noktalar olarak sınıflandırılmaktadır (Şekil 2.4). Bu sınıflandırmaya göre düzenlenen noktalardan aday köşeler (3B'lu köşeler) elde edilmektedir. Muhtemel kenarlar, aday köşelerden oluşmakta ve oluşturulduğu aday köşe tipi ile sınıflandırılmaktadır. Daha sonra olası kenarlardan yeni yüzler oluşturulur. Yüzeyler oluşturulurken bir takım kriterler tanımlanarak hatalı kenarlar ve yüzeyler silinmektedir. Elde edilen nesneler, olası nesnelerin girdi görüntülerindeki 2B'lu kenarları karşılayıp karşılamadığına bakılarak elenir ve son nesneye karar verilir. Eğri yüzeyleri oluşturmada başarılı sonuçlar veren metod, oldukça karmaşıktır ve uzun işlem zamanı gerektirmektedir.



- | | | | |
|---|-------------------|---|-----------------|
| □ | Teğetsel oluşumlu | ▲ | Hayali |
| ● | Standart | △ | Hayali oluşumlu |
| ■ | Teğetsel | | |

Şekil 2.4. Sakurai'nin 2B köşe tip sınıflandırması [6].

Haralick ve Queeney, Wesley ve Markowsky'nin yaklaşımını temel alan dört aşamalı bir yaklaşım geliştirmiştir [28]. Metot ile önce 2B köşelerden 3B köşeler, köşelerden yüzeyler belirlenir daha sonra yüzey görünüşlerinin gözlemlenmesi ile birbiri ardı sıra gelen köşeler bulunur. Bir birini takip eden temel değişken noktalar her bir yüzey parçası için tanımlanmaktadır. Çalışma, çoklu çözüm içermemektedir.

Gu ve arkadaşları Sakurai ve Gossard'a benzer tarzda, eğri nesne oryantasyonları üzerine bazı sınırlayıcılar uygulayarak 3B nesneler elde etmiştir [29].

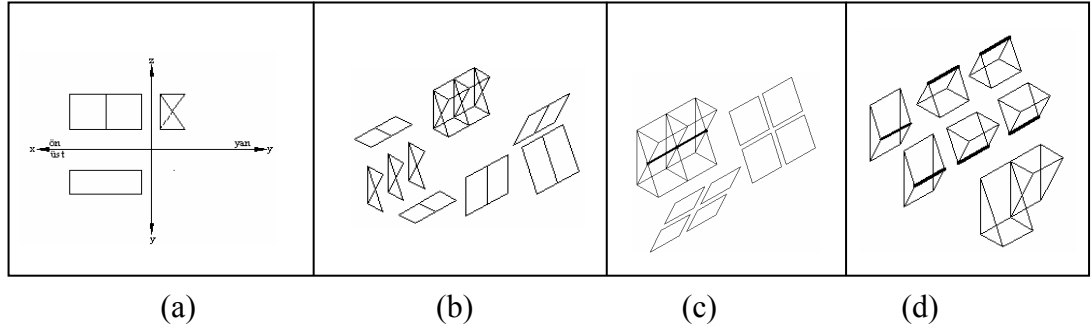
Gujar ve Nagendra araştırmalarında, Markowsky ve Wesley'in çalışmasını geliştirerek üç görünüşle ifade edilen izdüşüm görünüşlerinden düzlemsel yüzeyli nesneler oluşturmuştur [30]. İşlem altı ana adımdan oluşmaktadır. Bunlar; girdi oluşturma, muhtemel 3B noktalar ve kenarlar oluşturma, muhtemel yüzeyler oluşturma, yüzlerin birleşiminden muhtemel alt nesnelerin oluşturulması, montaj ve test. Son aşamada alt nesneler birleştirilerek 3B sınır temsili nesne oluşturulmaktadır. 3B nesne girdi görünüşleri ile karşılaştırılarak test edilmekte ve doğru nesne olup

olmadığına karar verilmektedir. Algoritma bütün olası nesneleri oluşturmakta, çoklu çözüm vermemektedir. En önemli avantajı önemli zaman tasarrufu sağlaması olmuştur.

You ve Yang'ın çalışmasında girdi altı görünüşten oluşturulmuştur [31]. Geliştirilen sistem görünüşlerdeki eksiklikleri belirleyip tanımlayabilmesi yönü ile önemlidir. Metod, yardımcı görünüşleri de kabul etmektedir. Nesneler düzlem, eğri ve yüzeylerle sınırlandırılmakta ve eğri yüzey eksenlerinin izdüşüm yönüne paralel olması istenmektedir. Algoritmada, önce çizimlerden tel kafes model, daha sonra tel kafesteki döngülerden yüzeyler oluşturulmaktadır. Çizimlerdeki görünen ve görünmeyen eğrileri de içeren bilgiler kullanılarak kenar ve yüzeylerdeki hatalar çıkarılmaktadır. Daha sonra tel kafeste aynı düzlemde yer almayan yüzeyler, diğer yüzeylerden değişken nesneler oluşturmak üzere bir böl ve belirle algoritması ile çıkartılmaktadır. Nesneler değişmeyen bir geometrik modelde Euler işlemcileri ile oluşturulur.

Yan ve arkadaşları, geometrik hesap hatalarını etkin bir şekilde düzelten bir algoritma tanıtmıştır [32]. Wesley ve Markowsky'nin yöntemine dayanan çalışma hatalı durumları saptayabilmekte ve tüm doğru çözümleri etkin bir şekilde belirleyebilmektedir. Ayrıca burada, girdi görünüşlerindeki görünmez çizgiler de dikkate alınmaktadır.

Çalışmanın önemli bir yeniliği, üst üste gelen yüzeylerdeki belirsizlikleri ortadan kaldıran bir yöntemin uygulanmış olmasıdır. Üst üste gelen ve bir birini uç noktalar dışında her hangi başka bir yerde kesen yüzeylere keskin kenar eklenmekte ve böylece yeni yüzeyler elde edilmektedir. Daha sonra bu yüzler kullanılarak sınır temsil blokları oluşturulmakta, görünmez kenarlar yorumlanarak hatalı bloklar elenmekte ve son olarak bu blokların birleştirilmesi ile de 3B'lu nesne elde oluşturulmaktadır (Şekil 2.5). Bu yaklaşım, temel katı elemanların tüm birleşimlerini incelemede yeterli değildir. Çünkü bu işlem, oldukça uzun zaman almaktadır. Ayrıca önerilen bu metod, eksik görünüşleri belirleyememektedir.



Şekil 2.5. Yan ve arkadaşlarının keskin kenar eklemi işlemi [32]:

- a) 2B üç görünüş
- b) üst üste gelen yüzeyler
- c) keskin kenar ekleme ve yeni yüzler oluşturma
- d) yeni sınır temsili bloklar ve 3B nesne

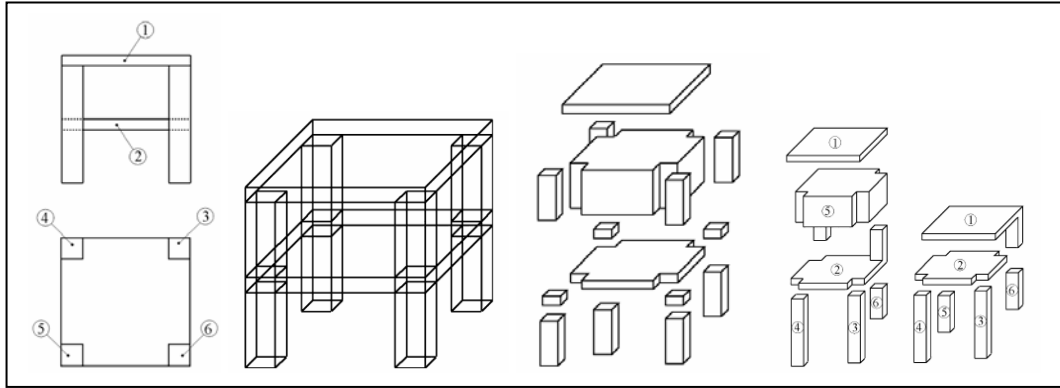
Shin ve Shin, Wesley ve Yan'ın çok yüzlü nesne yapım algoritmaları ile Sakurai'nin eğri nesne oluşturma yöntemini temel alan bir çalışma yapmıştır [19]. Çalışma silindirik, tor ve incelen nesneler gibi asal eksenlere paralel dört yüzlülerle sınırlıdır. Burada, 2B izdüşüm verileri ile 3B nesne oluşum aşamalarında elde edilen veriler arası ilişkiler kurularak deneme miktarını azaltma hedeflenmiştir. Sekiz adımdan oluşan algorithmada Sakurai'nin köşe tip sınıflandırması kullanılmaktadır. Shin, Sakurai'nin köşe tip sınıflandırılmasına ek olarak iki tip köşe daha tanımlamaktadır. Bunlar; hayali oluşumlu ve teğetsel oluşumlu köşe tipleridir.

Watanabe ve arkadaşları, aşağıdan-yukarı yaklaşımlarında ortaya çıkan görünüş tutarsızlıklarını ele aldıkları çalışmalarında, girdi görünüşlerindeki kullanıcı hatalarını veya çizim programlarından kaynaklanan yanlış anlamaları bulmaya çalışmışlardır [33]. Algoritma ile 3B nesne oluşumu sırasında çizgi türlerinin (görünür veya görünmez) yanlış seçilmesi, fazla veya eksik çizimden kaynaklanan tutarsızlıklar test edilmektedir. Test sonrası tutarsızlıkların kaynağını gösteren ve tutarlı görünüşler öneren bir yöntem önerilmektedir. Yöntemde girdi görünüşlerinde oluşan hatalar, elde edilen 3B'lu nesnelerin yeniden görünüşlerinin çıkartılması ve ilk girdi ile karşılaştırılarak belirlenmektedir.

Tanaka ve arkadaşları, diğer araştırmacılardan farklı olarak 2B'lu montaj çizimlerini ele almışlardır [34]. Çalışmada 2B'lu montaj resimleri ayrıştırılarak her bir parçanın 3B'lu nesne oluşumları incelenmektedir. Algoritma; düzlemsel, silindirik, konik ve küresel yüzleri kapsamakta, üç görünüşle ifade edilen parçaları işleme almaktadır. Tanaka ve arkadaşları, nokta, kenar ve yüz oluşturma işleminde koordinatsal eşitlikleri kullanmaktadır. Nokta ve kenar tanımlamada, Markowsky ve Sakurai'nin metodları kullanılmaktadır. Çalışmada çoklu çözüm oluşturabilmek için katı eleman eşitlikleri geliştirilmiştir. Böylece muhtemel çözümlerin hepsi oluşturulabilmektedir. 3B'lu nesnelerin doğruluğu 2B'lu izdüşümleri ile karşılaştırılarak test edilmektedir. İşlem sırasında çizim hataları da dahil olmak üzere, hatalı işlemler karşısında kullanıcı uyarılmaktadır. Araştırmacılar aynı konudaki daha güncel bir çalışmalarında, yöntemi iki görünüşle ifade edilen parçaların 2B'lu montaj resimlerine uygulamıştır [35]. Algoritma işlem adımları aşağıdaki şekildedir:

- 2B montaj çizimi
- 2B'lu montaj çizimlerini 2B'lu parça çizimlere ayrıştırma
- 2B'lu parça çizimlerinden blokları tanımlama (3B'lu)
- Her bir parça için doğru blok seçimi
- 3B'lu montaj çizim çıktısının elde edilmesi

Burada ayrıştırılan her bir parça için doğru blok seçimi yapılmakta ve 3B'lu montaj elde edilmektedir (Şekil 2.6). Yöntemin standart parçalarla sınırlı olması önemli bir dezavantajdır.



Şekil 2.6. 2B montaj resimlerinden 3B katı modellerin oluşturulması [35]

Kuo'nun çalışması dörtgen yüzeyli parçaları ele almaktadır [36]. Geliştirilen algoritmanın işlem adımları şu şekildedir:

- 2B'lu üç görünüş girdisinden tel kafes model oluşturma
- Minimum iç açılar arama metodu ile aday yüzeyler bulma
- Yüzeylerin birleştirilmesiyle katı model oluşturma.

Yaklaşım, çoklu çözümler vermektedir.

Araştırmacı, daha sonra matematiksel bir yöntemle düzlem yüzeyli nesnelerin katı modellerini oluşturmuştur [37]. Daha önce geliştirdiği minimum iç açılar arama yöntemi ile üç görünüşteki bütün düzlem yüzeyleri çıkartıp tel kafes modellerini oluşturmaktadır. Burada istenmeyen dış döngüler engellenerek iç döngüler tek aşamada oluşturulmaktadır. Çalışmada girdi izdüşüm eksenlerine paralel olma sınırlılığına sahiptir.

Dori ve Weiss'in çalışmasında 3B model nesne dönüştürme ağları ile oluşturulur [38]. Bu işlem için ilk olarak 2B'lu girdi boyutlarla açıklanmaktadır. Daha sonra bu boyutlar belirli sınırlar içinde değerlendirilerek birer görünüşe çevrilmekte ve her bir görünüş için 2B minimal grafikler elde edilmektedir. Bu grafikler, çoklu geometri, matris, cebir ve grafik teorileri bir araya getirilerek oluşturulmaktadır. Daha sonra

görünüşler arası ilişkiler kurularak, izdüşümlere ait bileşik bir ağ oluşturulmaktadır. Bu ağ yapısal ve topolojik ilişkileri belirlemektedir. Son işlem, nesne dönüştürme ağı kullanılarak 3B nesnenin elde edildiği işlemdir. Bu yaklaşım, temel katı elemanlarının tüm birleşimlerini incelemede yeterli değildir. Ayrıca önerilen yöntem, eksik görünüşleri belirleyememektedir.

Wei, eğimli yüzeylerin 3B tanımlamalarına ilişkin yeterli ve gerekli şartları bulmaya çalışmıştır [39]. Metod, dört serbestlik dereceli çok yüzlülere uygulanmakta ve tek görünüş girdisi kullanılmaktadır. Ayrıca, yüzeyler 3B'lu olarak tanımlanmaktadır. Wei'nin metodu serbestlik derecesinin sayısındaki sınırlılıktan dolayı sıkıntılıdır. Algoritma, bir çokyüzlüyü temsil ettiği bilinmesine rağmen, bazı durumlarda çizimleri ayıran ve orijinal çizimi birkaç ayrı nesne halinde gösteren bir yaklaşımı içermektedir [9]. Yöntem, eğri yüzeyli nesnelerin çizimlerine uygulanamamaktadır.

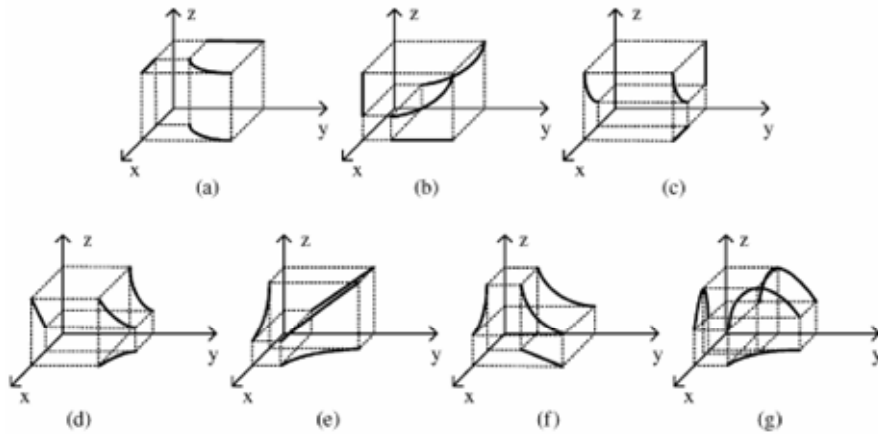
Wei ile benzer şekilde tek görünüş girdisi kullanan Sugihara, 2B çizimlerden nesnelerin 3B yapılarını çıkartmak için matematiksel bir yöntem kullanmıştır [40]. Çok yüzlülerin ele alındığı çalışmada uygun girdiye ihtiyaç duyulmaması diğer çalışmalarla arasındaki en önemli farktır [9]. Ayrıca girdinin elle yapılabilir ya da dijital görüntülerden elde edilebilmesi yönüyle farklı bir çalışmadır. Yöntem ayrıca, görünmez çizgileri de ele almaktadır. Algoritma dört temel aşamayı içerir [4]:

- a. Huffman ve Clowes'in etiketleme şemasına benzer bir şekilde görünüşlerdeki izdüşümlere ait geometrik bilgileri çıkartma.
- b. Doğru ve yanlış yorumlar arasında ayırım yapma. Bu aşamada gerekli ve yeterli koşullar matematiksel olarak tanımlanmaktadır.
- c. Bu adımda, gereksiz nesneler tespit edilerek ortadan kaldırılmaktadır.
- d. Son adımda 3B'lu nesneler belirlenmektedir.

Lamb ve Bandopadhyay, 3B yapıları düzgün olmayan 2B çizimlerden çıkarmak için bir sistem sunmuşlardır [41]. Algoritmada girdi olarak izometrik ve axonometrik çizimler kabul edilmektedir. Burada Waltz'ın etiketleme yaklaşımı kullanılarak, bir

bitişik grafik ile (örneğin köşe-kenar grafiği gibi) çizim etiketlenir. Algoritmanın amacı bütün köşeler için 3B koordinatlar atamaktır. Çalışmada, 3B nesnelerin tam olarak yeniden yapılandırılması sağlanamamıştır. Ayrıca, İki düzlemin paralel olup olmadığını ve birbiri ile simetrik olup olmadığını belirlemeden kaynaklanan problemler vardır.

Gong ve arkadaşları eğri köşelerin tel kafes modellerini oluşturmaya yönelik bir çalışma yapmışlardır [42]. Çalışmada doğrusal kenar türleri ve Bezier eğrilerinden yararlanılarak konik kenar türleri tanımlanmaktadır (Şekil 2.7). Düzlemsel ve konik bağlantı türlerine göre geliştirilen karar ağaçları ile eğri türüne karar verilmektedir. 3B gösterimin tel kafes olması, dürtgensel ve eğri yüzeyli katıların otomatik oluşturulmaması çalışmanın sınırlı yönleridir.



Şekil 2.7. 3B konik kenar çeşitleri [43]

Araştırmacılar daha sonra karma bir modelle düzlemsel yüzeylerin katı modellerini oluşturan bir çalışma yapmışlardır [43]. Metod 2B izdüşümlerin tanınmasında çoklu graf temsillerini kullanmaktadır. Burada, 2B'lu kenar ve köşeler arasındaki düzlemsel yüzeylerin topolojik ve geometrik ilişkilerinden yola çıkarak karma bir tel kafes model oluşturulmaktadır. Geliştirilen yöntemde işlem zamanı azaltılmıştır. Fakat burada, doğru sonuç için girdinin mükemmel olması istenmektedir.

Shpitalni ve Lipson, tel kafes olarak temsil edilen nesnelerdeki kenar döngülerinin belirlenmesinde ortaya çıkacak karmaşıklıklara bir çözüm olarak yüzey belirleme

yaklaşımını önermiştir [12]. Çalışmada, bu karmaşıklığı ortadan kaldırmak için değişken olmayan nesnelere uygulanabilen, yüzey-kenar-köşe ilişkileri de önerilmektedir. Yöntem, şekillerin karmaşıklık düzeyine bağlı olarak işlem ve sürenin uzun olması dezavantajına sahiptir.

Oh ve Kim, eğri yüzeyli nesnelerin katı modellerini oluşturmaya yönelik çalışmalarında iki görünüş girdisi kullanmışlardır [44]. Çalışmada ilk olarak görünüşlerdeki her bir kapalı alana karşılık gelen sınırlı yüzeyler elde edilir, daha sonra bu yüzeylerden alt nesneler oluşturulur. Katı model bu alt nesnelerin bir araya getirilmesiyle oluşturulur. Yöntem en uygun çözümü tercih edebilmek için çoklu çözümleri içermektedir. Algoritma ile konik, küresel ve silindirik yüzeyli nesnelerin katı modelleri oluşturulabilmektedir.

Zhang, konik parçaların katı modellerini oluşturan bir yöntem geliştirmiştir [45]. Burada cebirsel bir yöntemle bütün görünüşlerdeki 2B izdüşümlerin temsilleri tanımlanarak gerekli ilişkiler kurulmakta, daha sonra bütün cebirsel ilişkiler analitik olarak geometrik şekilleri karşılayacak tarzda dönüştürülmektedir. Konik katının yeri üç görünüşteki ilgili koniğin geometrik konumundan sağlanır. Son olarak gereksiz elemanlar silinerek tel kafes model oluşturulur. Çalışmada yüzey ve eksen sınırlaması yoktur.

Şahin ve Börklü yaptıkları çalışmada üç görünüşle ifade edilen prizmatik parçaların katı modellerini oluşturan bir çalışma yapmışlardır [46]. Aşağıdan yukarı yaklaşımı esas alan çalışmada kullanıcı etkileşimi olmaksızın silindirik unsurlar da içeren (silindirik çıkıntı ve delik vb.) prizmatik parçaların katı modelleri oluşturulmaktadır. Çalışma tek kalınlıklı nesnelerle sınırlıdır.

Liu, konik parçaların eksenler üzerinde olma şartını kaldırdığı çalışmasında koniklerin afin özellikleri ile geometrik özelliklerini bir araya getirerek katı modellerini oluşturmaktadır [47]. Fakat bu işlem oldukça uzundur. Koniklerin farklı

tipleri için gerekli olan bilginin çıkarılma aşamasının karmaşık ve uzun zaman gerektirmesi yöntemin en büyük sıkıntısıdır.

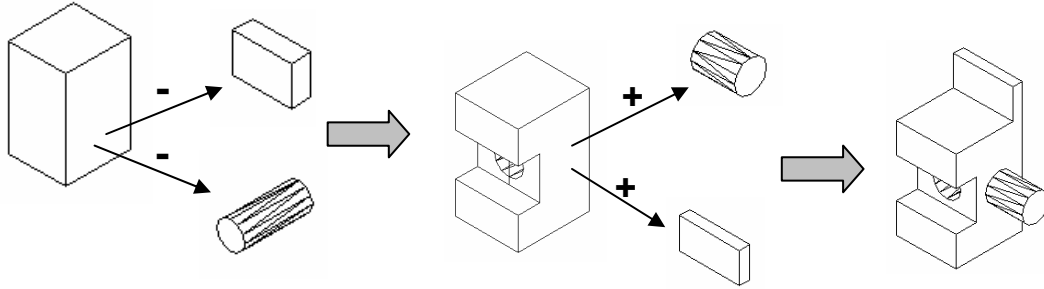
2.4. Hacime Dayalı Yaklaşımlar

Hacime dayalı yaklaşımlar, tel çerçeve olanlardan genelde daha etkindir, ancak karmaşık çizimlerden özel temel eleman biçimleri tanımak oldukça zordur. Bu tür yaklaşımlar, genelde aynı kalınlıkta veya dönel parça çizimlerine uygulanır. Hacime dayalı yaklaşımlarda çizimler temel eleman kümeleri olarak tanınmaktadır. Makine çizimlerinden unsur tanıma, hacim uyarlı yaklaşımlara benzemektedir. Temel eleman kümelerinin (YKG) ve son katı modelin oluşturulmasında üç farklı yöntemin kullanıldığı göze çarpmaktadır. Bunlar aşağıdaki şekilde sayılabilir;

- Klasik YKG yaklaşımını kullanan çalışmalar,
- Süpürme operasyonlarını kullanan çalışmalar,
- Hacimsel kesiştirme yaklaşımını kullanan çalışmalar.

2.4.1. Klasik YKG yaklaşımını kullanan çalışmalar

Klasik YKG yaklaşımı olarak tanımlanan yöntemde katı model YKG temel elemanları ile oluşturulmaktadır. Bunlar koni, silindir, küp, küre vb. ilkellerdir. Bu çalışmalarda 2B izdüşümler arası ilişkilerden YKG ilkelleri oluşturulmakta, daha sonra ilkellerin birbirleri ile ilişkilerinin belirlendiği YKG ağacı yardımıyla boolean operasyonlarına karar verilmektedir. YKG ilkelleri arasında yapılan Boolean operasyonları (birleştirme ve çıkartma) sonucu katı model oluşturulmaktadır (Şekil 2.8).



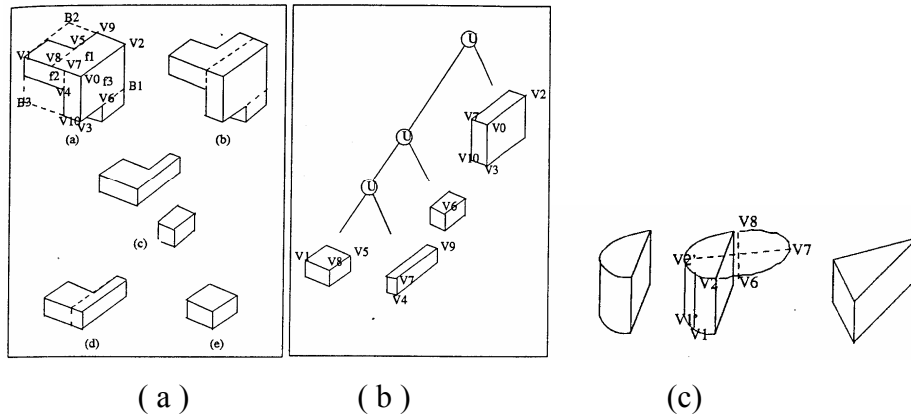
Şekil 2.8. Klasik YKG yaklaşımı ile katı model oluşturma

Aldefeld'in YKG elemanlarının belirlenmesi ve birleştirilmesine dayanan çalışmaları hacime dayanan çalışmaların öncüsü kabul edilebilir [48, 49]. Çalışmada, üç görünüş olarak girilen 2B izdüşümler yorumlanarak temel yapılandırma elemanları (YKG ilkeleri) elde edilmekte, daha sonra bu elemanlar birleştirilerek nihai katı model oluşturulmaktadır. Yarı otomatik bir yapıyı içeren yöntemde temel elemanların her biri ayrı ayrı ve bir bütünü oluşturacak şekilde tanımlanmaktadır. YKG ilkelinin doğru oluşturulabilmesi için hatalı eleman tanımlanmamalıdır. Kullanılan temel elemanlar; koni, küp, silindir ve altı yüzlülerden oluşmaktadır. Algoritma üst üste gelen parçaları yorumlayamamakta ve önceden tanımlı nesnelerle sınırlı olduğu için kısıtlı çözüm vermektedir. Ayrıca, kullanıcı etkileşimine ihtiyaç vardır.

Mekhabunchakij Aldefel'in yaklaşımını kullanarak 2B girdilerden temel elemanları, daha sonra bunların YKG ağaçlarını tanımlamaktadır [50]. Burada global sınırlandırıcılarla temel elemanların gerçek elemanlar olup olmadığı kontrol edilir. Sistem aynı kalınlıktaki nesneleri kapsamaktadır.

Ho, Aldefeld'in çalışmasından yola çıkarak bireysel parçaların ilişkilerini kullanmış ve YKG benzeri bir yapılandırma algoritması geliştirmiştir [51]. Çalışmada kullanıcı etkileşimi Aldefeld'in çalışmasından daha azdır. Temel elemanlar kullanıcı tarafından belirlenmektedir. Kullanıcı temel elemanın türünü girmekte, ayrıca bu temel elemanın nihai katının oluşumuna katkısının olup olmadığını da belirtmek zorundadır. Bunun yanında temel elemanın esas yüzeyi ve yüksekliği de kullanıcı tarafından girilmektedir. Ho'nun çalışması kullanılan temel eleman türünü arttırmakla beraber yoğun kullanıcı etkileşimine ihtiyaç duyması açısından sınırlılıklar içermektedir.

Wang ve Grinstein, bir çok yüzlüden temel elemanları otomatik olarak çıkartan YKG benzeri bir metod geliştirmiştir [52]. Metod doğrusal çok yüzlülere uygulanmıştır ve köşelerin üç yüzü birbirine dikey varsayılmıştır. Araştırmacılar çizimleri etiketleme şemaları ile etiketlemektedirler. Katı temel elemanlar köşe birleştirme türlerine göre tanımlanırlar. Çizgi etiketleri yapılacak boolean operasyonunun türünü de belirlemektedir. Şekil 2.9.(b) metod ile elde edilen YKG ağacını göstermektedir.



Şekil 2.9. Wang'ın YKG ağacı [52, 53]:

- a) işlem aşaması
- b) Elde edilen YKG ağacı
- c) yarı doğrusal YKG elemanlar

Wang daha sonra metodunu doğrusal parçalardan yarı doğrusal parçalara genişletmiştir [53]. Çalışmada silindirler işleme sokulmuştur. Ayrıca eğri temel elemanlar tanımlanmıştır. Doğrusal algoritma içindeki temel bloklar iç bükey (konvex) prizma olarak ele alınmıştır (Şekil 2.9.(c)).

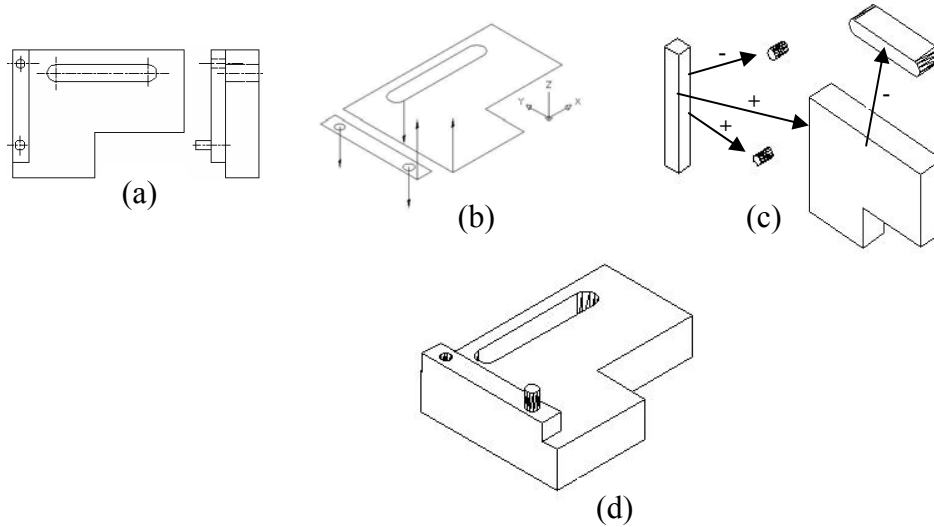
Wang'ın çalışmalarının en önemli farkı tek görünüş girdisi kullanması ve diğer araştırmacıların aksine girdi olarak 3B'lu çizimleri kullanmasıdır. Ayrıca bu çalışmalarda katı model oluşturmamış, katı model oluşturma için gerekli YKG ağaçlarını çıkartmıştır.

Araştırmacılar daha sonra bu YKG ağaçlarını kullanarak 3B modeller elde etmiştir [54]. Çalışmada, YKG ağaçlarından elde edilen temel elemanlar birleştirilerek 3B

nesne elde edilmiştir. Ancak burada da birden fazla çözüm ve kullanıcı etkileşimi vardır.

2.4.2. Süpürme operasyonları kullanan çalışmalar

Bu başlık altında incelenen çalışmalarda temel elemanlar düzlemsel ve silindirik kapalı bölgelerin normal vektörleri doğrultusunda doğrusal ve eksenal olarak süpürülmesi ile oluşturulmaktadır. Bunun yanında YKG ilkeleri de kullanılmaktadır. Süpürme operasyonları ile temel eleman tanımlama yöntemi daha fazla temel eleman tanımlanabilmesini sağlamıştır. Özellikle eğri yüzeyli parçaların katı modellerinin oluşturulmasında büyük kolaylıklar sağlamıştır. Bu yöntemin ana işlem adımları süpürme operasyonları ile katı temel elemanlar oluşturma ve boolean operasyonları (ekleme ve çıkartma) ile katı model oluşturmadır (Şekil 2.10).



Şekil 2.10. Süpürme operasyonu ile katı model oluşturma:

- a) 2B izdüşüm
- b) süpürme yönleri
- c) katı temel elemanlar
- d) katı model

Chen ve Perng, bu yöntemi kullanarak katı model oluşturma problemine üç aşamalı bir çözüm önermektedir [55]. Bu aşamalar: ayrıştırma (izdüşümdeki temel elemanları tanıma), yeniden yapılandırma (bunlardan 3B temel nesneler elde etme) ve

düzenlemedir. Chen'in algoritması farklı kalınlıktaki çok yüzölçümleri kullanıcı etkileşimli ele almayı mümkün kılmaktadır.

Nagasami çalışmasında bilgi tabanlı bir teknik kullanarak bilgisayar ve elle yapılan çizimlerin katı modellerini oluşturmaktadır [56]. Çalışmada pek çok YKG temel elemanlarından oluşan karmaşık parçalar da ele alınmaktadır. Farklı görünüşlerdeki temel elemanların unsur özellikleri kullanılarak alt nesneler tanımlanmaktadır. Burada, çıkıntı ve delik unsurları tanıyan ve katı modelini oluşturan bir sistem geliştirilmiştir.

Hosomura, YKG yaklaşımı kullandığı çalışmasında katı modeli yedi aşamalı bir algoritma ile oluşturmaktadır [57]. Bunlar, 2B girdi, DXF veri dosyası oluşturma, süpürme operasyonları ile YKG temel elemanları oluşturma, temel elemanlar arasındaki ilişkileri kullanarak katı model oluşturma şeklinde belirtilebilir.

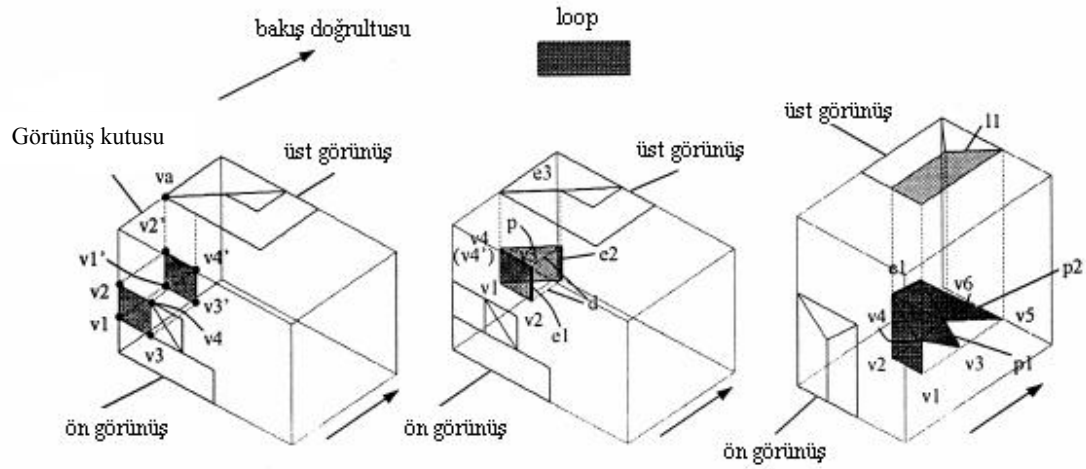
Dutta ve Srivinas'ın algoritması, iki görünüş girdisi kullanmakta, düzlemsel ve eğri kenarları ele almaktadır [58]. Çalışmada öncelikle verilen iki görünüşten yararlanarak üçüncü görünüşler oluşturulmaktadır. Daha sonra üç temel görünüşe karşılık gelen tüm muhtemel 3B katılar elde edilmektedir. Bir sonraki aşamada dairesel yayların (köşe kavisi vb.) yerleştirileceği doğrular belirlenmektedir. Son işlem değiştirilen görünüşlerden 3B nesneler oluşturmaktır. Algoritmada verilen görünüşlere karşılık gelen tüm katı kümeleri belirlenmektedir. Eğer katı kümesi boş bir küme ise görünüşler uyumsuzdur. İşlem, tüm geçerli yüz oluşumlarını dikkate almaktadır. Algoritmada üçüncü görünüşü tamamlama işlemi için kullanıcı katkısı gerekmektedir.

Wu ve Lin'in çalışmasında girdi tek görünüş, IGES formatında ve izometriktir [59]. Çalışmada 2B izdüşümün görülebilir yüzeyleri bulunmakta ve süpürme operasyonları ile olası bütün temel katılar oluşturulmaktadır. Katı model temel katılar arasında gerçekleştirilen birleştirme operasyonları sonucu oluşturulur. Algoritmada ele alınan nesne türleri basit doğrusal çok yüzölçümlerle sınırlandırılmıştır.

Tam ve Atkinson silindirik ve prizmatik parçaların katı modellerini oluşturdukları çalışmalarında, önce parçanın simetri kontrollerini yapmaktadır [60]. Simetri kontrolleri yapılan girdilerin türüne göre doğrusal ve eksenel süpürme operasyonları uygulanarak katı temel elemanlar oluşturulmaktadır. Son işlem temel ve ana katı arasında ekleme ve çıkarma operasyonları ile katı modelin oluşturulmasıdır. Süpürme operasyonlarının etkin olarak kullanıldığı çalışmada bezier eğrileri de ele alınmaktadır.

Suh ve arkadaşları çok yüzlü nesneleri ele aldıkları çalışmalarında girdi olarak iki görünüşle ifade edilen izdüşümleri almıştır [61]. Çalışmada, gösterimi belirsiz iki görünüşlü izdüşümlerin araştırma sayısı azaltılmaya çalışılmıştır. Yöntem iki aşamalıdır. İlk aşama, iki görünüş girdisi üzerindeki kapalı alanlara uyan aday yüzlerden aday nesnelerin oluşturulmasıdır. İkinci aşama, bu nesnelerin geometrik doğrulukları ile birbirlerine eklenmesidir.

Kim ve Zhao, çok yüzlü nesnelerin oluşturulması ile ilgili bir çalışma yapmıştır. İki görünüş girdisi kullanılan çalışmada eksik görünüşler tamamlanabilmektedir [62]. VRT (Visual Reasoning Tutor) ismi verilen sistemle özel olarak mühendislik öğrencilerine görünüşler arası ilişkileri ve bu görünüşlerin 3B'lu karşılıklarını göstermek hedeflenmiştir. Sistem bir görünüş kutusu (3B'lu) üzerine yerleştirilen 2B izdüşümlerin normal vektörleri doğrultusunda süpürülmesi esasına dayanır. Her iki görünüşün süpürülmesi sonucu elde edilen ortak bölgesi son katıyı verir. Üç türlü süpürme operasyonu vardır; yüzey, kenar ve köşe süpürme operasyonları. Çalışmada iki görünüşü verilen izdüşümlerin üçüncü görünüşleri de bulunmaktadır (Şekil 2.11).



Şekil 2.11. VRT’de süpürme operasyonu [62]:

- a) yüzey süpürme
- b) kenar süpürme
- c) köşe süpürme

Hubbard ve Kim, Kim ve Zhao’nun çalışmasının dairesel ve eliptik yayları içine alacak şekilde genişletmiştir [63]. İki görüş girdisi kullanılan çalışma makine parçalarında bulunan silindirik ve küresel yüzeylerin katı modellerini oluşturulmaktadır. Araştırmacıların ARCS (Asistant for Reasoning and Construction of Solids) ismini verdikleri sistemde, katı model oluşturma süreci daha önceki çalışmanın benzeridir. Hazırlanan kullanıcı arabirimi ile kullanıcının farklı eğri yüzeyleri oluşturması sağlanmıştır. Sistem düzlemsel ve silindirik iki ana birimden oluşur. Katı model, 2B izdüşümlerden kullanıcı destekli olarak model alanlarının tespit edilmesi ve bu bölgelerin bulundukları izdüşüm bölgelerinde normal vektörleri doğrultusunda süpürülmesi ile oluşturulur.

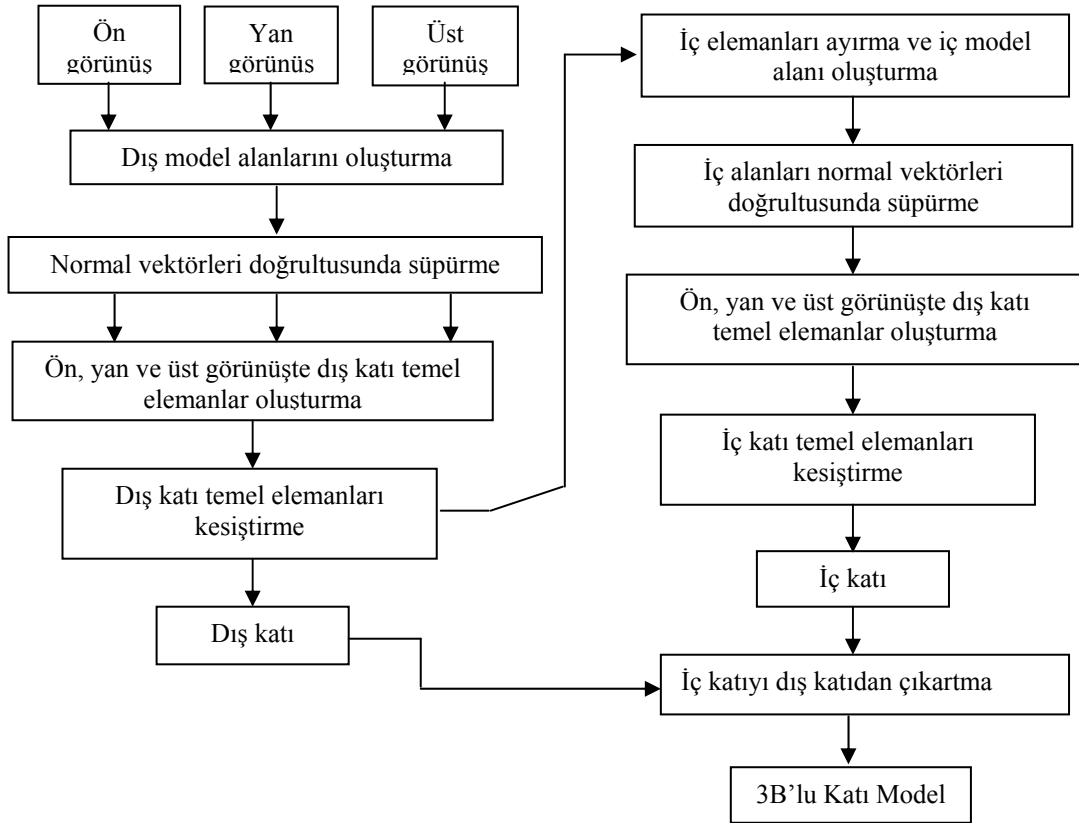
Dimri ve Gurumoorthy, kesit görüşleri ele alan bir çalışma yapmıştır [64]. Çalışmada girdi görüşlerinin yorumlanması ile oluşturulan kapalı kenar kümeleri süpürme operasyonları ile katı temel elemanlara dönüştürülür. Daha sonra temel elemanlar arasında boolean operasyonları uygulanarak katı model oluşturulmaktadır. Çalışmada tam ve yarım kesitlerle ifa edilen parçalar ele alınmaktadır.

2.4.3. Hacimsel kesiştirme yaklaşımı kullanan çalışmalar

Bu yaklaşımda temel katılar süpürme operasyonları ile elde edilmektedir. Süpürme operasyonları ile katı model oluşturma yönteminden en önemli farkı temel katı ve son katı oluşturma aşamalarında “kesiştirme” operasyonlarının uygulanmasıdır.

Shum tarafından geliştirilen yaklaşım iki aşamalı bir yapı arz etmektedir [9]. Bu iki aşamalı yapıda katı temel elemanlar dış ve iç katı elemanlar olmak üzere ikiye ayrılmaktadır. Sırasıyla önce dış katı temel elemanlar izdüşüm normal vektörleri doğrultusunda süpürülerek oluşturulmaktadır. Bu şekilde her bir görünüş bölgesinde dış katılar oluşturulur. Daha sonra görünüş bölgelerindeki dış katılar arasında hacimsel kesiştirme operasyonu gerçekleştirilerek nihai dış katı oluşturulmaktadır. İç katılar, dış katıya benzer şekilde oluşturulur. Katı model dış ve iç katı arasında çıkartma işlemi sonunda elde edilir (Şekil 2.12).

Shum bu yaklaşımda ilk olarak altı görünüş girdisi kullanmıştır [9]. Çalışma sadece görünebilir kenarları ele almaktadır. Araştırmacılar daha sonra görünmeyen kenarları katı model oluşturma sürecine dahil etmiş aynı zamanda girdi görünüş sayısını üçe düşürmüştür [65]. Bu çalışmada ayrıca, ilk çalışmada yer almayan iç unsurlar (cep, iç kademe v.b.) ve görünmeyen kenarlar da ele alınmıştır.



Şekil 2.12. Hacimsel kesiştirme yaklaşımı ile katı model oluşturma [65]

Shum ve arkadaşları tarafından yapılan başka bir çalışmada 2B'lu girdi X-ray cihazı tarafından elde edilmiştir [66]. X-ray cihazı ile film şeritlerine düşürülen nesnenin 2B'lu görüntüleri (ön, yan ve üst görünüş olarak) resim formatına dönüştürülmektedir. CorelDRAW programı ile vektörel dönüşümleri yapılan girdi görüntüleri 2B'lu çizimlere dönüştürülmektedir. Daha sonra 2B'lu girdinin hacimsel kesiştirme yaklaşımı ile katı modeli oluşturulmaktadır. Yöntem alternatif girdi metodu sunması ile alana yenilik katmakla birlikte, X-ray cihazı aracılığı ile elde edilen görüntü ölçülerinin orijinal nesne ölçülerini karşılaması zordur. Ayrıca, çalışma sadece düzlemsel çok yüzlüleri ele almaktadır.

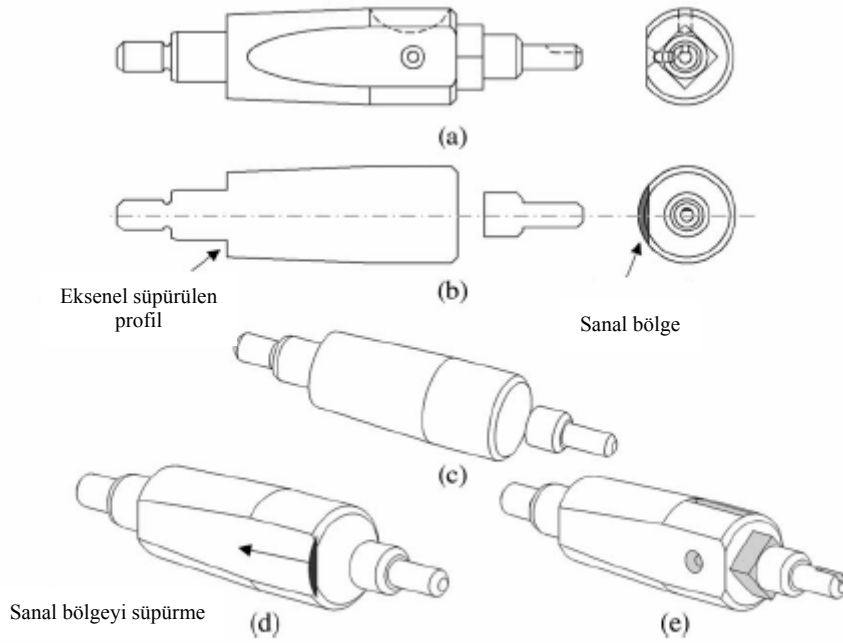
Shum ve arkadaşlarının hacimsel kesiştirme yaklaşımını esas alan değişik çalışmalarda girdi görünüş sayısının azaltılmaya, iki aşamalı işlemi tek aşamada gerçekleştirmeye ve ele alınan unsur tipini arttırmaya yönelik algoritmalar geliştirilmiştir [67 - 69].

Şahin ve Börklü'nün çalışmaları işlemi tek aşamada gerçekleştirmektedir [67, 68]. Çalışmalarda, Shum'dan farklı olarak görünüş bölgelerinde ana ve temel katılar arasında ekleme çıkartma operasyonları ile ön katılar daha sonra ön katılar hacimsel kesiştirme işlemine tabi tutularak katı model oluşturulmaktadır. Geliştirilen algoritma prizmatik parçalara hitap etmekte ve üç görünüş girdisi kullanmaktadır. Kademeli parçalarda ortaya çıkan yüzey farklarından ötürü yanlış nesneler oluşturabilmesi ve prizmatik parçalarla sınırlı olması yönüyle kısıtlıdır.

Çiçek ve Gülesin'in çalışması da Shum'un yöntemini temel almaktadır [69]. Çalışma temel makine unsurlarının katı modellerini oluşturmayı hedeflemektedir. Araştırmacılar, hacimsel kesiştirme yaklaşımını kullanarak makine unsurlarının (genelde delik) çoğunun katı modellerini oluşturmuştur. Çalışmada üç görünüş girdisi kullanılmıştır.

IGES formatındaki girdileri kullanan Liu ve arkadaşları, 2B girdilere ait köşe girdilerini çıkartmak için böl ve yapılandır stratejisi kullanılmıştır [70]. Çalışmada unsur eşleştirme işlemi ile 3B unsurların tanınması amaçlanmaktadır. Bu amaçla bir takım üretim kuralları oluşturulmuştur. Katı model, süpürme operasyonları ve hacimsel kesişme yaklaşımları kullanılarak oluşturulmaktadır.

Lee ve Han, silindirik elemanların eksenel süpürme yöntemi ile katı modellerinin oluşturulmasına dayanan bir algoritma geliştirmiştir [71]. Burada öncelikle 2B girdilerin izdüşümleri yorumlanarak silindirik parçanın simetrik olup olmadığına karar verilmektedir. Eğer simetrik ise 2B'lu profil kendi eksenine etrafında süpürme işlemine tabi tutularak katı model oluşturulur. Eğer simetrik değilse, simetrikliği bozan bölgelerde sanal bölgeler oluşturularak parça simetrik hale getirilmekte, daha sonra eksenel süpürme operasyonu ile ana katı oluşturulmaktadır. Sanal bölgelerden oluşturulan katı temel elemanlar ana katıdan çıkartılarak katı model oluşturulur. Çalışma, dönel simetrik elemanlarla sınırlıdır (Şekil 2.13).



Şekil 2.13. Silindirik elemanların hacimsel kesiştirme yaklaşımı ile katı modellerinin oluşturulması [71].

Şahin ve Börklü geliştirdikleri hacimsel yaklaşımla silindirik ve prizmatik parçaların katı modellerini oluşturmuşlardır [72]. Çalışma üç görünüş girdisini kullanmaktadır. Algoritma ana adımları; 2B izdüşümler yorumlanarak görünüş bölgelerinde temel katı elemanlar oluşturma, bu katı elemanlar arasında boolean operasyonları yaparak görünüş bölgelerinde ön katılar oluşturma, ön katıların hacimsel kesiştirilmesi ile katı model oluşturma. Kullanıcı etkileşimi olmayan çalışmada silindirik parçaların katı modellerini oluşturmada problemler vardır.

Şahin ve Börklü bir diğer çalışmalarında iki görünüşle ifade edilen silindirik elemanların katı modellerini oluşturmaya yönelik bir çalışma yapmıştır [73]. Çalışmada tek görünüşle ifade edilen dönel simetrik elemanların katı modelleri de oluşturulmaktadır. Çalışma ile tek ve iki görünüşle ifade edilen simetrik ve simetrik olmayan parçaların katı modelleri oluşturulmaktadır.

2.5 Karma Yaklaşımlar (YKG-BRep)

Kaynaklarda hacme dayalı ve tel kafes yöntemlerin yanında her iki yöntemi bir arada kullanan çalışmalara da rastlanmaktadır. Bu çalışmalarda katı model oluşturma sürecinin ilk aşamaları (3B nokta, kenar ve yüzey oluşturma) aşamaları tel kafes yaklaşımıyla gerçekleştirilmektedir. Daha sonra tel kafes yaklaşımıyla oluşturulan yüzeylerden tel kafes bir model oluşturulmaktadır. Temel katılar tel kafes modeli oluşturan yüzeylerden yapılandırılmakta ve boolean operasyonları ile katı model oluşturulmaktadır.

Masuda ve Numao'nun değişken olmayan topoloji ve kabule bağlı doğru bakım sistemine (ATMS) dayalı çalışması karma bir yapı içermektedir [74]. Yöntem, tüm katı adaylarını değişmeyen topoloji kullanan hücresel bir modelle kontrol etmektedir. Çalışmada önce Idesawa ve Wesley'in metoduna benzer bir şekilde tel kafes model elde edilmektedir. Tel kafes modeldeki kenar döngülerinin araştırılması ile yüzey model oluşturulmaktadır. Yüzey modelden hücresel model oluşturulmaktadır. Hücresel modeldeki hücre kombinasyonları bir katının biçimini tanımlamaktadır. Bu tanımlamada temel görünürlere karşılık gelen katılar, hücre ve görünürlükler arasındaki sınırlayıcıların çözülmesinden oluşturulmaktadır. Son olarak Boolean operasyonları (çıkartma) kullanılarak katı model oluşturulmaktadır.

Wang ve Latif'in çalışmasında ilk olarak katı modelin yarı tel kafes modeli oluşturulur [75]. Daha sonra aday katı modelleri oluşturmak ve yaratmak için YKG yaklaşım kullanılmaktadır. Çalışmada yarı-tel kafes olarak yeni bir kavram tanımlanmaktadır. Bu yöntem işlem zamanının azaltmakta ve model oluşturma işlemini kolaylaştırmaktadır.

Geng ve arkadaşları Idesaewa'nın aşağıdan yukarı yaklaşımını kullanarak 3B kenarları oluşturmaktadır [76]. Daha sonra bu kenarların birleşimlerinden süpürme operasyonu uygulanacak profiller elde edilir. YKG elemanlar bu profillerden

oluşturulur. Katı model, YKG elemanlara boolean operasyonları uygulanması ile oluşturulur. Çalışma kullanıcı katkısına ihtiyaç duymaktadır.

2.6. Değerlendirme

Bu çalışmada, şimdiye kadar yapılan sınıflandırma yöntemlerinden farklı olarak katı model oluşturma işlemi kullanılan temsil tekniklerine göre üçe ayrılmıştır. Bunlar;

- tel kafese dayalı (Sınır temsili)
- hacime dayalı (YKG) ve
- karma yöntemler

Daha önceki çalışmalarda bu sınıflandırma tel kafese ve hacime dayalı çalışmalar olmak üzere iki ana başlık altında incelenmiştir. Fakat son yıllarda CAD teknolojisindeki gelişmeler, bu iki temsil tekniğini bir arada kullanmayı mümkün kılmıştır. Buna paralel olarak otomatik katı model oluşturma çalışmalarında çok yeni de olsa bu iki yöntem bir arada kullanılmaya başlanmıştır. Bu yeni sınıflandırma yöntemi ile çalışmaların daha sistematik bir şekilde ele alınması mümkün olacaktır.

Bunların yanında bugünkü çalışmalara temel teşkil eden bazı öncü çalışmalara da değinilmiştir. Bu çalışmalar daha çok 3B nesne bilgisi elde etmeye yönelik ve tek görünüş girdisi kullanan çalışmalardır [7, 8, 10 – 14]. Tek görünüş girdisi nesne oluşumunda bazı belirsizliklere neden olması nedeni ile günümüz yeniden yapılandırma çalışmalarında fazla kullanılmamaktadır. Bazı çalışmalarda kullanıldığı görülse bile bunlarda girdi izometrik [39, 59] veya 3B'ludur [52 - 54]. Bu çalışmalarda kullanıcı etkileşiminin oldukça fazla olması da önemli bir dezavantajdır.

Tel kafes ve sınır temsili model elde eden çalışmalar tel kafese dayalı çalışmalar başlığı altında incelenmiştir. Çünkü bu temsil tekniklerinin her ikisi de Idesawa'nın aşağıdan-yukarı yaklaşımına dayanmaktadır. Tel kafes yaklaşımlar çoklu çözüm

hallerinde ve karmaşık şekilli parçalarda başarılı sonuçlar vermektedir. Bu yaklaşımda hemen bütün yüzey biçimlerinin 3B nesne oluşumları gerçekleştirilebilmektedir. Fakat bunun yanında aşağıda belirtilen eksiklikler hala giderilememiştir:

- Tel kafes yaklaşımlarda izdüşümlerde anlam ifade eden ancak nesne topolojik özelliklerini karşılamada yetersiz bazı kenarlar oluşmaktadır,
- Özellikle eğri yüzeyli parçalarda uzun işlem zamanları gerekmektedir,
- Hatalı kenar oluşumları 3B oluşturma etkinliğini azaltmakta ve farklı yorumlamalara sebebiyet vermektedir
- Ayrıca, model kontrolü de güçtür.

Hacim tabanlı (veya YKG) yaklaşımlar sınır temsili yaklaşımlara oranla yeni bir alandır. Fakat son yıllarda artan sayıda çalışmalar mevcuttur. Hacim tabanlı çalışmalar için iki veya daha fazla tam ve uyumlu görünüş gerektirmektedir. Hacim tabanlı çalışmalarda karmaşıklık oranı yüksek eğri yüzeyli ve karmaşık parçaların modellenmesi güçtür. Yaklaşım, YKG temel elemanları ile oluşturulabilen nesne türlerinin katı modellerinin oluşturulmasında çok başarılıdır.

Bu çalışmalar klasik YKG yaklaşımını kullananlar, süpürme operasyonlarını kullananlar ve hacimsel kesiştirme yaklaşımını kullananlar olmak üzere üç alt başlık altında incelenmiştir. 2B izdüşüm yorumlarından elde ettiği YKG ilkelerini kullanarak katı model oluşturan çalışmalar klasik YKG yaklaşımları olarak isimlendirilmiştir. Bu çalışmalar hacim tabanlı çalışmaların ilk dönem uygulamalarından oluşmaktadır [48-54]. Düzlemsel veya silindirik kapalı bölgelere doğrusal veya eksenel süpürme operasyonları tatbik ederek katı model oluşturan çalışmalar ise süpürme operasyonları ile katı model oluşturan çalışmalar olarak isimlendirilmiştir [55-64]. Hacimsel kesiştirme yaklaşımları ise, süpürme operasyonları ile oluşturulan temel katı elemanlara kendi aralarında hacimsel kesiştirme operasyonları uygulamak suretiyle katı model oluşturmaktadır. Son yıllarda bu iki yöntemi kullanan çalışma sayılarında büyük bir artış vardır [9, 65-73].

Süpürme operasyonları ile YKG model elde etme YKG temel eleman türlerinin sınırlılığından meydana gelen problemlerin aşılmasında büyük kolaylıklar sağlamıştır. Son katı oluşturmada hacimsel kesiştirme kullanılması ile de boolean operasyonlarından doğan (birleştirme ve çıkarma) anlam bozukluklarına çözüm getirilmiştir.

Son yıllarda kullanılmaya başlayan karma yöntemlerle tel kafes ve hacim tabanlı yaklaşımların üstün yanları bir araya getirilmeye çalışılmaktadır [74-76]. Yöntem etkili sonuçlar vermekle birlikte, aşırı kullanıcı katkısı içermesi yönüyle önemli bir dezavantaja sahiptir.

3. YAPAY ZEKA ve YAPAY ZEKA TEKNİKLERİ

3.1. Yapay Zeka

Derleme, değerlendirme, saklama vb. gibi beyin gücü gerektiren faaliyetlerde bilgisayarları daha etkin ve daha faydalı kullanmak için değişik algoritmalar geliştirilmiştir. Yapay zeka (YZ) ve alt gruplarına dahil edilebilecek çalışma alanları bu algoritmalar kapsamında değerlendirilmektedir.

YZ kavramını 1956 yılında ilk olarak ortaya atan C. Shannon, M. Minsky ve J. McCarty YZ kavramını aşağıdaki şekilde tanımlamıştır [77]:

“YZ, insanların birbirlerinde zekice kabul ettikleri davranışlara sahip bilgisayarların yapılması ile ilgilenen bilgisayar bilimleri alt alanıdır.”

Yapay zeka, zeka kavramından yola çıkarak geliştirilen algoritmaların, bilgisayar donanım ve yazılımlarına uygulanarak düşünce üretim sistemleri oluşturma çalışmalarının bir bütünüdür.

YZ uygulamaları insan davranışlarını belirli noktalarda taklit edebilen çalışmaları içerir. Bu çalışmalara örnek olarak seri üretim sistemlerinde kullanılan robotlar verilebilir. YZ bunların yanında, insan zekası ve konuşma, anlama ve görme gibi fonksiyonların da bilgisayarlar tarafından yerine getirilmesi için programlar geliştirmeyi de içerir. Burada amaç akıllı makinalar yaparken zekanın doğasını ortaya çıkarmaktır [78].

Bir programın insan zekasını taklit edebilmesi ve bunu başarabilmesi için, çözülecek olan problem alanı konusunda yüksek düzeyde bilgiye sahip olması gerekir. Böylece, çok dar bir problem alanında insan gibi uzman olan bilgisayar programları geliştirilebilir. YZ ile ilgili bilgisayar programları, öncelikle karmaşıklığı, belirsizliği ve bulanıklığı içeren, algoritmik çözümlerin üretilmediği araştırmaların gerekli

olduğu alanlar için uygun olmaktadır. YZ'da problem çözme metodu bilimsel ve mühendislik hesaplamalarındaki çözüm yollarından oldukça farklı olmaktadır [79].

3.2. Yapay Zeka Teknikleri

YZ, insan zekasını bilgisayar ortamında taklit edecek teknikler geliştirmekle ilgilenir. Bu tekniklerden en çok kullanılanları Uzman Sistemler (Expert Systems), Yapay Sinir Ağları (Artificial Neural Network), Genetik Algoritmalar (Genetic Algorithms), Bulanık Mantık (Fuzzy Logic) ve Kaos Teorisi'dir. Bunların yanında pek çok YZ tekniği uygulamalar kullanılmaktadır. Bunlar aşağıdaki şekilde sayılabilir [77]:

1. Endüktif öğrenme,
2. Açıklama tabanlı öğrenme,
3. Benzerliğe dayana öğrenme,
4. Kalitatif muhakeme (çıkarım) veya sağduyu bilgi işleme,
5. Veri tabanlı muhakeme,
6. Model tabanlı muhakeme,
8. Paralel Yapay Zeka sistemleri,
9. Nesne tabanı zeki sistemler,
10. Zeki veri tabanları,
11. Kavramsal grafikler,
12. Mantık programlama,

3.2.1. Uzman sistemler (US)

YZ çalışmalarının temel amacı; insan gibi düşünüp yorum yapabilen, çıkarımlarda bulunup karar verebilen bilgisayar programları oluşturabilmektir. US, yapay zeka çalışmalarının bu amaçlarını gerçekleştirebilmek için oluşturulmuştur.

Bazen bilgi tabanlı bir karar destek sistemi olarak ifade edilen US daha iyi karar

vermeye yardım eden bir araçtır. Bir US, çok dar bir alanda uzman insanın uzmanlığını kullanma davranışına özenen bir bilgisayar programıdır. Bu program bir uzmanın çok özel bir görevde kullandığı bilgi ve bulguları içine alır. YZ ve US uygulamaları alanında ilk çalışmaları yapan Prof. Edward Freigenbaum US'i aşağıdaki şekilde tanımlamaktadır [80].

"Çözümleri için çok iyi yetişmiş uzman kişilere ihtiyaç duyulan zor problemlerde çıkarım işlemlerini ve bilgiyi kullanan zeki bir bilgisayar programıdır."

Bir US, özel uzmanlık gereken bir konuda karar veren veya tavsiyede bulunan bir bilgisayar programı olarak ifade edilmektedir [81].

Yani US karşılaştığı problemlere insan gibi çözüm üretebilen bilgisayar sistemleri olarak tanımlanabilir [82–84]. Bu programlar, geçmişten gelen göz önüne alınabilir kural, karar ve deneyimlerden yararlanarak uygun şekilde çözümler verir [85].

Son yıllarda makine ve imalat sektöründe US yaygın olarak kullanılmaktadır. Bir makine elemanının tasarım aşamasından, pazarlama aşamasına kadar geçen süreçte US'den yoğun olarak faydalanılmaktadır. Özellikle tecrübe ve bilgiye dayalı kararların alınmasında US'ler kullanılmaktadır. Makine imalatı sürecindeki işlemlerin otomatik ve etkili kılınması için, uzman sistem teknikleri kullanımı, çok fazla bilgi ve muhakeme kararlarını bilgi tabanında saklamaya müsaade ettiklerinden dolayı daha avantajlı görülmektedir [86].

Genel YZ programları ile US programları amaç yönünden farklılaşır. US, uzman bir insanın çözebileceği problemleri çözer. Genel YZ programları ise, herhangi bir insanın çözebileceği problemleri çözmeye çalışır. US'lerin bilgi işlemeye müsait yapıları geleneksel dillerden onları ayırır. Geleneksel diller veri işleme esası üzerine kurgulanmışlardır. Ayrıca, sayılar yerine semboller ve listeler işlenmekte sayısal işlemde öte sembolik işlem kullanılabilir. Veri işlemede, veri tabanı bir algoritmaya bağlı olarak etkin bir şekilde işlenirken bilgi işlemede herhangi bir

algoritmaya bağılı kalınmadan örneğin tecrübeye dayalı (heuristic) metodla çıkarılmış kurallar ve gerçeklerden oluşan bilgi tabanı etkin bir şekilde işlenir [87].

US'lerin amacı, uzmanın yerine geçmek değil, onun bilgisini daha yaygın kullanıma sunmaktır. Özetle US'ler, uzman insanın olmadığı yerde diğer insanların verimliliklerini ve kararlarının kalitesini artırarak problemleri daha iyi çözmeyi amaçlarlar. İnsan uzmanlar genellikle çok dar problem çözme alanlarında veya görevlerinde uzmanlaşma eğilimindedirler.

Temel kaynak olan bilgiye sadece birkaç uzman sahip olabilir. Bu yüzden bilginin elde edilmesi önemlidir. Bilgi ancak bu yolla başkalarının da kullanımına sunulabilir. Uzman kişinin hastalık durumunda ondan yararlanılamaz. Kitaplara gelince; bunlar bilginin tamamına sahip olmakla birlikte, bu bilgilerin uygulanması ancak okunmalarıyla mümkündür. US'Ier ise uzmanlığa başvurma şansını doğrudan sağlarlar. Bunun için uzman bilgisini bilgisayarda saklar ve ilgililerin kullanımına sunar.

Bir US'in aşağıdaki özelliklere sahip olması gerekir [80]:

Yüksek verim: Ele alınan probleme yetişmiş uzmana eşdeğer veya daha iyi çözümler getirme özelliğine sahip olmalıdır.

Yeterli cevaplama zamanı: Problemi cevaplama süresi, uzman kişinin kararını oluşturma süresinden daha iyi olmalıdır.

Güvenilirlik: Güvenilir olmalıdır.

Anlaşılabilirlik: Çözüm aşamasındaki her adım sebeplerle açıklanabilmelidir.

Esneklik: Yeni bilgi ekleme, çıkarma ve değiştirme gibi mekanizmalara sahip olmalıdır. Kurallar etkili ve modüler depolama kapasitesine sahip olmalıdır.

US aşağıdaki aşamalar takip edilerek oluşturulur:

- Tanımlama,

- Kavramsallaştırma,
- Formüle etme (Yazılım),
- Test etme ve
- Değerlendirme.

Bilginin sunulma teknikleri

Uzman sistemde uzman bilgisinin bilgisayarın kullanabileceği bir formatta tanımlanması gerekir. Bu amaçla bir dizi bilgi temsil teknikleri geliştirilmiştir. Bunlar aşağıdaki şekilde sayılabilir:

- imalat kuralları
- Çerçeveler
- Semantik ağlar

İmalat kuralları: Bilginin sunulmasında en çok kullanılan yöntemdir. Eylem ve eylemin yerine getirilmesi sonucunda oluşan durum modelinin uygulanması koşulunu içeren bilgi modelidir [77]. Bu yöntemde bilgi “*eğer – o halde*” (*If... then*) yapısında temsil edilir. Bu yapıda kurallar şart cümleleri şeklindedir. Bir şart cümlesi aşağıdaki yapıda yapılandırılır.

EĞER

·
·
·

} Şartlar

ise

SONUÇ

Eğer – O Halde yapısı modüler yapısı, kural eklemeye uygun olması, şeffaf olması ve değiştirilebilir olması yönleriyle tercih edilmektedir [77].

Çerçeveler: Çerçeveler nesneleri tanımlamak için kullanılan hiyerarşik veri yapılarıdır. Bu yapılarda bilgi tabanında nesne özellikleri depolanır. Nesneler, özellikler, sınırlandırmalar, yöntemler ve diğer çerçevelerle olan ilişkileri çerçevesinde tanımlanır.

Çerçeveler, standartlaştırılmış bir durumu gösteren veri yapılarıdır. Bir çerçeveye bağlı olan birçok bilgi çeşidi vardır. Bu bilgiler, çerçevenin nasıl kullanılacağını, kullanıcının bir sonraki aşamada ne olacağını tahmin etmesini ve bu tahminler gerçekleşmediğinde neler yapılacağını açıklar.

Çerçeve yapıları, oluk (slot) ve göz (face)'lerden oluşur. Oluklar nesnenin standart özellik veya niteliklerini temsil eder. Her özellik bir olukla temsil edilir. Oluklar ayrıca çıkarım kurallarını da içerir.

Çerçeve sistemler diğer bilgi sunma tekniklerinden daha karmaşık bir yapıdadır. Bu durum çıkarım mekanizmasının hızını düşürürken bilgi tabanında yapılacak değişiklikleri zorlaştırır [77].

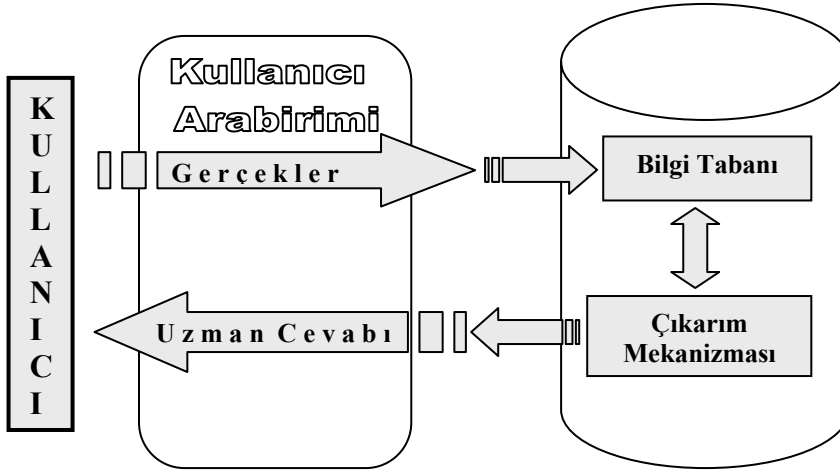
Semantik ağlar: Bilginin semantik kurallar halinde sunulması diğer yöntemlerden farklıdır. Burada daha genel ve grafiksel bir sunum söz konusudur. Semantik kurallar, bilgi tabanında tanımlanan nesneler topluluğunu ve onlar arasındaki ilişkileri anlamlı bir bütün halinde gösterir [78]. Kurallar, bir nesne grubuna ait olmayı tanımlar.

Semantik ağların esneklikleri, varolan düğümler arasında yeni ilişkiler tanımlanabilmesi en önemli avantajlarıdır. Semantik ağın en kötü tarafı ise, ilişkilerin ve düğümlerin tanımlanması için standart bir yapının bulunmamasıdır.

Uzman sistemin yapısı

Genel olarak tüm US'lerde bulunması gereken üç ana modül vardır [77]:

1. Bilgi Tabanı (Knowledge Base)
2. Çıkarım Mekanizması (Inference)
3. Kullanıcı Arabirimi (User Interface) (Şekil 3.1).



Şekil 3.1. Uzman sistemin genel yapısı

Bilgi tabanı (Knowledge base)

Bilgi tabanı uzman sistem için özel bir anlam taşıyan temsili bilgilerden oluşur. Bu bilgiler, uzmanlardan, yayınlanmış veya yayınlanmamış kaynaklardan ve kitaplardan elde edilir. Bilgi tabanında saklanan bilgiler kullanılması gerektiğinde kolaylıkla erişilebilir, sistem yapısı aynı kalmak şartı ile konu kapsamı genişletilmek istendiğinde kolayca genişletilebilir ve ilave yapılabilir olmalıdır. Uzman sistemin bir kişi gibi hareket edebilme yeteneği bilginin iyi temsili ile doğru orantılı olarak artacaktır [88].

Sonuç çıkarım mekanizması (Inference)

Bilgi tabanındaki bilginin nasıl uygulanacağını kontrol eden bileşene sonuç çıkarım mekanizması denir [89]. Sonuç çıkarım mekanizması problem çözme teknikleri yardımı ile bilgi tabanındaki kuralları, gerçekleri ve diğer tüm bilgileri kullanarak sonuca varmaya çalışan mekanizmadır. Bu işlemler yapılırken arama teknikleri ve

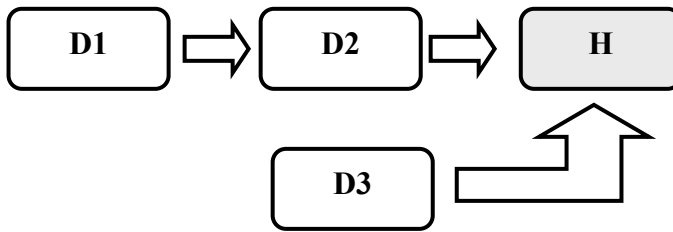
kontrol stratejileri kullanılır. Arama teknikleri *derinlik - ilk*, *genişlik - ilk*, *en iyi - ilk* olarak üç bölümde; kontrol stratejileri ise *geriye zincirleme* ve *ileriye zincirleme* olarak iki bölümde kullanılır [88, 90, 91].

İleriye Zincirleme: Eldeki bilgi ile başlar ve kendisine uyan bir kural bulunca bu kuralın mevcut şartlarını tatmin etmeye çalışır [92]. Şartlar tatmin edilince problemin çözümüne ulaşılır [93]. Şekil 3.2'deki örnekle açıklanacak olursa, ileriye zincirleme yönteminde eğer D1 delili ispatlanırsa D2'ye geçilir ve bu da kanıtlanırsa D3'e geçilir [77]. Şayet D3 delili de ispatlanırsa H hipotezinin doğruluğu ispatlanmış olur.



Şekil 3.2. İleriye zincirleme [77]

Geriye Zincirleme: Bu tür zincirleme tümdengelim ilkesini temel alır ve sonuç kısmında yer alan hipotezin doğruluğunu ispatlayacak deliller aranır. Şekil 3.3'te görülen H hipotezini doğruluğunu ispatlamak için D2 ve D3 delillerini (kurallarını) sorgulamak gerekir. Fakat D2 de D1 'in sorgulanmasını istemektedir. D3'ü sorgulamakla ise D1 ve D2'ye ulaşmadan H hipotezi kanıtlanabilir [77].



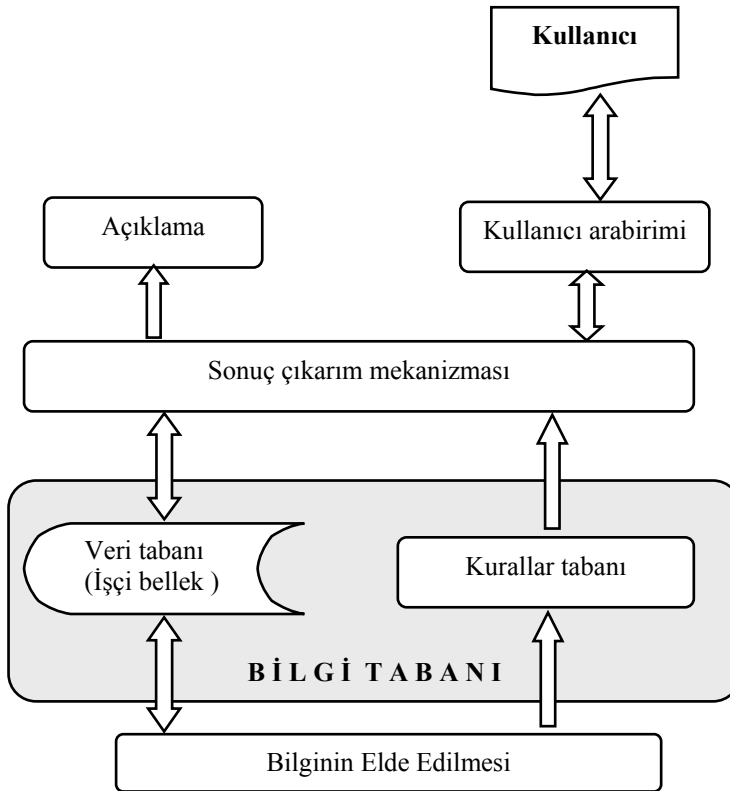
Şekil 3.3. Geriye zincirleme [77]

Kullanıcı arabirimi: Sistemi kullanan ile sistem arasındaki köprüdür. Genellikle neden (why) ve nasıl (how) sorularına cevap veren bir açıklama ünitesini içerir [79]. Bunun yanında, doğal dil ve grafik etkileşimi ile sistemle kullanıcı arasındaki etkileşim sağlanmaktadır. Kullanıcı arabirimine programlarda kullanılan menüler

örnek verilebilir.

Bilgi tabanı çok iyi hazırlanmış, çıkarım mekanizması etkin olan bir uzman sistemde kullanıcının uzman sistem ile etkileşimi yetersiz ise US'in performansından iyi yönde bahsetmek imkansızdır [88].

Şekil 3.4'te Uzman Sistemin detaylı yapısı görülmektedir. Yukarıda bahsedilen üç ana modülün dışında sistemin yapacağı işi desteklemek ve kullanıcı ile olan ilişkiyi sağlamak için bilginin elde edilmesi, tabii dilde etkileşim araçları ve açıklama alt sistemleri de böyle bir sistemin yapısına dahil olmaktadır [77].



Şekil 3.4. US'in ayrıntılı yapısı [77]

Uzman sistemlerin avantajları

Zaman ve emek isteyen bir iş olmasına rağmen US geliştirmenin ve kullanmanın

çeşitli avantajları vardır. Bunlardan bazıları şu şekilde sıralanabilir [77, 94, 95]:

1. US uzman kişilerden daha hızlı olduğu için az zamanda daha fazla üretim yapabilir, bu da verimliliği artırır.
2. Uzman kişilerden farklı olarak sürekli aynı kapasite ile kullanılabilir.
3. Uzman kişinin yüksek maliyetine karşın US'ler uzun vadede işletmelere daha ucuza mal olmaktadır.
4. Hata oranları düşüktür.
5. Tutarlı karar verebilme yetenekleri kaliteyi yükseltmektedir.
6. Karmaşık donanımları daha kolay çalışabilir hale getirir.
7. Tehlikeli ortamlarda çalışılmasını gerektiren durumlarda kişinin bu tehlikenin dışında kalmasını sağlar.
8. Eğitim amaçlı kullanılabilir. Bu özelliği ile yeterli bilgi sahibi olmayan kullanıcılara açıklama özelliği sayesinde bilgi verebilir.
9. Kişinin bilgisini ve kapasitesini aşan problemlere çözümler üretebilir.
10. İhmal etmeden bütün detayları inceledikleri için güvenilir sistemlerdir.
11. US'ler problem alanında çözüme ulaşmak için verimli bir şekilde araştırma yeteneklerine sahiptirler. Bu özellikleri ile problemlere sağlıklı öneriler üretebilirler.
12. Tam ve kesin olmayan bilgi ile çalışabilirler.
13. Fazla verinin incelenmesi gerektiği durumlarda cevap verme süresi kısadır.

Uzman sistemlerin dezavantajları

US'in yukarıda sayılan avantajlarına rağmen bir takım dezavantajlara sahiptir. Bunlar şu şekildedir [77]:

1. Her uzmanın aynı olaya farklı yaklaşma ihtimali oldukça yüksektir. Bu da birden fazla uzman ile çalışılması durumunda çelişkili durumlar ortaya çıkabilir.
2. US'lerden verim alınması henüz yalnızca dar alanlarda çalışılması ile mümkündür.
3. Yönetici ve patronların yeni teknolojiye şüphe ile bakmaları ve fazla para

harcamak istememeleri US yazılımların bu sınıf tarafından zor kabullenilmesine neden olmaktadır.

4. US'i geliştirmede önemli elemanlardan biri olan bilgi mühendisi maliyeti yükselten bir unsurdur.
5. Sistem geliştirme zamanının uzun ve maliyetinin yüksek olması uygulamaları kısıtlayıcı bir etken olabilir.
6. Uzman bir kişi özellikle alışlagelmemiş durumlarda bir US'den daha yaratıcı olmaktadır.

Uzman sistemin uygulama alanları

Son yıllarda yapılan çalışmalarla anlaşılmıştır ki US üretimi arttırmakta, kaliteyi yükseltmekte ve maliyeti düşürmektedir [96]. Makine ve imalat sektöründe US'ler özellikle tecrübeye ve bilgiye dayalı kararların alınmasında çok fazla kullanım alanına sahiptir [97, 98]. Üretim ve tasarım ile ilgili işlemlerde muhakeme sürecini otomatikleştirmek ve işlem hızını arttırmak için US teknikleri kullanılmaktadır. Yapılan işlemlerin otomatikleştirilmesi ve etkisinin artırılması için US kullanımı, çok fazla bilgiyi ve muhakeme kurallarını bilgi tabanında saklamaya izin verdiğinden dolayı daha avantajlıdır [86].

US mühendislik, iş ve süreç planlama, tıp, askerlik ve ekonomik analizlere kadar pek çok alanda kullanılmaktadır. Uygulama alanları fonksiyonlarına göre ayrıntılı olarak aşağıdaki şekilde belirtilebilir [77, 99 - 101]:

- Teşhis
- Sınıflandırma
- Seçim
- Veri analiz ve yorumu
- Tasarım
- Tahmin
- Simülasyon
- İzleme

- Danışmanlık
- Zeki yardım-destek
- Planlama ve programlama
- Görüntüleme
- Hata ayıklama
- Kontrol ve eğitim.

3.2.2. Genetik algoritmalar (GA)

Günümüzün karmaşık ve zor koşulları problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle sert optimizasyon teknikleri yerine, yumuşak hesaplama (softcomputing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel yaklaşımlardan olan genetik algoritmalar da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır. Uygulama başarıları artan ve sürekli geliştirilmeye çalışılan genetik algoritmalar diğer yumuşak hesaplama yöntemleri ile birlikte kullanılarak karma çözümler geliştirilmesine çalışılmaktadır [102].

Doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemi olan genetik algoritmalar hakkında ilk çalışmalar 1960'larda John Holland ve ekibi tarafından başlatılmıştır. Holland ve ekibinin tasarladığı algoritmalar, genlerin ve kromozomların davranışlarıyla benzer özellikler taşımaktadır. Mutasyona uğrama, çiftleşme, çoğalma, yetersiz olanların elenmesi ve başarılı bireylerin birleşip yeni bir nesil oluşturması algoritmaların en önemli özellikleridir. Bu özelliklerinden ötürü bu algoritmalar Holland tarafından "Genetik Algoritmalar" olarak adlandırılmışlardır [103].

Genetik algoritmaların işlem adımları

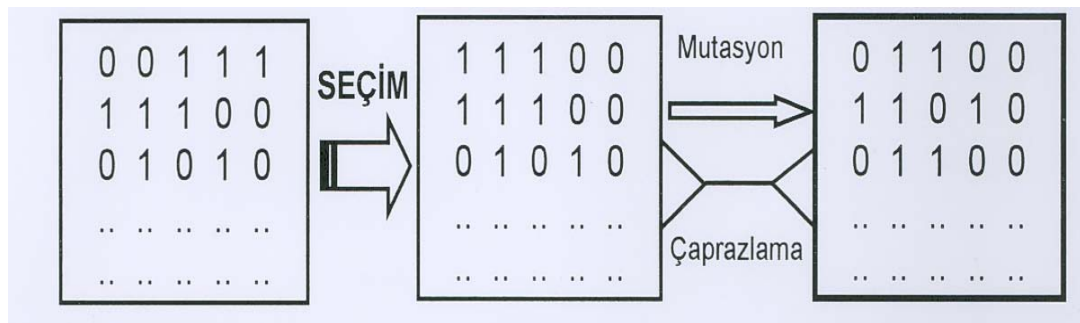
GA rassal arama teknikleri kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayalı bir arama tekniğidir [104]. GA, pek çok problem türü için uygun parametreler ile çalışıldığı taktirde makul çözümler verir. GA'nın çalışma yöntemi

Darwin'in doğal seçim "en iyi olan yaşar" prensibine dayanmaktadır [105, 106]. GA doğadaki canlıların geçirdiği evrim sürecini dikkate alır. Amaç doğal sistemlerin uyum sağlama özelliğini dikkate alarak yapay sistemleri tasarlamaktır [107].

GA'larda ilk adım problemin arama uzayını temsil edecek kodlama yapısının seçimidir. ikili düzende kodlama yöntemi (0 ve 1) en çok kullanılan yöntemdir. Fakat, çok değişkenli bir fonksiyon optimizasyonunda alt ve üst sınırlara bağlı olarak dizi uzunluğu çok büyük olacağından gerçel sayı kodlama ve permutasyon kodlama türleri de kullanılmaktadır [101].

Arama uzayını temsil eden kodlara *gen* denilir. Genlerin oluşturduğu her bir kromozom problem için alternatif bir çözüm getirir. Fakat burada önemli olan en iyi çözümü bulabilmektir. Her kromozomun problem çözümüne uygunluğunu gösteren bir uygunluk fonksiyonuna ihtiyaç vardır. Bir GA belirli sayılardaki kromozomlar üzerinde işlevini görür, belirli bir seçim kriterine göre seçilen iki kromozomu birleştirerek (çaprazlama) yeni kromozomlar oluşturur (Şekil 3.5).

Bazı durumlarda çözüme daha önce varabilmek için kromozom üzerinde bilgi değişikliği yapmak gerekir. Buna *mutasyon* denir. Eski kromozomlar ile üretilen yeni kromozomlar uygunluk fonksiyonu kullanılarak bir değerlendirme operasyonundan geçirilir. Daha sonraki adımlarda yeni kromozomları oluşturacak kromozomlar seti belirlenir (yeniden üretim) [108].



Şekil 3.5. Genetik Algoritma işlemleri [109]

Çaprazlama, mutasyon, değerlendirme ve yeniden oluşturma işlemleri daha iyi sonuçlar üretilinceye kadar devam eder. Sonuçta uygunluk fonksiyonu en yüksek olan kromozom çözüm kromozomu olarak belirlenir. Bu kromozomun gösterdiği çözüm, problemin de çözümü olarak kabul edilir [110].

GA, çözümlerin kodlanması, uygunlukların hesaplanması, çoğalma, çaprazlama ve mutasyon operatörlerinin uygulanmasını içerir [111]. GA'nın en büyük özelliği, problemin değişen koşullarına algoritmanın kendisini adapte edebilmesidir [112].

Yukarıda özetlenen algoritma işlem adımları altı ana aşamada özetlenebilir. Bunlar [113]:

1. Başlangıç yığınının oluşturulması,
2. Bireylerin uygunluk değerinin hesaplanması,
3. Uygunluk değeri yüksek bireylerin seçilmesi (ebeveynler),
4. Ebeveynlerden yeni nesillerin türetilmesi (çaprazlama, mutasyon),
5. Yeni nesillerin değerlendirilmesi,
6. Amaca ulaşana kadar 3 - 5. adımların tekrarlanmasıdır.

Genetik algoritmaların özellikleri

Genetik algoritmaların diğer metodlardan üstünlükleri ve bazı farklılıkları aşağıdaki şekilde sayılabilir;

- GA, parametrelerin kendisi ile değil, parametre dizilerinin kodlanması ile çalışırlar. Kodlama biçimi genelde ikilik sayı sistemindedir (0 ve 1). Bu sistem bilgisayarın çalışma sistemi ile aynıdır [114].
- GA birçok cephede aynı anda optimum çözümü aramaktadır ve dolayısıyla yerel bir optimumda sıkışma riski minimumdur [1115]. Yerel değil global arama yaparak sonuca ulaşmaya çalışır. Bir tek yerden değil bir grup çözüm içinden

arama yapar.

- Olasılık kurallarına göre çalışan GA, yalnızca amaç fonksiyonuna ihtiyaç duyar. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar [114].
- Diğer bir önemli üstünlükleri ise çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır [102].
- Diğer metodlarda olduğu gibi doğrudan parametreler üzerinde çalışmaz. GA, optimize edilecek parametreleri kodlar ve parametreler üzerinde değil, bu kodlar üzerinde işlem yapar.

Mühendislik tasarımında karşılaşılan pek çok problemde tasarım değişkenleri ayrıktır. Yani, bir problemin çözüm kümesi birbiri ile alakası olmayan değerlerden meydana geliyor olabilir [110]. Tasarım değişkenlerinin ayrıık olması durumunu ele alan çok az algoritma geliştirilmiştir. Ayrıık değişkenlerin kullanıldığı GA, zaman alıcı karmaşık çözüm uzayları ve bilinen doğrusal programlama metotlarıyla çözülemeyen optimizasyon problemlerinin çözümü için esnek ve etkili sonuçlar vermektedir [104].

Genetik algoritmaların uygulama alanları

GA'nın, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır. Karmaşık problemleri hızlı ve en uygun sonuca yakın olarak çözebilen genetik algoritmalar, çeşitli problem tiplerine uygulanabilmektedir. GA'lar deneysel ve pratik uygulamalar ile sınıflandırıcı sistemlerde yaygın olarak uygulanmaktadır [116].

Pratik uygulamalar endüstri problemlerinin çözümünde kullanılmaktadır. Sayısal optimizasyon yöntemleri, görüntü işleme ve çizelge problemleri bunlara örnek olarak verilebilir.

Bilgi çıkarma amacı ile kullanılan GA'lar ise sınıflandırıcı sistemlerde

kullanılmaktadır. Deneysel uygulamalar ise GA'ların diğer optimizasyon yöntemlerine karşı üstünlüklerini ispatlamak amacını gütmektedir [116]. Çözüm uzayının büyük olduğu yerlerde genetik algoritmalar kısa sürede, etkili ve kesin çözümler bulabilmektedir [117].

3.2.3. Yapay sinir ağları (YSA)

Beynin üstün özellikleri, bilim adamlarını üzerinde çalışmaya zorlamış ve beynin nörofiziksel yapısından esinlenerek matematiksel modeli çıkarılmaya çalışılmıştır. Beynin bütün davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Böylece YSA denen yeni ve günümüz bilgisayarlarının algoritmik hesaplama yönteminden farklı bir bilim alanı ortaya çıkmıştır.

YSA, insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır. YSA, genel olarak insan beyninin ya da merkezi sinir sisteminin çalışma prensiplerini taklit eden bilgi işleme sistemleridir. YSA, yapay zeka çalışmaları kapsamında, insan beynindeki birçok işlem elemanının veya basit işlemcilerin çalışma prensiplerine göre tasarlanmıştır. Olaylar arasındaki ilişkileri bilinen örnekleri kullanarak öğrenme, karar verme, sonuç çıkarma davranışlarının yapay olarak modellenmesi esasına dayalı, paralel çalışma özelliği olan bir bilgi işleme sistemidir [108]. Bu yüzden YSA bazen, Paralel Dağıtılmış İşleme Sistemleri (Paralel Distributed Processing Systems) ya da Bağlantıcı Sistemler (Connectionist Systems) olarak da adlandırılırlar.

En genel anlamda bir YSA, bir çok nöronun, ya da yapay olarak basit işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanması ile oluşan karmaşık bir sistem olarak düşünülebilir [118]. YSA'nın en önemli özelliği, öğrenme kabiliyetinin bulunmasıdır. YSA, öğrenmenin yanında ezberleme ve bilgiler arasında ilişkiler kurma yeteneğine de sahiptir.

Yapay sinir ağlarının tarihçesi

YSA çalışmaları 1940'lara kadar dayanmaktadır. Bu alandaki ilk çalışma Mc. Culloch ve Pitts'in elektrik devreleriyle modelledikleri basit bir sinir ağıdır. Daha sonra 1949'da Donald Hebb beynin öğrenme mekanizması üzerine ortaya koyduğu biyolojik öğrenme kuralını geliştirdi. Hebb kuralı olarak adlandırılan bu kural, sinir ağının bağlantı sayısı değiştirilebilirse, öğrenebileceğini öngörmekteydi [106]. 1958'de Rosenblatt Hebb'in kuralından yola çıkarak nöronlara dayalı bir hesaplama modeli geliştirmiştir. Rosenblatt Perceptron'u geliştirerek bu alandaki çalışmaların hızlanmasına yol açmıştır. Perceptron, tek katmalı, eğitilebilen ve tek çıkışa sahip olan yapay sinir ağıdır [106].

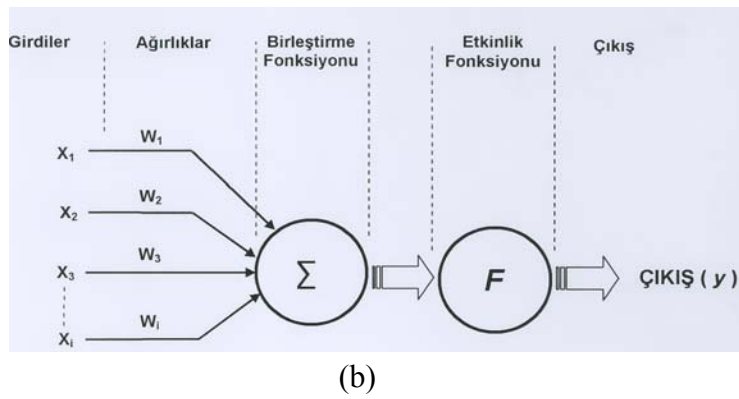
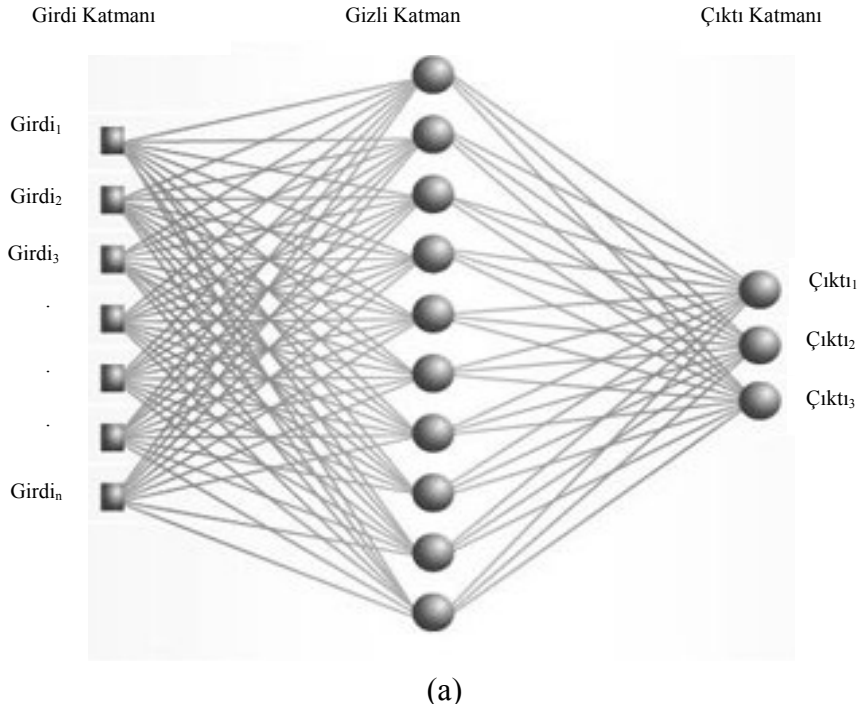
1982 yılında J.J. Hopfield tarafından yayınlanan "Neural Networks and Physical Systems" adlı çalışma ile YSA'da yeni bir devir başlamıştır [118]. Hopfield bu çalışması ile açık ve matematiksel analizleri kullanarak, beyne benzeyen kullanışlı bir aletin nasıl çalışabileceğini ve ne yapabileceğini göstermiştir [106].

1986'da Rumelhart ve arkadaşları ileri beslemeli yeni öğrenme modeli olan hatanın geriye yayılma algoritmasını geliştirmiştir [119]. Bu gün YSA uygulamalarında bu öğrenme yöntemi, öğrenme yöntemi kolay olması sebebi ile yaygın olarak kullanılmaktadır.

Yapay sinir ağlarının yapısı

Yapay sinir hücreleri, YSA'nın çalışmasına esas teşkil eden en küçük bilgi işleme birimidir. Geliştirilen hücre modellerinde bazı farklılıklar olmakla birlikte genel özellikleri ile bir yapay sinir modeli, girdiler, ağırlıklar, birleştirme fonksiyonu (toplama), aktivasyon (etkinleştirme) fonksiyonu ve çıktılar olmak üzere 5 bileşenden meydana gelir (Şekil 3.6).

Girdiler (X_1, X_2, \dots, X_i), diğer hücrelerden ya da dış ortamlardan hücreye giren bilgilerdir (Şekil 3.6). Bilgi, bir önceki sinirlerden veya dış dünyadan gelir. Bir sinir genellikle gelişigüzel bir çok girdileri alır [106].



Şekil 3.6. Yapay bir sinirin:
a) katmanları
b) temel bileşenleri

Bilgiler, bağlantılar üzerindeki *ağırlıklar* (w_1, w_2, \dots, w_i) üzerinden hücreye girer ve ağırlıklar, ilgili girişin hücre üzerindeki etkisini belirler. Öğrenme esnasında sürekli değişerek girdi ve çıktı arasındaki ilişkiyi yakalamaya (optimize etmeye)

çalışır. Her bağlantının bir ağırlığı vardır. Ağırlık büyüdükçe etki de büyür. Ağırlığın sıfır olması, hiçbir etkinin olmaması; negatif olması ise etkinin ters yönde olması demektir [108].

Birleştirme fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur. Net girdi genellikle girişlerin ilgili ağırlıkla çarpımlarının toplamıdır. Toplama yanında, çarpım, maksimum, minimum, çoğunluk ve kümülatif toplam fonksiyonları da birleştirme fonksiyonu olarak kullanılmaktadır [120].

Aktivasyon (etkinlik) fonksiyonu, birleştirme fonksiyonundan elde edilen net girdiyi bir işlemden geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur. Bir etkinlik fonksiyonunun kullanım amacı, zaman söz konusu olduğunda toplama fonksiyonunun çıkışının değişmesine izin vermektir [106].

Çıkış Fonksiyonu, etkinlik fonksiyonu sonuçlarının gönderildiği yerdir. Çıkış fonksiyonu, etkinlik fonksiyonundan aldığı sonuçları ya bir sonraki işlem elemanına (sinirlere) veya ağına dışına gönderir. Bir çıkış, kendinden sonra gelen herhangi bir sayıdaki diğer sinirlere giriş olabilir.

Yukarıda bahsedilen beş bileşen arasında yapılan işlemler kısaca şu şekildedir: X_i sembolü ile gösterilen girişlerin her biri ağırlık ile (w_i) çarpılır. Çarpımların toplamı ($\sum x_i \times w_i$) ile elde edilen sonuç birleştirme (toplama) fonksiyonu sonucudur. Elde edilen bu değer giriş değerleri ile karşılaştırılır. Eğer toplam başlangıç değerinden büyükse doğrusal olmayan (F) etkinlik fonksiyonunu kullanarak bir çıkış (y_j) hesaplar. y_j çıkış değeri bu toplam ile başlangıç değeri arasındaki farkın doğrusal olmayan (F) fonksiyonudur [117]. Aktivasyon (Etkinlik) fonksiyonu $F(s) = 1 / (1 + e^{-s})$ olmak üzere çıkış fonksiyonu $y = F(\sum x_i * w_i - t)$ şeklinde gösterilir. Burada t başlangıç değeridir. Doğrusal olmayan F etkinlik fonksiyonu, bir modelleme seçimi ve yapay sinir ağı modelinde istenen çıkış işareti cinsinden bir fonksiyondur. Bu fonksiyon için en fazla kullanılan seçimler ise sigmoid, basamak ve rampa fonksiyonudur [121]. Tüm yapay sinir ağları bu temel yapıdan türetilmiştir.

Yapay sinir ağlarının üstünlük ve sakıncaları

YSA'nın en önemli üstünlükleri, öğrenme kabiliyeti olması ve farklı öğrenme algoritmaları kullanabilmesidir. Bunun yanı sıra en sık belirtilen sakıncaları ise sistemin çalışmasının analiz edilememesi ve öğrenme işleminde başarılı olmama riski olmasıdır. Bunların yanısıra YSA'ların üstünlükleri aşağıdaki şekilde sıralanabilir [106]:

- Uzman sistemler gibi bilgiyi kurallar halinde istemezler,
- Öğrenebilir ve hiç karşılaşmadıkları bir problemi çözebilirler,
- Paralel yapıları nedeniyle çok hızlı çalışırlar,
- Matematiksel modele ihtiyaç duymazlar,
- Sistemin herhangi bir işlem noktasında ortaya çıkabilecek bir hatanın sistemi çökertmemesi için hataya izin verebilecek bir tarzda tasarlanabilir.

YSA'ların bazı sakıncaları ise aşağıda sıralanmıştır:

- Çıkardıkları sonuçları nasıl ve neden çıkardığını açıklayamaz (kapalı kutu),
- Eğitimleri oldukça zaman alıcı ve zordur,
- Bazı ağlar hariç kararlılık analizleri yapılamaz,
- Farklı sistemlere uyarlanması zor olabilir.

YSA'nın uygulama alanları

Son yıllarda YSA' ları, özellikle günümüze kadar çözümü güç ve karmaşık olan ya da ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmış ve genellikle başarılı sonuçlar alınabilmektedir. Geleneksel bilgisayarların sadece algoritmik ve kesin aritmetik işlemlerde hızlı olması YSA'nın kullanım alanını genişletmiştir. YSA, önceleri tıp ve işlem elemanı modelleme çalışmalarında kullanılırken bugün, sayısal optimizasyon tasarımlarını da içine alacak şekilde, klasik yöntemlerle çözülemeyen problemlerin çözümünde de kullanılmaktadır [108].

YSA makine tasarımı çalışmaları, plastik kalıpcılık, robot kontrolü vb. alanlarda modelleme amaçlı olarak kullanılmaktadır. Bunların yanında, sınıflandırma, tahmin, kontrol sistemleri, optimizasyon ve karar verme gibi bazı uygulamaları bulunmaktadır [120].

Genel olarak YSA model seçimi ve sınıflandırılması, işlev tahmini, en uygun değeri bulma ve veri sınıflandırması gibi işlerde başarılıdır. Bugün ise sayısal optimizasyon tasarımlarını da içine alacak şekilde, klasik yöntemlerle çözülemeyen problemlere başarılı çözümler getirmektedir. Böylece matematik ve fizik gibi temel bilimlerle, elektrik, bilgisayar ve makine mühendisliği gibi uygulamalı alanlarda kullanılır hale gelmiştir. Bunların yanında YSA'nın uygulama alanları aşağıdaki şekilde sayılabilir:

- Arıza Analizi ve Tespiti
- Tıp Alanında
- Savunma Sanayi
- Haberleşme
- Üretim
- Otomasyon
- Kalite kontrol
- Görüntü tanıma

3.2.4. Bulanık mantık (BM)

Aristo'da sembolleşen klasik mantığın 1 ve 0 ya da "doğru" ve "yanlış" yargıları uç değerleri temsil etmektedir. Böyle bir durumda karmaşık sistemleri modellemek ve kontrol etmek zordur. Günlük hayatta da kesin sayılar ve ifadeler yerine çoğunlukla ara değerler kullanılmaktadır. Klasik mantığın gerçek dünya problemleri için yeterli olmadığı durumlar sonucunda ortaya çıkan bulanık mantık (BM), 1965 yılında Lutfi A. Zadeh tarafından geliştirilmiştir [122]. Zadeh'in geliştirdiği bu yöntemle, insana özgü tecrübe ve öğrenme olayı kolayca modellenenilmiş ve belirsiz durum ve kavramlar matematiksel olarak ifade edilebilmiştir.

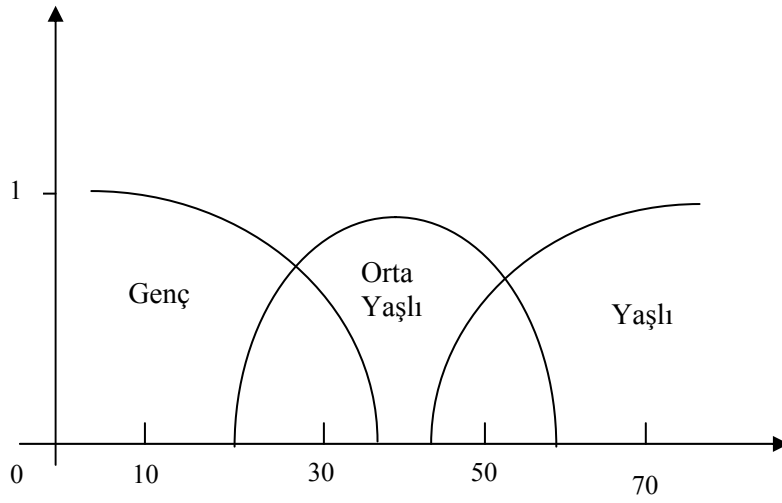
Bulanık mantığın yapısı

Geleneksel küme kavramına göre herhangi bir nesne bir kümeye ya aittir, ya da ait değildir. Bu tür kümeler keskin kümeler olarak adlandırılır. Bulanık kümeler ise günlük yaşamda daha çok kullanılan, sınırları "bulanık" sayılar, ifadeler ve nesne sınıflarından oluşur [119].

Bulanık küme kavramında nesnelerin kümeye aitliği derecelere bölünmektedir; herhangi bir nesnenin bulanık kümeye aitliği 0.5'tir, 0.8'dir veya 0.25'tir gibi deyimlerle burada sıkça karşılaşılmaktadır. Böylece nesnelerin bulanık kümeye aitliği üyelik dereceleri ile belirlenir ve bu dereceler de üyelik fonksiyonları ile karakterize edilir.

Bulanık küme kavramının keskin kümelerden farkı basit bir örnekle gösterilebilir. Eğer 40 yaşı orta yaş olarak ele alınırsa, keskin kümelerde 30 yaşın altındaki kişiler "genç", 30-50 yaş arası "orta yaşlı", 50 yaşın üstündekiler ise "yaşlı" kümelerine dahil edilebilir. Buna göre 29.5 yaşındaki biri "genç" sayılırken 30.5 yaşındaki diğer bir kişi "orta yaşlı" sayılacaktır. Aynı zamanda 35 yaşındaki bir insana "orta yaşlı" denemeyeceği gibi "genç" de denemez. Belki de bu kişiyi hem "genç" hem de "orta yaşlı" olarak düşünmek gerekecektir. Bulanık kümeler bu düşüncüyü olası kılar. Kümelerin keskin çizgilerle ayrılmamış olması, aralarında belirli bir örtüşüm olması, 35 yaş için isteneni düşünebilmeye imkan tanır [119].

Bulanık kümeleri ifade etmek için kullanılan üyelik fonksiyonlarına bir örnek Şekil 3.7'de verilmektedir. Buradaki üyelik fonksiyonu verilen yaş örneğini ele almaktadır.

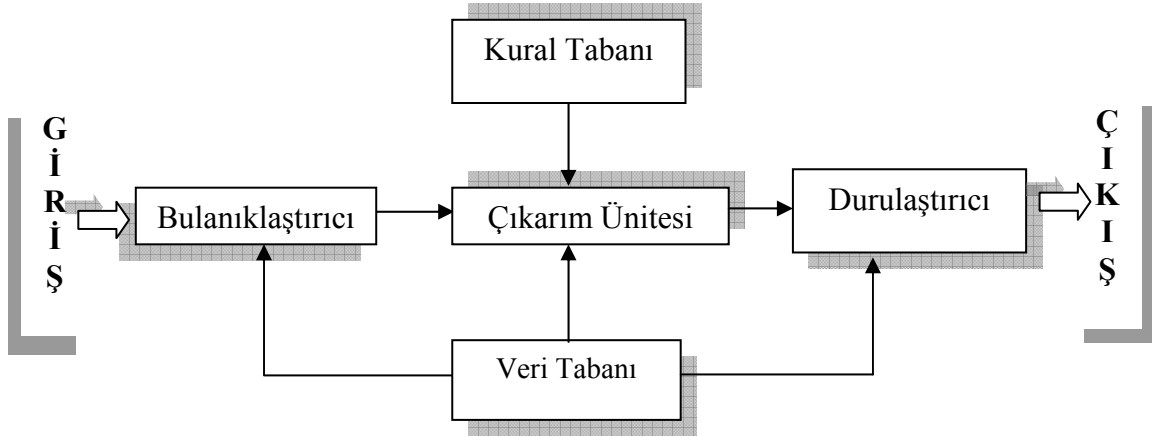


Şekil 3.7. Bulanık kümelerde üyelik fonksiyonu [119]

BM, teorik ve matematik aksiyonlu yaklaşımlardan bağımsız bir çözüm algoritması içerdiğinden, herhangi bir sistemin modellenmesinde yaklaşık ve kolay çözüm verir [101]. Zadeh, BM teorisinde kullanılan teknik ve kavramların olasılık teorisinde de kullanılması gerektiğini belirtmektedir. Çünkü olasılık teorisi gerçek problemlerdeki belirsizlikleri ele almada yetersizdir [123]. Buna gerekçe olarak da olasılık teorisi ve BM birbirinden farklı olması, belirsizlik durumlarında olasılık teorisinin yetersiz kalması ve bu iki teorinin birbirini tamamlaması gösterilmektedir [101].

Bulanık değişkenlere üyelik değerlerinin ya da fonksiyonların atanabilmesi için kullanılan yöntemlere sezgisel, çıkarım, açısız fuzzy kümeler, yapay sinir ağları, genetik algoritmalar, indüktif çıkarım örnek olarak gösterilebilir [121].

Şekil 3.8'de bir bulanık denetleyicinin yapısı görülmektedir. Bulanık denetleyiciler Veri tabanı, bulanıklaştırıcı, kural tabanı, çıkarım ünitesi ve durulaştırıcıdan meydana gelmektedir [124].



Şekil 3.8. Bulanık bir denetleyicinin yapısı [124]

Kural tabanı: Kural tabanında girdi ve çıktı üyelik fonksiyonları arasındaki ilişkiyi belirleyen kurallar bulunur.

Çıkarım ünitesi: Bulanık deneysel veriler, kurallar çerçevesinde bulanık mantık yürütülerek mantıksal hale dönüştürülür.

Durulaştırıcı: Bulanık bilgilerden yararlanabilmek ve kesin sayılar haline dönüştürebilmek için bilgiler durulaştırılır. Bir bulanık küme işleminin sonucundaki bulanık kümenin tek sayı haline dönüştürülmesi işlemine durulaştırma adı verilir.

Bulanık mantığın avantaj ve dezavantajları

Bulanık mantığın avantajları şu şekilde sıralanabilir [123]:

- Bulanık mantık, eksik ve şüpheli bilgiden kaynaklanan belirsizliği hesaba katmanın sistematik bir temelini sunmaktadır.
- Keskin sınırları olmayan kümeler bulanık kümeleri kullanılarak kolaylıkla modellenebilir.
- Bulanık mantık uzman bilgisini kullanmaya imkan sağlar, bu bilgiyi tutarlı ve yapısal bir yolla işleme yeteneğine sahiptir.

- İnsana özgü tecrübe ile öğrenme olayı kolayca modellenenebilir ve belirsiz kavramlar bile matematiksel olarak ifade edilebilir.

Bulanık mantığın en önemli dezavantajı öğrenemeyen ya da öğretilemeyen bir yapıya sahip olmalarıdır.

Uygulama alanları

Bulanık mantığın uygulama alanları aşağıdaki şekilde sayılabilir:

- Oto elektroniği fren sistemleri (Otomotiv)
- Elektronik denetim sistemleri
- Karar verme
- İşlem planlama

3.3. Sonuç

Bu tezde 2B görüşlerde 3B katı temel eleman yorumları oluşturma aşamasında yapay zeka tekniklerinden Uzman Sistem (US) kullanılmıştır. Katı temel elemanların türüne karar verme ve buna bağlı olarak yapılacak boolean operasyonunu belirleme 2B görüşlerin birbirleri ile ilişkileri değerlendirilerek gerçekleştirilir. Bu aşamada, kullanılacak yapay zeka tekniğinin kural tabanlı ve genişletilebilir olması tercih edilmiştir.

Katı temel elemanlar 2B görüşlerin birbirleri ile ilişkilerinin yorumlanması ile oluşturulmaktadır. Bu yapıda problem, kural kümeleri ile tanımlanmaktadır. Kural cümleleri problemi kolay tanımlaması yanında, kolay değiştirilebilir olmaları yönüyle tercih edilmiştir.

Bu aşamada, yapay zeka teknikleri içerisinde kural tabanlı bir yapı kullanmaları sebebiyle birbirlerine çok benzeyen Bulanık Mantık ve Uzman Sistemin sisteme uyarlanabilirliği incelenmiştir. Bu iki teknik, küme kavramları açısından birbirinden ayrılmaktadır.

Geleneksel küme kavramı, herhangi bir nesnenin kümeye ait olma veya olmaması ile ilgilenir. Bu mantığa göre bir şey ya uzundur, ya kısa veya ya soğuktur ya da sıcak. Bulanık kümede ise, bir nesnenin bir kümeye aidiyeti üyelik derecesi ile ifade edilir. Bulanık kümelerde nesnenin kümeye aitliği derecelerle ifade edilir. Bir nesne aynı anda birden fazla kümeye üye olabilir. Birine çok fazla üye iken, diğerine daha az üye olabilir. Bu işlem için, bulanık mantık yaklaşımında eldeki veri bulanıklaştırılır. Burada, “girdilerin bulanıklaştırılması ile tez hedeflerine ulaşılabilir mi ve bulanık kural tabanı ile temel eleman türleri elde edilebilir mi?” sorularına aranan cevap nihai olarak kullanılacak tekniği belirlemiştir.

Yürütülmekte olan tez çalışmasında kullanılan veriler temel izdüşümlerin fiziksel özelliklerini ifade etmektedir. Bunlar 2B temel elemanların (doğru, yay veya daire) köşe koordinatları (x ve y), yay ve dairelerin yarı çapları, yayların açıları, görünürlük durumları gibi verilerdir.

Bir sistemde bulanık mantık uygulayabilmek için o sistemdeki verilerin ara değerler halinde temsil edilmeleri gerekmektedir. Yani, verilerin bulanık ifadelerle, sözel değişkenlerle büyüklüğünün ifade edilebilmesi gerekmektedir. Bizim sistemimizde veriler kesin ifadeler şeklindedir.

Ayrıca, bu çalışmada ele alınan parçalar standart parçalar değildir. Belli bir biçim, büyüklük ve unsur sınırlandırması yoktur. Yani standart, belli bir biçimdeki bir parçanın parametrik tasarımından ziyade, serbest girdinin izdüşümlerin birbirleri ile kesin ilişkileri yardımı ile yorumlanıp katı modelinin oluşturulması söz konusudur.

2B'lu izdüşüm girdilerine bir sınırlama getirilmediği için, köşe koordinatlarının üyelik fonksiyonlarını oluşturmak zordur. Bu noktaların hangi aralıklarda küçük, büyük veya negatif küçük, pozitif büyük aralıklarında yer alacağına karar vermek sağlıklı sonuç üretilmesini engellemektedir.

Veri yapısının keskin ve belirli bir şablona uymaması nedeniyle çalışmada Uzman Sistem kullanılmasına karar verilmiştir. Bunun yanında, yeni problemler karşısında kural tabanının genişletilebilir olması ve sisteme müdahale edebilme kolaylığı US'i tercih edilebilir bir teknik yapmıştır.

4. 3B NESNE OLUŞTURMADA KULLANILAN NESNE TEMSİL TEKNİKLERİ

İlk BDT sistemleri sadece iki boyutlu teknik resim çizimleri oluşturan sistemlerdir. Bu sistemler kullanıcıya düz çizgi, yaylar, daireler ve elipsler sağlamıştır. Yetmişli yılların ortalarından itibaren BDT kavramının gelişmesi ile birlikte 3B'lu BDT sistemleri geliştirilmeye başlanmıştır. Bu sistemler nesnelerin 3B'lu modellerini oluşturmaktadır. 3B modelleme sistemlerini üç ana grupta toplamak mümkündür. Bunlar aşağıdaki şekilde sıralanabilir:

- a) Tel kafes modelleme
- b) Yüzey modelleme
- c) Katı modelleme

İlk 3B'lu BDT sistemleri tel çerçeve modellerden oluşmaktadır. Bunlar, iki buçuk boyutlu nesne oluşturmada bazı avantajlar sağlamıştır. Bu sistemlerin nesne biçimini tanımlama ve gösterimdeki sınırlılıkları yüzey modelleyicileri ortaya çıkarmıştır. Yüzey modelleyiciler otomotiv, uçak ve uzay sanayiinde yaygın kullanım alanına kavuşmuştur.

Katı modelleme, tel çerçeve ve yüzey modelleme sistemlerinin parça yorumlamadaki yetersizlikleri ve karmaşık yapıları sonucu ortaya çıkmıştır. Katı modelleme sistemleri 3B gösterim, analiz ve yorumlama işlemlerinde büyük avantajlara sahiptir ve model oluşturmak kolaydır.

Katı modelleme sistemleri model oluşturma sürecinde kullandıkları nesne temsil tekniklerine göre iki ana grupta toplanır

- Sınır temsili modelleme
- Yapısal Katı Gometri (YKG)

Yaygın olarak kullanılan bu iki tekniğin yanında süpürme temsili, hücre ayırıştırma şeması, temel eleman oluşturma gibi teknikler de kullanılmaktadır. Ayrıca, son yıllarda sınır temsili ve yapısal katı geometri metodlarını bir arada kullanan karma yöntemler de kullanılmaya başlanmıştır.

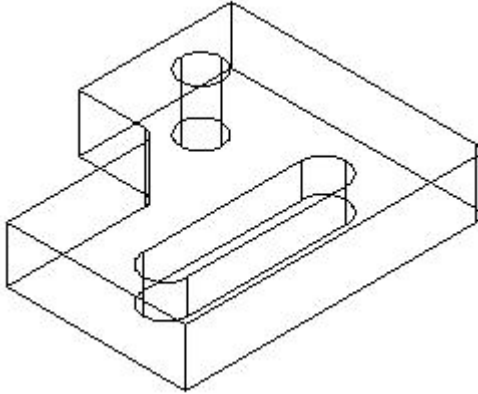
Bunların yanında, son yıllarda geliştirilen katı model programlarında unsur tabanlı sistemler kullanılmaktadır. Bu sistemler, nesne temsil teknikleri yardımıyla (çoğunlukla YKG) makine unsurlarının (delik, cep, kanal vb.) oluşturulması esasına dayalı olarak çalışırlar.

4.1. Tel kafes modelleme

Bir 3B'lu tel kafes model, köşeler ve köşeleri birbirine bağlayan kenarlardan oluşur. 3B'lu tel kafes model, x, y koordinat çiftlerinden oluşan 2B'lu tel çerçeve modellere z koordinat değerinin eklenmiş biçimidir. Üç koordinatla ifade edilen noktaları birleştiren doğruların oluşturduğu kapalı döngüler kümesi, 3B'lu tel çerçeve modeli oluşturmaktadır.

Nesnenin tel kafes modelle ifadesi nokta, doğru, yay, çember ve diğer eğri çeşitlerinden ibaret olan 2B'lu çizim elemanları ile gerçekleştirilir. Tel kafes model parçanın sadece dış kenar ve köşelerinin gösterildiği bir iskelet modeldir (Şekil 4.1).

Tel kafes modelde, nesnenin sadece dış hatları oluşturulmaktadır. Nesnenin içi modellenmemektedir. Dolayısıyla bir hacim içermemektedir. Ayrıca bu modelleme türünde yüzeyler modellenmemektedir. Bu yüzden yüzey oluşumu yoktur ve gerçek bir 3B nesneden bahsetmek mümkün değildir. Burada sadece nesneye 3B görüntü kazandırılmaktadır.



Şekil 4.1. Örnek tel kafes model

Tel kafes modelleme kolay ve basittir, bunun yanında bir takım yetersizliklere sahiptir. Bunlar [15, 125]:

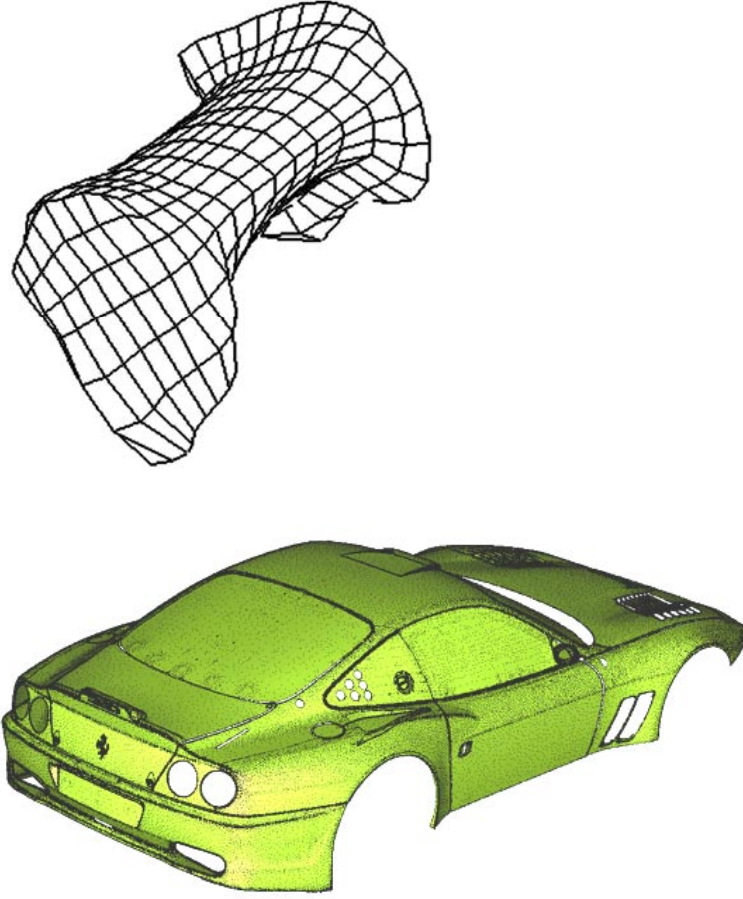
1. Tasarıma ait hacim ve kütle özellikleri hesaplanamaz. Düşük seviyeli veri içermesinden dolayı (çizgi, daire vb.) kesit alma, NC yolu oluşturma gibi işlemlerde problemler ortaya çıkabilir.
2. Fiili kenarlar bulunmamaktadır. Bu kenarlara örnek olarak profil veya eğrisel yüz derinlikleri sayılabilir.
3. Yanlış anlaşılması muhtemel modeller oluşturulabilmektedir.
4. Parça tanımlama zor ve çok işlem gerektirmektedir.

4.2. Yüzey Modelleme

Bir yüzey model, köşe, kenar ve yüzlerle tanımlanır. Bu teknikte nesneler, sınır yüzeyleri ile tanımlanmaktadır. Yüzey modellemenin tel kafes modelden farkı, nesne temsiliinde yüzey birleştirilmelerinin yer almasıdır. Özellikle eğri yüzeylerin modellenmesinde tercih edilen bir yöntemdir. Yüzey modellerle karmaşık parçaların modellenmesi mümkündür.

Yüzey modelleme ile nesnenin sadece dış kabuğu modellenmektedir. Bundan dolayı nesne tel kafeste olduğu gibi hacim içermemekte fakat 3B nesne görünümüne sahip

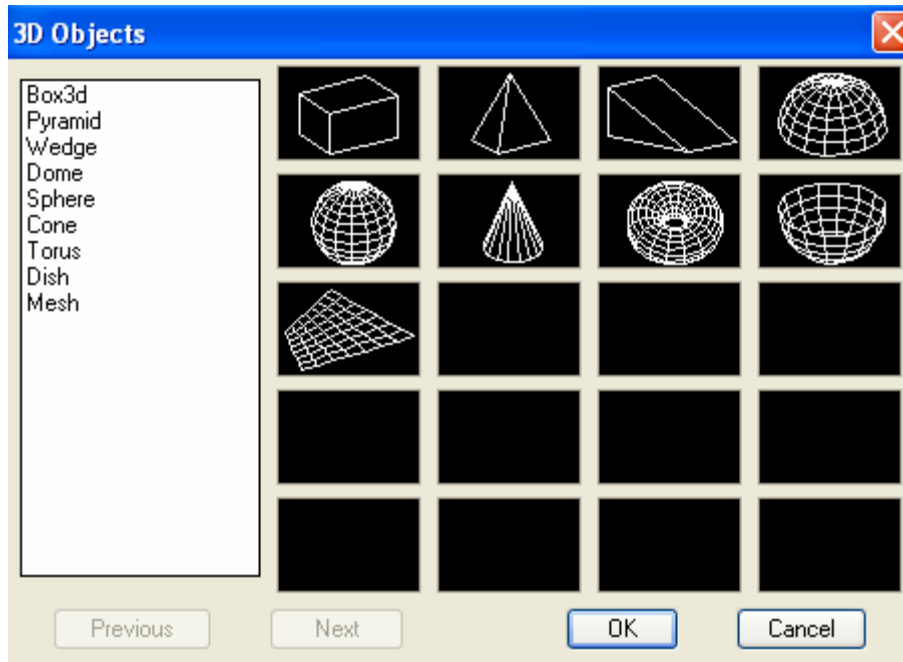
olmaktadır (Şekil 4.2). Bu açıdan yüzey modelleme sistemleri, tel çerçeveden daha iyi grafik etkileşim araçları sunmaktadır.



Şekil 4.2. Örnek yüzey modeller

Yüzey modelin en önemli dezavantajı topolojik bilgi vermemesidir. Ayrıca modelin yapılabirliği de garanti değildir [15].

Bir nesnenin yüzey modelinin oluşturulmasında, döndürülmüş yüzey, Bezier eğrileri ve B-Spline yüzey vb. elemanlar kullanılmaktadır. Ayrıca bazı CAD programlarında hazır yüzey elemanları bulunmaktadır (Şekil 4.3). Bunların birleştirilmeleri ile 3B nesne oluşturulmaktadır.



Şekil 4.3. AutoCAD’de hazır 3B yüzeyler

Yüzey modeller tel çerçeve modellerden daha avantajlı olmasına karşın birtakım dezavantajlara sahiptir. Bunlar [15]:

1. Modellenen yüzeylerdeki köşe ve kenarların bağlanabilirlik ve topolojileri hakkında bilgi depolanmaz. Bu yüzden parçanın içi ve dışı hakkında bir tahminde bulunmak mümkün değildir.
2. Modellenen parçanın hacim, kütle vb. özellikleri hesaplanamaz
3. Parça tanımlama tel çerçeve modellere oranla daha verimli sonuçlar verse bile tam bir tanımlama söz konusu değildir.

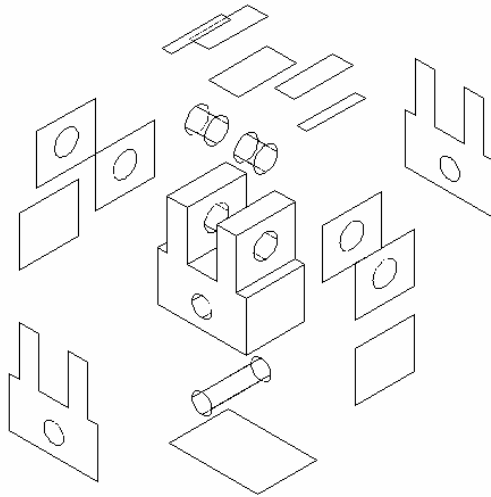
4.3. Katı Modelleme

Katı modelleme, tel kafes ve yüzey modellemenin yorumlanması güç olan modeller için elverişli olmamasından ötürü geliştirilmiştir [126]. Katı modelleme şemalarının geliştirilmesi ile; çoklu dik izdüşüm çizimleri, tel kafesler ve yüzey modelleme zorluklarına çözümler bulunmuştur. Katı model nesne görünümü açısından yüksek çözümler verir. Katı modellerin önemli bir avantajı diğer modelleme sistemlerinden

daha fazla geometrik ve topolojik bilgiyi kullanarak kütle özellikleri hesabı, sonlu elemanlar ağı üretme gibi işlemleri otomatik olarak yapabilmektedir [79, 126]. Bunların yanında bir katı modele, NC parça programı ve işlem planı gibi işlemler uygulanabilmektedir. Katı modelleme sistemleri, boyut, şekil ve ağırlık gibi hesaplamaları da gerçekleştirebilmektedir. Katı modelleme amaçlı kullanılan bir dizi temsil tekniği vardır. Katı modelleme temsil teknikleri esas olarak ikiye ayrılmaktadır: Sınır Temsili (BRep) ve Yapısal Katı Geometri. Bu iki temsil tekniğinin dışında kullanılan temsil teknikleri de bulunmaktadır. Bunlar: süpürme temsili, hücre ayrıştırma şeması ve analitik katı modelleme teknikleridir.

4.3.1. Sınır temsili (Boundary Representation-BRep) modelleme

Sınır temsilleri nesnelerin köşe, kenar ve yüzey sınırları ile temsil edilmesidir. Sınır temsili yüzey modelleme tekniklerine dayandırılmaktadır. Katı bir nesnenin sınır temsili, nesnenin iç ve dış taraflarını tanımlayan tüm yüzeylerin tanımlanması ile oluşturulur (Şekil 3.6).

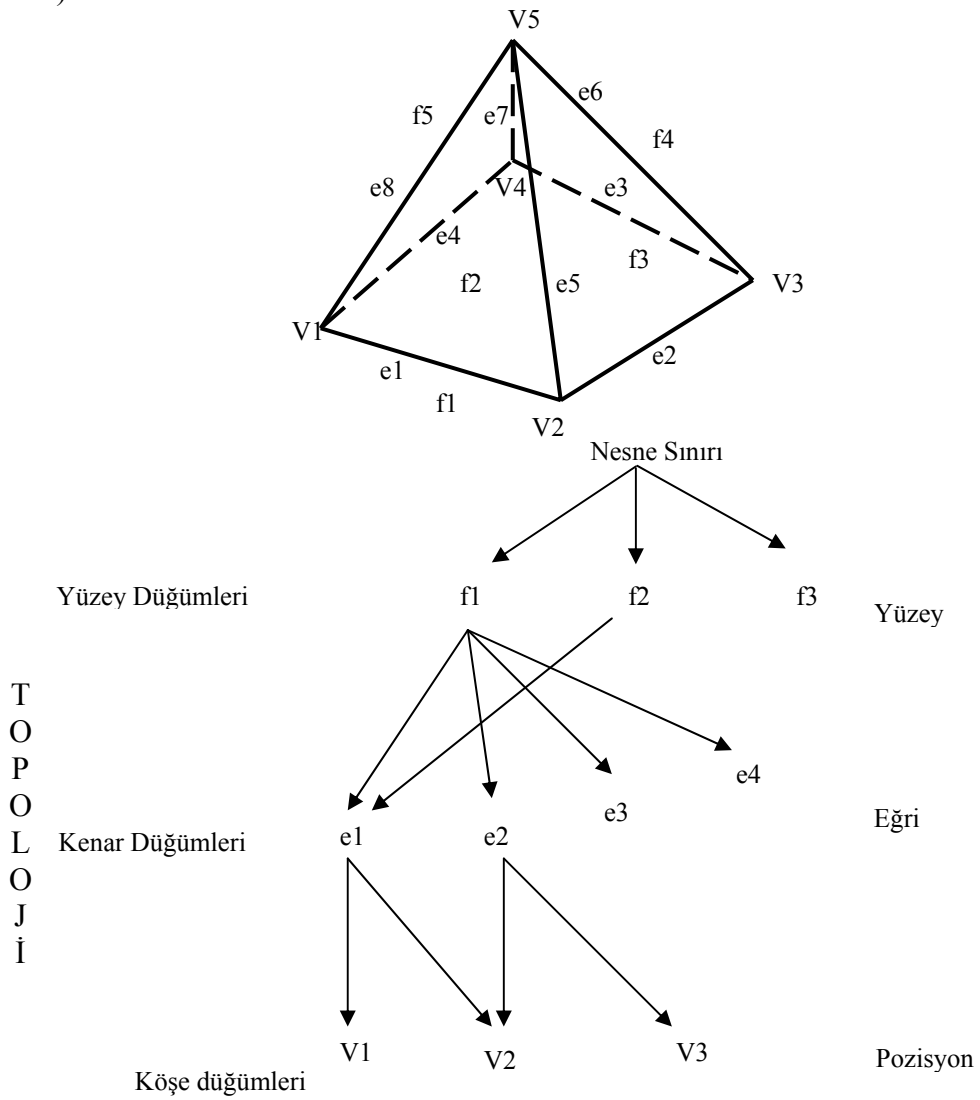


Şekil 4.4. Bir katı modelin sınır temsili [15]

Bir katı modelin sınır temsili, yüzeyler, kenarlar ve noktalar gibi bir katıyı tanımlayan basit yüzeylerin topolojisi olarak anlaşılabilir. Ayrıca parçanın

tanımlanabilmesi için kabuk (Shell) ve halka (loop) öğeleri kullanılmaktadır. Halka birbirine bağlı kenarları kapalı bir döngüsüdür. Kabuk ise birbirine bağlı yüzlerin oluşturduğu kapalı bir gruptur [79, 127].

Sınır temsili tanımlamalar topolojik ve geometrik tanımlamaları içermektedir. Geometrik tanımlamalar, nokta, eğri yüzey gibi elemanların geometrik olarak tanımlanmasıdır. Topolojik tanımlamalar ise köşe noktaları, kenar ve yüzeylerin birbiri ile bağlantı durumlarını kapsamaktadır. Değiştirme, yüzey normalleri için kabullenilir; katıya dışardan bakıldığında yüzey normalleri dışarıya yöneliktir (Şekil 4.5).



Şekil 4.5. Bir katı modelin sınır temsili - [126]

Teorik olarak sınır temsili ile her nesneyi modellemek mümkün iken, eğri yüzeylerin temsili, genelde zordur ve yaklaşık olarak poligonlarla ifade edilmektedir. Sınır temsili modelleme yöntemi bir takım avantajlara ve dezavantajlara sahiptir. Sınır temsili modellemenin avantajları aşağıdaki şekilde sayılabilir [126]:

1. Her yüzeye bireysel referans yapılabilir. Bu özelliği BRep'i toleranslamada kullanılabilir bir modelleme tekniği yapar
2. Sınır girdisi kolaylıkla yapılır
3. Model kontrolü geçerliliği nispeten kolaydır.

Bunların yanında, bilgisayar hafızasında fazla yer kaplaması, değişiklikler ortaya çıktığında güncelleştirmenin sıkıcı ve zor olması BRep'in dezavantajlarıdır.

4.3.2. Yapısal katı geometri (Constructive Solid Geometry - CSG)

Yapısal Katı Geometri (YKG) temsil sisteminde katılar, temel elemanlar olarak adlandırılan basit elemanların (küp, silindir, koni vb.) Boolean işlemciler (düzenli küme işlemcisi) kullanılarak organize edilmesi ile elde edilmektedir. Temel elemanlar parametrik denklemlerle tanımlanmıştır. Boolean işlemleri ekleme, çıkarma ve arakesit alma gibi işlemlerden oluşmaktadır. Bu yöntemle bir model oluşturmak çok kolay ve model tanımında kullanılan veriler çok azdır. Bir YKG ağacı Şekil 4.6'de görülmektedir. Bir YKG ağacında temel elemanlar ağaç dallarında yer alırken Boolean işlemciler, düğümlerde bulunmaktadır. YKG temsillerin avantajları ve dezavantajları aşağıdaki şekilde sayılabilir [126];

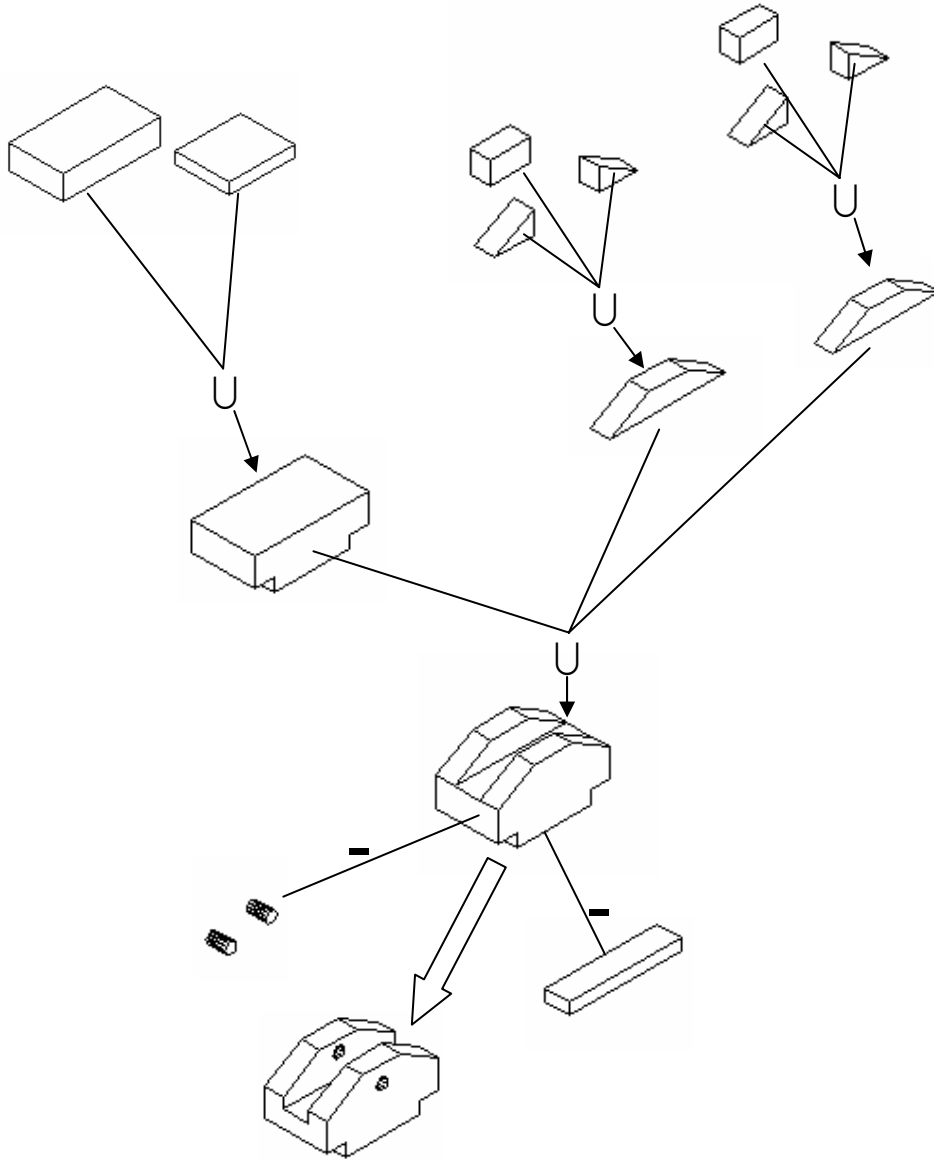
Avantajları:

1. Model tanımlamada kullanılan veriler ve girdileri kolaydır,
2. Temsiller matematiksel olarak sağlıklı bir şekilde tarif edilmektedir.
3. Bir model oluşturmak nispeten kolay ve öğrenilebilirdir.
4. YKG ağaçları katının oluşum aşamalarını gösterir.

5. Bilgisayarda fazla hafıza alanı işgal etmez.

Dezavantajları:

1. Modelin sınırının genişletilmesi çok zordur
2. YKG temsillerinin kullanımı temel eleman tipleri ile sınırlıdır ve serbest yüzeylerle katıların YKG ağaçları kullanılarak temsili zordur.
3. YKG temsilinin çözümü tek değildir.



Şekil 4.6. Bir YKG ağacı örneği

4.3.3. Diğer temsil teknikleri

Süpürme Temsili: Süpürme tekniği, poligon veya çok yüzlülerin tanımlanmış bir yörünge etrafında doğrusal veya dairesel süpürülmesidir. Silindirik parçalar 2B’lu bir profilin bir eksen etrafında döndürülmesi ile oluşturulur.

Temel eleman oluşturma metodu: Bir nesnenin potansiyel temel elemanla aynı topolojiye fakat farklı geometriye sahip olduğu fikrine dayanmaktadır. Örneğin topolojik olarak tanımlanan bir civatanın temel geometrisi bazı parametrelerin (kenar sayısı, uzunluk, adım, çap) girilmesi ile oluşturulabilir.

Hücre ayrıştırma şeması: Bu metodda katı nesne veya bunun toplam hacmi, küçük hücrelere ayrıştırılarak temsil edilir. Genelde bu yaklaşık bir temsildir. Üç çeşit hücre vardır. Bunlar, boş, dolu ve kısmen dolu hücrelerdir. Genelde hücreler küp şeklindedir. Kısmi hücrelerde daha sonra tekrar, kısmi, boş ve alt hücrelere ayrıştırılmaktadır. Bu hücrelerin tümü parçanın tamamını oluşturur.

Analitik katı modelleme (ASM): Bu metod; yüzeyleri temsil ve kullanılan tensor çarpım metodu s , t ve u gibi parametrelerle üç boyutlu parametrik uzaya genişletilmiştir. Bu, bir boyutlu parametrik uzayı bir parametre ile (t gibi) ve bir yüzeyi iki boyutlu bir uzayda iki parametre ile (s ve t) gibi temsile benzemektedir. Spline ve yama eğri parçaları ve yüzey yamaları oluşturma denklemleri, ASM’de geçerlidir.

Çizelge 4.1 Katı modelleme tekniklerinin karşılaştırılması - [2, 128].

	BREP	CGS	Süpürme	ASM	Hücre ayrıştırma	Temel eleman oluşturma
Hassasiyet	Yüksek hassasiyet için eğri yüzey desteği gerekli	Yüksek hassasiyet için temel elemanlar çok yüzölçü olmamalı	Yüksek hassasiyet	İyi hassasiyet	Genelde hassasiyetli uygulamalarda kullanılır	Süpürmeye benzer
Alan	Teorik olarak her nesneyi destekler	BREP'e benzer	Bazı alanlarda kullanılmaktadır	Bilgisayar sal etki olarak orta düzeyli	Yaklaşık her nesne	Süpürmeye benzer
Teklik	Hayır	Hayır	Dikkatli tanım gerekir	Hayır	Basit ızgara ve genelde tek nesneler oluşturur	Süpürmeye benzer
Geçerlilik	Kontrolü güç	Kontrolü kolay	Kontrolü kolay	Kontrolü kolay	Kontrolü kolay	Kontrolü kolay
Kapanırlık	Orta	Orta	Yapılması güç	Orta	Orta	Yapılması güç
Uyumluluk ve etkinlik	Dahili işletmelerde iyi	Kullanıcı arabirimlerin de iyi	Arabirimler de iyi	Dahili kullanımda iyi	Arabirimlerde iyi	Arabirimler de iyi

4.4. Tasarımda Unsur Kavramı

Klasik geometrik modelleme, katı modelleri geometrilerine bağlı olarak tanımlamaktadır. Bu sistemlerde katı model, parça geometrisine bir hacim kazandırılarak oluşturulur. Çıkıntı ve deliklerin oluşturulması ise bazı ilave işlemler sonucunda gerçekleştirilir. Bu işlemler yine parça geometrisine bağlı olarak yapılır.

Bu yöntem, geometrik modeller (tel kafes veya katı) oluşturmaya rağmen 3B nesneleri tam olarak tanımlamamaktadır. Bir tasarımcı istenen fonksiyonları gerçekleştirebilmek için geometrik unsurlara bağlı olarak düşünmek zorundadır [129]. Örneğin bir üretim planlaması sistemi, üretilebilecek unsur şekilleri hakkında bilgi sahibi olmayı gerektirir. Aynı şekilde montaj planlama sistemi iki elemanın montajını kolaylaştıracak unsurları arar. Bu arayış, tasarımda unsur kavramını ortaya çıkarmıştır.

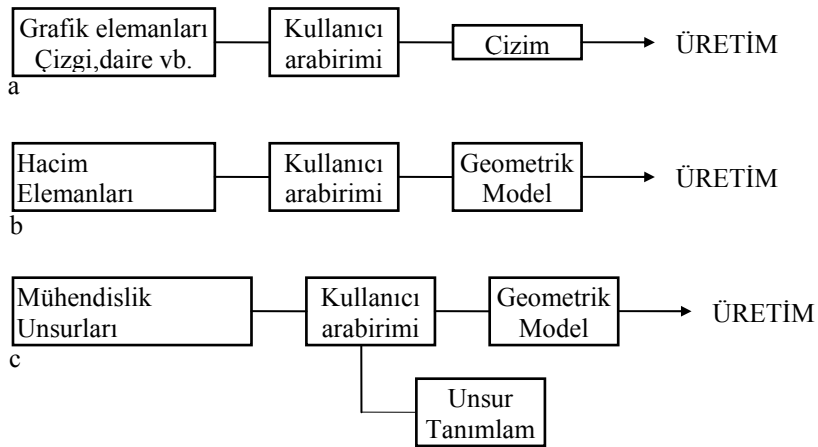
Unsur, bir geometri veya ürünü tanımlamak için bloklar inşa etmek olarak tanımlanabilir. Bunun yanında, birbirlerine bağlanmış yüzeylerin oluşturduğu özel bir işlem tipi olarak da tanımlanabilir [130]. Unsur, katı model oluşturma yaklaşımlarından birisidir. Unsur yaklaşımları, YKG ilkeleri (küp, koni, silindir vb.) kullanan ve katı modellerin geometrilerine bağlı olarak tanımlanması üzerine yoğunlaşan yaklaşımlara göre daha genel çözümler sunması nedeniyle kolaylık sağlar [131]. Tasarımda unsur kullanımı karmaşık parçaların yaratılmasını kolaylaştırmıştır.

Bir BDT yazılımı için unsur, bazı anlamlar içeren bir alt biçim olarak tanımlanabilir. Bunlara örnek olarak delikler, çıkıntılar ve kanallar verilebilir. Unsur kavramı, pah, vida gibi geometrik biçim veya tasarım unsurları ile başlamış ve zamanla kapsamı genişlemiştir [132]. Bu kapsam genişliği, unsurların alt sınıflara ayrılmasını beraberinde getirmiştir. Araştırmacılar unsurları değişik şekilde sınıflandırmışlardır. Genel bir sınıflandırmayla unsurlar aşağıdaki şekilde sınıflandırılabilir [133]:

1. Şekil unsurları
2. Tolerans unsurları
3. Montaj unsurları
4. Fonksiyonel unsurlar
5. Malzeme unsurları

Bunun yanında, tasarım ve üretimde kullanılacak unsur tiplerini dikkate alan sınıflandırma yaklaşımları da vardır [134]. Bunlar, biçim unsurları, desen unsurları, birleştirme, özellik ve uygulama unsurları olarak sayılabilir. Biçim unsurları vida, delik vb. gibi unsurları, desen unsurlarına ise dikdörtgen ve dairesel desenler örnek olarak verilebilir. Özellik unsurları yüzey işleme kalitesi gibi ayrıntıları ele alır, uygulama unsurları ise montaj sırası gibi unsurları ele almaktadır [134, 135]

Şekil 4.7’de Unsur tabanlı tasarımın BDT yapısı içerisindeki konumu, çizim sistemi ve aralarındaki fark görülmektedir.



Şekil 4.7. BDT Sisteminin yapısı a - Çizim Sistemi b - Modelleme Sistemi c - Unsur Tabanlı Tasarım [136]

4.4.1. Unsur oluşturma yöntemleri

Unsur tanımanın esası Bilgisayar Destekli Tasarım ve Bilgisayar Destekli Üretim (BDT / BDÜ) arasındaki entegrasyonu sağlayan BDİP’nin işlem sırası, işlem tipi, hacim vb. bulmak için kullanılan veri tabanı dosyasının hazırlanmasını sağlamaktır [80]. BDT ve BDÜ arasındaki entegrasyon, büyük ticari rekabetin söz konusu olduğu bu alanda önemli bir üstünlüktür. Burada unsurlarla parça tasarımının çözüm uzayı genişlerken, BDT/BDÜ entegrasyonu ile de süreç otomatikleştirilmektedir.

Bir unsur modelleme sistemi, unsur modellerini yaratmak için kolaylıklar içermelidir. Bu kolaylıkların kullanılabilirliği bir unsur tabanlı sistemin kullanılabilirliği üzerinde büyük bir etkiye sahiptir. Bir unsur tabanlı model, birbirleriyle ilişkili olan iki bileşenden oluşur, bunlar; unsur modeli ve geometrik modeldir. Geometrik model YKG, sınır temsili veya diğer temsil teknikleriyle oluşturulan nesnenin temsili içerir. Bir geometrik model topolojik varlıklar, topolojik graf, geometrik nesneler, topolojik ve geometrik noktalardan oluşur [133].

Unsur modeli ise, bilgi kümeleri, unsur özellikleri ve diğer yüksek seviyeli bilgiyi içerir [133]. Bir unsur modeli, şekil tanımı, boyut özellikleri, unsur pozisyonu, geometrik sınırlandırmalar ve geometrik olmayan özelliklerden oluşur.

Unsurlar, geometrik modeller üzerinde oluşturulurlar. Geometrik modeller, 3B model oluşturmada kullanılan nesne temsil teknikleri kullanılarak oluşturulurlar. Bunlar, YKG, sınır temsili ve bu iki temsil tekniğini bir arada kullanan karma yaklaşımlardır.

YKG yaklaşımı kullanarak unsur oluşturma

Bu yaklaşımda unsurlar geleneksel YKG ilkeleri (küre, koni, kutu, silindir vb.) kullanılarak oluşturulur. YKG ilkelerine uygulanan birleştirme, çıkartma veya kesiştirme gibi boolean operasyonlarıyla unsur modeli oluşturulmaktadır. YKG kolay model oluşturabilmesi ve unsur tabanlı modelin yazım kolaylığı sebebiyle unsur tanımlamada tercih edilmektedir. YKG modelleri vasıtasıyla tanımlanan unsurların dezavantajı YKG temsillerinde benzerlikler olmasıdır [79].

Sınır temsili (B-rep) yaklaşımı unsur oluşturma

Sınır temsili (B-rep), YKG yöntemine göre daha çok kullanılan bir katı model oluşturma yöntemidir. Sınır temsili, nesne topolojisini daha iyi tanımladığı için tercih edilmektedir. Bu temsil tekniğinde yüz, kenar ve köşe noktalarının açık bir şekilde ifade edilebilmesi, sınır değerlerine daha kolay ulaşılması bu tercihin en önemli

sebebidir. Burada dikkat edilmesi gereken husus, unsur bilgilerini depolamak için bir veri yapısının oluşturulmasıdır. Sınır temsilinin en önemli dezavantajı hacimsel unsurlarla temel geometrik unsurlar arasında anlamlı bir ilişki kurulamamasıdır [134].

Karma (YKG-Rep) yaklaşım kullanarak unsur oluşturma

Bu yaklaşımlarda unsur modeli, YKG ve sınır temsili yaklaşımların üstün yanları birleştirilerek oluşturulur. Bu yöntemde daha kolay parça tanımlanabilmekte ve yüksek verim elde edilmektedir. Bu yaklaşımda YKG'nin unsur operasyonlarındaki (silme, ekleme, çıkarma vb.) üstünlüğü ile sınır temsilinin sınır değerlerine ulaşmadaki esnekliği bir arada kullanılarak yüksek verim elde edilmektedir.

Unsur tabanlı modeller oluşturmak için pek çok yöntem geliştirilmiştir. Bunları üç ana başlık altında toplamak mümkündür. Bunlar [133]:

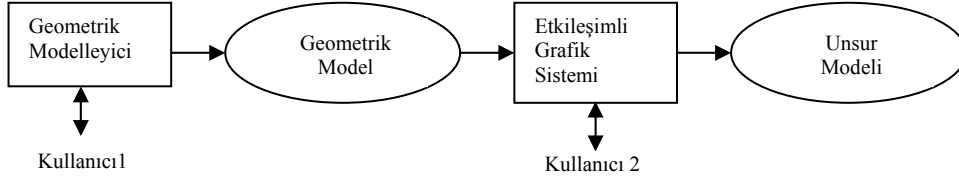
- etkileşimli unsur tanımlama,
- otomatik unsur tanıma ve
- unsurlar ile tasarımıdır.

Unsurlar ile tasarım diğer iki yöntemden ayrılmaktadır. Etkileşimli unsur tanıma ve otomatik unsur tanımlama yöntemlerinde ilk aşamada geometrik model oluşturulur, daha sonra bu model üzerinde unsur modeli oluşturulur. Unsurlar ile tasarımı ise, geometrik model, unsur kütüphanesi kullanılarak unsurlarla oluşturulur. Yani burada unsur modeli aynı zamanda geometrik modeldir.

Etkileşimli unsur tanımlama

Bu yaklaşımda önce bir katı modelleyici (yazılım programı) kullanılarak geometrik model oluşturulur. Bu model YKG, sınır temsili hatta tel kafes olabilir. Daha sonra, geometrik model bir dosya halinde etkileşimli grafik programa gönderilir (Şekil 4.8).

Geometrik model, kullanıcının parça geometrisini anlamasına ve onun üzerinde unsur oluşturmaya izin verecek bir resim haline dönüştürülür.



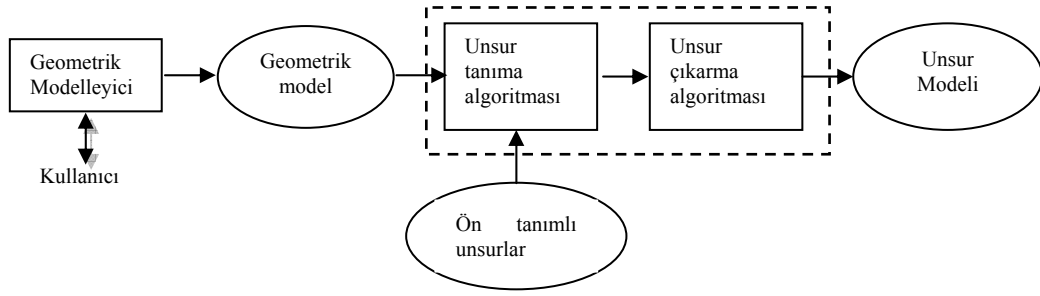
Şekil 4.8. Etkileşimli unsur tanımlama [133]

Kullanıcı, etkileşimli grafik program aracılığıyla, bir unsuru oluşturmak için ihtiyaç duyduğu köşe, kenar, yüzey bilgilerini toplayabilmektedir. Bu bilgi bazen tolerans, delik çapı gibi yüksek seviyeli bilgileri de içerebilir [133]. Bu bilgiler yardımıyla unsur modeli oluşturulmaktadır.

Etkileşimli unsur oluşturma, bazen unsur sınıflandırma oluşumlarını içeren unsur kütüphanesiyle birleştirilecek şekilde geliştirilirler. Unsur sınıflandırmasının kütüphane tanımı çeşitli özellikleri içerir [79].

Otomatik unsur tanıma

İşlem planlama, NC parça programlama gibi uygulamalar tarafından kullanılacak gerekli veri girdisini insan katkısı olmaksızın sağlamak için değişik teknikler geliştirilmiştir. Burada amaç, kullanıcı katkısından kaynaklanacak uzun işlem zamanı ve hataları ortadan kaldırmaktır. Otomatik unsur tanıma yaklaşımında öncelikle tasarımcı tarafından geometrik model oluşturulur. Daha sonra önceden tanımlı unsur bilgilerini kullanan bir unsur tanıma algoritması ile unsurlar tanımlanır. Son işlem unsur çıkartma algoritması ile geometrik model üzerinden unsurların çıkartılıp unsur modelinin oluşturulmasıdır (Şekil 4.9).



Şekil 4.9. Otomatik unsur tanıma [133]

Unsur tanıma algoritmaları, veri tabanı unsur bilgilerini hazır olarak saklamaya uygun olmayan BDT sistemlerinde model üzerindeki unsurları tanımak için geliştirilir. BDT ve BDÜ’de unsur tanıma ve çıkartma algoritmalarının en önemli rolü bu iki sistemi bütünleştirmektir.

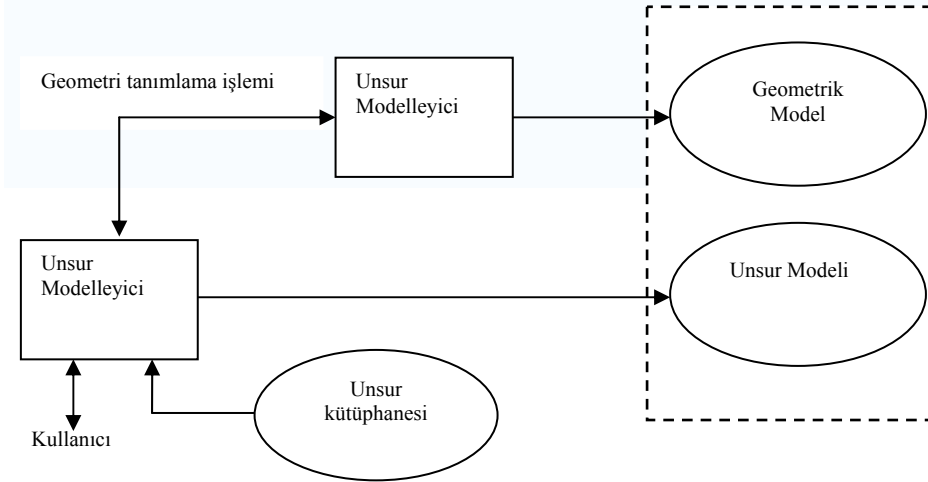
Otomatik unsur tanıma sistemleri, üretim otomasyonunun başarılmasında çok önemli bir role sahiptir. Bu sistemler, unsur olarak tanımlanacak öğeleri toplama, bu unsurlar için uygun işleme yöntemini seçme ve unsurların sağlayamadığı lüzumlu bilgileri sağlama adımlarıyla ifade edilebilecek üç aşamalı bir yöntemi takip ederler. AFR sistemi, işlenebilir unsurların çıkarılması, bir operasyon planının ve takım yolunun oluşturulması adımlarını içermektedir [137].

Unsurlar ile tasarım

Unsurlar ile tasarımda, unsur yapıları özel operasyonlar (boolean) kullanılarak bir model içinde doğrudan oluşturulur. Unsurlar ile tasarım, kaliteli ve iyi bir tasarım için bütün fonksiyonel ihtiyaçların tatmin edilmesini hedefler [138]. Bir tasarımcı ürünün unsurlara bağlı fonksiyonel ihtiyaçlarını görür. Detaylandırılmış tasarım, unsurların ilave edilmesiyle üretim ve montajı desteklemek için geliştirilmiştir [79].

Bu yaklaşımın diğer yaklaşımlardan en önemli farkı, geometrik modelleyicinin kullanılmamasıdır. Geometrik model ve unsur modeli, unsur modelleyici tarafından

oluşturulur (Şekil 4.10). Parçalar, unsurlardan ve unsurlarla oluşturulmuş geometrik modellerden doğrudan oluşturulur.



Şekil 4.10. Unsurlar ile tasarım [133]

Bu sistem unsur kütüphanesine ihtiyaç duyar. Unsur modelleyici unsur kütüphanesinden beslenerek unsur modelini oluşturur. Unsur kütüphanesi, örnek unsurlarla (delik, cep, kanal vb) ilgili özel boyutlar, yerleştirme parametreleri, sınırlayıcılar, ilişkiler ve nitelikleri içerir [133]. Unsur modeli, işlenmemiş parça ve unsurlar arasında boolean operasyonları sonucu oluşturulur.

4.5. Değerlendirme

Tez kapsamında yürütülen çalışmada katı model oluşturma aşamasında nesne temsil tekniği olarak hacimsel yaklaşımlar tercih edilmiştir. Katı temel elemanların oluşturulmasında, süpürme teknikleri ve klasik YKG elemanları birlikte kullanılmaktadır.

Tez çalışması çerçevesinde geliştirilen sistemde, 2B izdüşümlerin yorumlaması ile oluşturulan ana katı üzerine, US tabanlı karar mekanizması ile oluşturulan katı temel elemanlar boolean operasyonları ile birleştirilip çıkartılmaktadır. Program bu

yapısıyla, birden fazla nesne temsil tekniğini kullanabilen geometrik bir modelleyicidir.

Çalışmada katı temel elemanlar, klasik YKG temel elemanları (koni) ve süpürme operasyonları (doğrusal ve eksenel) ile oluşturulmaktadır. Burada boolean operasyonları (birleştirme, çıkartma ve kesiştirme) etkin bir şekilde kullanılmaktadır.

Katı temel elemanlar oluşturma aşamasında unsur kavramından bahsedilse de unsurların etkin kullanımı yoktur. Burada bir unsur kütüphanesi kullanılmamaktadır. Burada geliştirilen sistem, unsur tabanlı tasarım yöntemlerinden etkileşimli unsur tanımlama sistemine benzemekle birlikte, katı model oluşturma aşamalarında unsur kullanımı yoktur.

5. PROGRAM GELİŞTİRME VE PROGRAMLAMA TEKNİKLERİ

Bilgisayarlar işlemlerin kendi anlayacağı dilde (makine dili) açıkça yazılmasını isterler. Bu işlemlerin makine dilinde bir programcı tarafından yapılması yerine programlamanın yapısal biçimde bir dil aracılığıyla yapılmasını ve daha sonra çevrilerek bilgisayara anlatılması işlemini programlama dilleri üstlenirler. Bu çevirme işlemine derleme ya da yorumlama denir. Bir program bilgisayara ne yapması gerektiğini söyleyen bir dizi komuttur.

Gerçek hayattaki durumların modellenmesi olarak da tanımlanabilecek olan programlama yardımıyla bu modeller bilgisayar ortamına aktarılır. Yazılım olarak da ifade edilebilen bilgisayar programları, bilgisayarın donanım kısmına hayat verir. Yazılım, donanım dışında kalan ve kullanıcının kodlama yaparak oluşturduğu programlardır.

Programlar olmadan bilgisayar herhangi bir işlemi gerçekleştiremez. Bilgisayarı işe yarar hale getiren yazılımlar (word, exel, AutoCAD vb), yazılım geliştirme dilleri (programlama) ile hazırlanmaktadır. Bilgisayar içerisinde kullanılabilen program geliştirmeye yönelik faaliyetler “uygulama geliştirme” olarak da isimlendirilebilir.

Uygulamalar, yazılım geliştirmede kullanılan programlama dilleri (C, C++, Visual Basic, LISP vb.) kullanılarak hazırlanmaktadır. Bu diller, geliştirilen programların bilgisayar ortamında doğrudan çalışabilmesini sağlamak için program kodlarını makine diline çevirirler.

Amaca uygun olarak seçilen programlama dili, programlama diline uygun programlama tekniği ve doğru algoritma, yazılım geliştirmenin önemli unsurları olarak sayılabilir. Algoritma; bilgisayar programlarının tasarlanması aşamasında yararlanılan ve tasarım sürecindeki bütün işlemlerin ifade edildiği işlemler dizisidir. Program, genel anlamda yazılımın bir alt kümesidir; hiyerarşinin en üstünde yazılım vardır. Daha altında program, onun altında belirli kısmi bir işi yapacak algoritma

gelir. Algoritma genel olarak, belirli bir iş veya görevin adım adım ortaya konmasıdır. [139] Algoritma, programın her adımının hedefini kesin ve net bir şekilde açıklayan ifadelerle oluşturulur. Bu ifade herhangi bir programlama diline bağlı olmamalıdır. Bu amaçla algoritma günlük konuşma dilinde hazırlanır.

Bir problemin çözümü için birden çok algoritma geliştirilebilir. Hangi algoritmanın tercih edileceği, doğrudan uygulamaya ve programcının birikimine bağlıdır.

5.1. Programlama Teknikleri

Programlama teknikleri, programlama dilinin belirlenmesinde en önemli unsurdur. Bir program geliştirilirken kullanılan program ve program tekniği arasında paralellik olmalıdır. Aksi takdirde doğru sonuç elde edilemez. Her programlama dili bir programlama tekniğini kullanarak faaliyet gösterir. En çok kullanılan programlama teknikleri dört ana başlık altında incelenebilir:

1. Yapısal programlama (Procedural)
2. Fonksiyonel programlama
3. Nesne tabanlı programlama
4. Modüler programlama

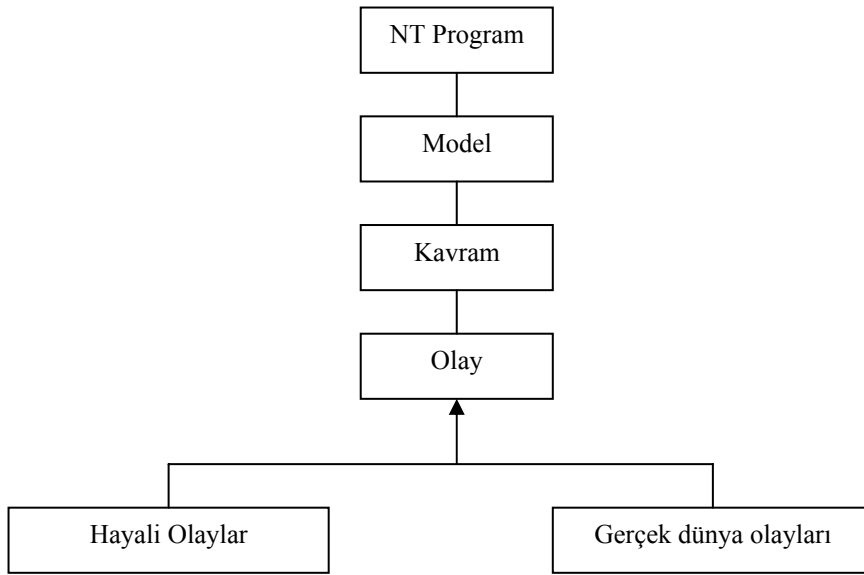
5.1.1. Nesne tabanlı programlama

Nesne tabanlı programlama, uygulamaların nesneler kullanarak yani bir çok nesnenin bir araya getirilmesiyle oluşturulmasıdır. Nesne tabanlı programlama, gerçek dünyadaki nesnelerin bir yazılımın sunduğu soyut modelde bire bir karşılığının bulunması temeline dayanır.

Nesne tabanlı programlama (NTP) ile gerçek varlık ve ilişkilerin modellenmesi mümkündür. NTP'nin bu yapısı özellikle kurumsal bilgi akışlarının modellendiği iş

akışı (work flow) yazılımlarında ciddi üstünlük sağlamaktadır [140]. Modüler yapıda program yazımı NTP'nin bir diğer üstünlüğüdür.

NTP'de bir program fiziksel bir modelle ilişkilidir. Bu model ya gerçek dünyayı taklid eder ya da gerçek dünyanın hayali bir parçasıdır [141] Şekil 5.1.NTP ve gerçek modeller arasındaki ilişkiyi göstermektedir.



Şekil 5.1. Nesne tabanlı programlama yazılımı [141]

NTP'nin en önemli özelliği yordamlara getirdiği kolaylıktır. Burada yordamlar, uygulama koduna değil sınıflar içine yazılmakta ve bu sınıflardan türetilen nesneler üzerinden çağrılmaktadır. Nesne tabanlı programlama, veri ve nesneler etrafında organize edilen bir programlama modelidir. NTP'de ilk adım kullanmak etmek istenilen bütün nesneleri tanımlamaktır ve birbirleriyle nasıl ilişkili olduklarını tanımlamaktır. Veri modelleme olarak bilinen bir uygulamadır. Bir nesne bir kez tanımlandığında, nesnenin bir sınıfı olarak genelleştirilir ve kullanılabilen herhangi bir mantık dizisi olan veri çeşidi tanımlanır.

NTP'de ilk adım kullanılacak nesneleri ve nesnelerin birbirleriyle ilişkilerini tanımlamaktır. Buna veri modelleme denir. Bir nesne bir kez tanımlandığında,

nesnenin bir sınıfı ve veri çeşidi olarak genelleştirilir. Bu genelleştirme, nesnenin kullandığı mantık zincirini de kapsar.

Sistemdeki kavramlar (şeyler, aktiviteler, ilişkiler) ve gerçek olaylar sınıflar ve nesnelerle temsil edilirler [142]. NTP yöntemi ile hazırlanan bir bilgisayar programı sınıflar, nesneler ve metodlardan oluşur.

Sınıflar

Sınıflar, NTP'nin merkezini temsil eder ve tasarlanan nesnenin bir örneğidir [143]. Sınıflar, nesnelerin yaratıldığı şablonlardır. Sınıflar, ortak özellikleri ve davranışları olan nesnelerin söz konusu ortak özelliklerini ve ortak davranışlarını barındırır [bora Güngören]. Her sınıf içinde, o sınıftan üretilecek nesnelerin özellikleri, davranışları tespit edilir. Örneğin, dişli kutusu isminde bir sınıfa ait nesneler, dişliler, rulmanlar, emniyet segmanları gibi parçalardan oluşacaktır. Sınıflar, nesne üretme yanında başka bir özelliğe daha sahiptir. Sınıflar arasında akraba ve iş ilişkileri kurmak mümkündür.

Nesneler

Bir nesne, bir yazılım yığınıdır; davranış ve durumla ilişkili. Yazılım nesneleri çoğu kez, günlük yaşamda bulunabilen gerçek dünya nesnelerinin modeli olarak kullanılır.

Bir nesne, uygulama çalışırken hafızada ya da başka bir kayıt ortamında (dosya, veritabanı vb.) bulunan ve ait olduğu sınıfın tanımladığı davranışa uyan bir örnektir [140]. NTP'de nesneler sınıflardan üretilir. Nesneler, sınıfların aksine canlıdır ve kimlikleri vardır. Aynı sınıftan üretilmiş iki nesnenin sahip olduğu değişkenler değişik değerlere sahiptir.

Metodlar

Bir metod, verilerin özel bir kümesi üzerine uygulanacak bir operasyonu tanımlamak için kullanılır [143]. Metodlar dis dünyaya açılan kapılardır. Bir nesnenin içinde ne olduğu, nesnenin nasıl çalıştığı dışarıdan görünmez. Herbir yöntem, nesnenin yapabileceği bir davranışı simgeler. Program içinde nesneler ile etkileşim metodlar aracılığı ile olur. NTP’de gerçek dünyanın taklidi metodlar aracılığı ile gerçekleştirilir. Bir nesne yapması gereken davranışın ne olduğunu metoddan aldığı mesajdan alır (tıkla, sürükle, bırak vb.).

NTP yaklaşımı, programları sadece bir komut dizisi veya fonksiyon kümesi olmaktan çıkartmıştır. NTP’nin gerçek hayata dayanan yapısı, bu tekniği kullanan programların (Visual BASIC, JAVA, C++ vb.) anlaşılabilirliğini arttırmıştır. NTP ayrıca, yüksek esneklik ve bakım kolaylığı sunmaktadır. Bu sebepten dolayı, günümüzün geniş çaplı yazılım projelerinde yaygın olarak kullanılmaktadır.

5.1.2. Fonksiyonel programlama

Bu dillerle yazılan programlar fonksiyonlardan oluşurlar ve tekrar temellidirler. Fonksiyonel programlamada matematiksel fonksiyonlar taklid etmek amaçlanır. Problemin matematiksel fonksiyonların değerlendirilmesi şeklinde tanımlanıp çözüldüğü bir programlama türüdür. Fonksiyonel programlamanın bir diğer özelliği değişebilir veri ve durumlardan kaçınan bir programlama tekniği olmasıdır.

Fonksiyonel proglamlarla genelde derleme yapılamaz ve bir yorumlayıcı aracılığıyla kullanılırlar. Bu programlar yorumlayıcı aracılığıyla kullanıldığı için “exe” dosyalarını oluşturmak zordur.

Fonksiyonel programlama tekniğinde problem ayrıştırılarak çözülür. Problem, küçük fonksiyon gruplarına bölünür. İdeal olarak, fonksiyonlar sadece girdi alır ve çıktı

üretir. Fonksiyonel diller programlamada veri için doğal bir seçenek sağlar [144]. Ayrıca, verilen girdi için çıktı üretirken herhangi bir müdahale yapılamaz.

Fonksiyonel bir programda girdinin akışı girdi fonksiyonları aracılığıyla gerçekleştirilir. Her bir fonksiyon kendi girdisini çalıştırarak çıktı üretir. Fonksiyonel programlama NTP'nin tersi gibi düşünülebilir. Fonksiyonel programlama durum değişimlerinden mümkün olduğunca kaçınmak ister ve fonksiyonlar arasında veri akışı ile çalışır.

Fonksiyonel programlamada kod okunurluğu artar. Yapıya bakan biri dilin temellerine hakimse hangi işin nerde bittiğini, süreçlerin nerde ne iş yaptığını anlayabilir. Fonksiyonel programlamanın özellikleri aşağıdaki şekilde sayılabilir;

1. Fonksiyonel programlamanın diğer programlama yaklaşımlarından en büyük farkı değişkenlere ihtiyaç duymamasıdır.
2. Bir fonksiyon aynı parametreler verildiğinde daima aynı sonucu üretir
3. Atamalar basittir
4. Programlar otomatik olarak eşzamanlı yapılabilir
5. Basit bir yapıya sahiptirler
6. En büyük dezavantajları verimsiz çalışmalarıdır.

Programların fonksiyonel stilde yazılmasının bazı avantajları vardır. Bunlar;

- Biçimsel kanıtlanabilirlik
- Modülerlik
- Oluşturulabilirlik
- Hata ve test kolaylığı olarak sayılabilir.

Fonksiyonel diller basit ve temiz bir programlama yapısına sahiptir. Problemi küçük parçalar içine bölebilmeleri, karmaşık programlarda programı okuyabilme ve hataları kontrol etmede büyük avantajlar sağlar. Bu özellik ayrıca, fonksiyon uygulamaları ile

bütün hesaplamaların çalıştırılmasına ve programlama sürecinin daha açık ve özet ifade edilmesine yardımcı olur [145].

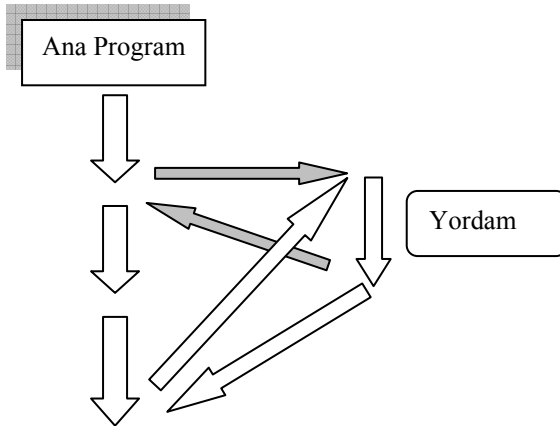
Fonksiyonel programlamanın tercih edilmesinin bir diğer nedeni imperative programlardan kolay olmasıdır. Fonksiyonel programların çalışmaması durumlarında hatanın, tüm programla uğraşmadan fonksiyonlar içinde halledilebilmesi de önemli bir avantaj olarak görülmektedir [146].

Fonksiyonel programlama dilleri ticari yazılım geliştirmeden ziyade akademik çalışmalarda kullanılır. Fakat, bazı fonksiyonel programlama dilleri endüstri ve ticari uygulamalar içeren çalışmalarda da kullanılmaktadır. Haskell ve LISP en çok kullanılan fonksiyonel programlama dillerindendir.

5.1.3. Yordamsal (Prosedürel) programlama

Yordamsal programlamada yazılımlar birbirini çağıran bir dizi yordam (procedure) ve işlevler topluluğu olarak geliştirilir. Her yordam ve işlev kendi yerel verisini, yine kendi yerel değişkenlerinde tutar. Paylaşılması gereken veriler yordam çağırma komutlarında parametre olarak yordamdan yordama geçirilir. Parametrelere sığmayacak büyük veriler ise genel (global) değişkenler içerisinde herkesin kullanımına açılır. Yordamsal programlama yaklaşımının ilke çizimi aşağıdadır:

Yordamsal programlama tekniğinde, program içinde tanımlanan durumlar prosedürler halinde tanımlanır. Her bir global veya yerel durum için bir yordam tanımlanır. Programın işleyişi sırasında, ihtiyaç olduğu zaman programın akışı ana programdan yordama yönlendirilir, yordamda işlem bittiği zaman ana programa dönülür ve program kaldığı yerden devam eder (Şekil 5.2).



Şekil 5.2. Yordamsal programlamada ana program –yordam ilişkisi

Bu teknikte, programlar yordamlar ve alt yordamlarla yazıldığı sürece daha yapısal ve hatasız olarak çalışır. Eğer, bir yordam doğruysa, her zaman doğru sonuçlar üretmek için kullanılır. Programın yordamlar halinde yazılması, hata durumunda gereksiz aramaları da ortadan kaldırır.

Bu yöntemde hiyerarşik bir yapıdan bahsedilebilir. Veri prosedürlerde işlenir, bir kez program sona erdiğinde sonuç verisi gösterilir. Bütün bu işlemin akışını ana program yönlendirir. Yapının en üstünde ana program, altında ise prosedürler bulunur. Ana program, yordamlara geçişi ve veri kontrolünü sağlar. Program karmaşıklığı orta seviyede ve verimli bir program yönetimi sağlamaktadır.

Yordamsal programlamada verinin saklanma biçimi, veriye erişim ve verinin işleme biçimini doğrudan etkiler. Verinin işleme biçimi (dizi boyutu vb.) veya türü değiştiğinde veriyle ilişkili tüm yordamların kontrol edilip, değişikliğin yapılması gerekir. Bu işlemler sırasında kodlarla ilgili ekleme ve çıkartmaların yapılması gerekeceği göz önünde tutulduğunda, karmaşık ve hataya açık bir programlama tekniği olduğu söylenebilir.

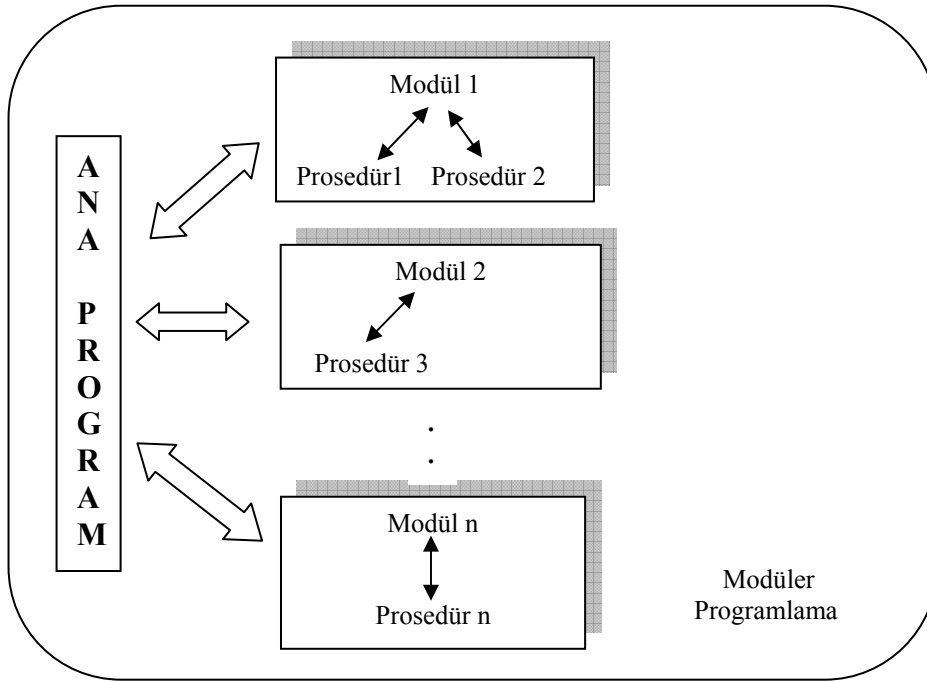
Yordamsal programlamanın en büyük sınırlılığı tekrar kullanılabilir kod sayısının az olmasıdır. Bu durum kodlama çok fazla tekrara yol açmakta ve program esnekliğini kısıtlamaktadır. Yordamsal programlamanın dezavantajları aşağıdaki şekilde

sayılabilir [147]:

1. Genel kullanıma açılan veriler tümüyle korumasız kalır.
2. Verinin kullanım amacı veri üzerinde yapılabilecek işlemleri hiçbir biçimde sınırlamaz.
3. Amaç dışı kullanımdan kaynaklanan yanlışlıklar ortaya çıktıktan sonra, yanlışlığa neden olan program kesiminin saptanması zordur. Bunun için söz konusu veriye erişen tüm yordamların tek tek incelenmesi gerekir.
Genel amaçlı parametrik programların tasarımı, gerçekleştirilmesi ve sınanması daha zor, pahalı ve zaman alıcıdır. Bunların anlaşılması ve bakımı da zordur. Hataların saptanması ve düzeltilmesi zordur.
4. Program, kullanıcı gereksinimlerini ve yenilik isteklerini tam karşılayacak hızda değiştirilemez.
5. Yapılan ek ya da değişiklikler, programın daha önce çalışan ve değiştirilmeyen kesimlerinde bile hataların oluşmasına yol açabilir.
6. Mevcut program kodlarında ciddi değişiklikler yapılmaksızın yeni ihtiyaçların karşılanması güçtür.

5.1.4. Modüler programlama

Modüler programlamada, genel fonksiyonel işlemler, ayrı modüllerin içinde birlikte gruplanmaktadır. Bu yapıda, bir program tek bir parçadan oluşmamaktadır. Program, prosedürlerle oluşturulan etkileşimli birkaç küçük program şeklinde oluşturulur. Ana program, modülleri koordine ederek farklı modüllerdeki prosedürleri çağırarak işlem yapar (Şekil 5.3).



Şekil 5.3. Modüler programlamada ana program – modül ilişkisi

Modül, birbiriyle ilgili işleri yapan bir fonksiyon topluluğudur. Modüler programlamada her bir modül kendi verisine sahiptir. Bir modül, bir bütünün içinde diğer elemanlardan bağımsız çalışabilen bir parça olarak tanımlanabilir. Modüllerin en önemli özelliği olabildiğince bağımsız ve kendi içinde yeterli olmalarıdır.

Ayrıca, belirli ve iyi tanımlanmış bir işlevi yerine getirmek üzere tanımlanmışlardır ve tek bir giriş ve tek bir çıkışları bulunur. Küçük ve kendi içinde yeterli birimlerin kullanılması, içlerindeki mantığın hem geliştiren hem de kullanan açısından daha kolay tasarlanmasını ve anlaşılmasını sağlar.

Modüler programları geliştirme kolaydır, çünkü her modül bağımsız olarak mükemmelleştirilebilir. Modüler tasarım aynı zamanda bir programın test edilmesini ve hatalarının bulunmasını da kolaylaştırır, hatalar kolay ayıklanabilir.

Ek işlevler yapabilecek yeni modüller geliştirilerek varolan tutarlı ve düzenli yapıya kolayca dahil edilebilir. Modüler programlamayı kullanan diller arasında Fortran 90

veya C sayılabilir. Bunun yanında nesne tabanlı programlar da modüler programlama yaklaşımını kullanmaktadır.

5.2. Sonuç

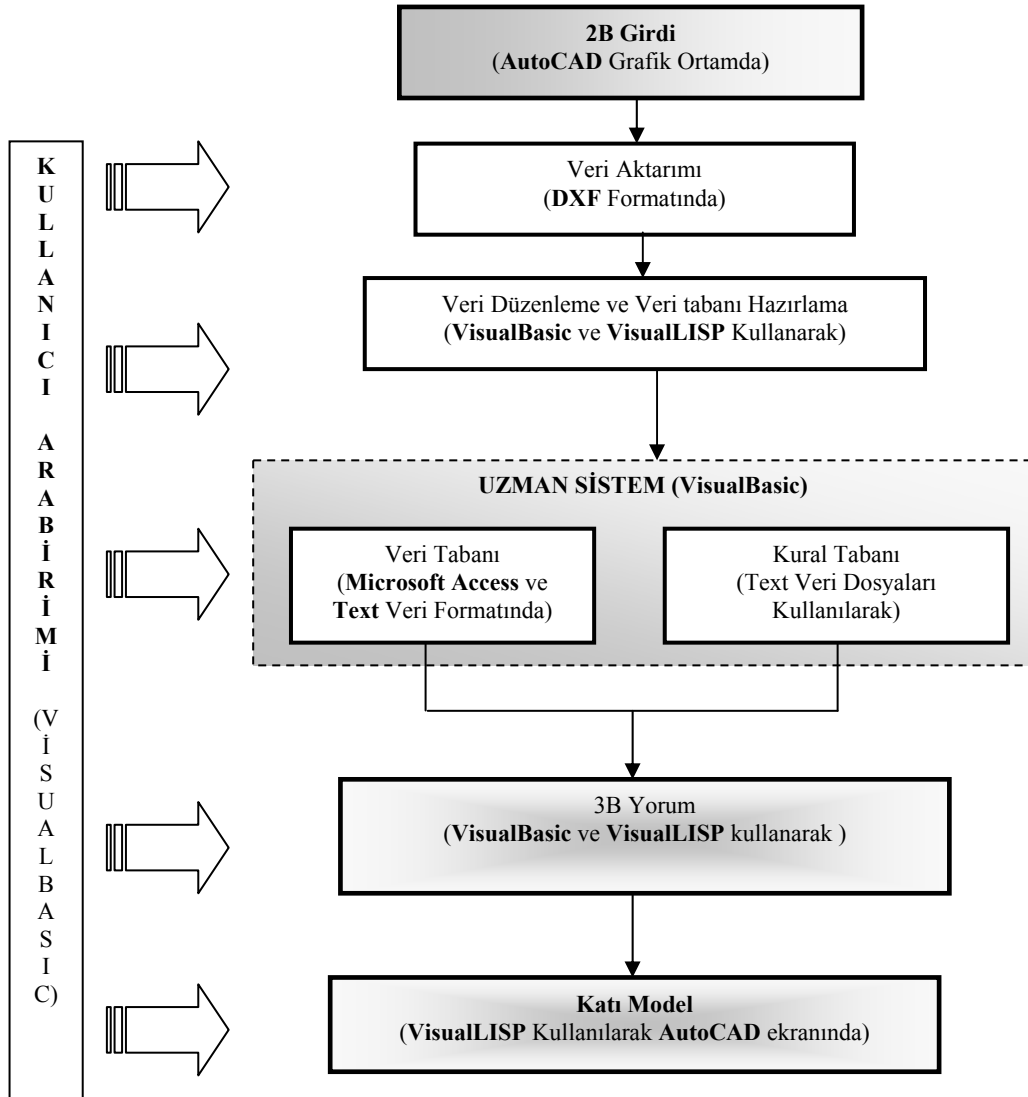
Yürütülen tez çalışması kapsamında geliştirilen bilgisayar programında nesne tabanlı programlama tekniğini kullanan Visual BASIC 6.0 ve fonksiyonel programlama dillerinden VisualLISP kullanılmıştır.

VisualBASIC'in güçlü nesne tabanlı yapısı ile veri tabanı, kullanıcı ara birimi oluşturma, AutoCAD ile kullanıcı arayüzü arasındaki iletişimi sağlama ve VisualLISP programlarını çalıştırabilmek için AutoCAD komut satırına komut gönderme işlemleri gerçekleştirilmektedir.

VisualBASIC ile, nesne tabanlı programlama tekniklerini kullanabilmenin yanında, modüler programlamaya müsait bir yapıya sahip olması, AutoCAD için kod yazılabilmesi tez çalışmasında tercih edilmesinin bir diğer sebebidir.

Fonksiyonel programlama dillerinden olan VisualLISP karmaşık matematiksel işlem ve karşılaştırmalarda kullanılmıştır. VisualLISP, AutoCAD ortamında yapılan çizim ve katı model oluşturma işlemlerinde AutoCAD komutlarını esnek bir şekilde kullanabilmesi nedeniyle tez çalışmasında tercih edilmiştir.

Hazırlanan bilgisayar programında grafik ortam olarak AutoCAD kullanılmıştır. Veri tabanı oluşturmada Microsoft Access programı ve text veri tabanı kullanılmıştır. Çalışma kapsamında hazırlanan uzman sistem VisualBASIC programlama dilinde hazırlanmıştır. Veri tabanı olarak Microsoft Access ve text veri tabanları kullanılmıştır. Şekil 5.4, hazırlanan bilgisayar programında kullanılan yazılım ve paket programları, kullanıldıkları adımları da belirtir bir şekilde görülmektedir.



Şekil 5.4 Geliştirilen algorithmada kullanılan programlar

6. ARAŞTIRMANIN KAPSAMI

6.1. Giriş

Tez çalışması kapsamında tek, iki ve üç görünüşle ifade edilen prizmatik ve silindirik parçaların katı modellerini oluşturan bir bilgisayar programı hazırlanmıştır. Program iki ana modülden oluşmaktadır. Bunlar:

- 1) Dönel simetrik parçaların ele alındığı modül,
- 2) İki ve üç görünüşle ifade edilen silindirik ve prizmatik parçalarla ilgili modül.

Hazırlanan programda katı model oluşturma işleminde hacimsel bir yaklaşım benimsenmiştir. Yaklaşım kapalı yüzeylerin doğrusal (düzlemsel) ve eksenel (dönel simetrik parçalarda) süpürme operasyonları ile oluşturulan temel katı elemanlardan boolean operasyonları ile katı model oluşturmaktadır. Bunun yanında çalışmada YKG ilkeleri de kullanılmaktadır.

Yukarıda verilen program modüllerinin veri tabanı işlemleri; kayıt ve 2B'lu temel elemanlarla ilgili veri tabanı işlemleri ortaktır. Modüller arası geçiş hazırlanan arabirim vasıtasıyla kullanıcı yönlendirilerek gerçekleştirilmektedir.

Programda CAD sistemlerinde ortak veri standardı olarak yaygın bir şekilde kullanılan DXF (Drawing Exchange Format) kullanılmıştır. Bu çalışmada DXF (Drawing Exchange File) veri standardının tercih edilme sebebi, 2B'lu verilerle ilgili işlemlerde önemli üstünlüklere sahip olmasıdır.

Geliştirilen bilgisayar yazılımı, Visual Basic 6.0 ve VisualLISP programlama dilleri kullanılarak hazırlanmıştır.

3B temel elemanların türüne ve yapılacak boolean operasyonlarına karar verme aşamasında, yapay zeka tekniklerinden uzman sistem kullanılmıştır.

Araştırma kapsamında geliştirilen programla AutoCAD ekranında yapılan serbest teknik resim çizimlerinin yanında, standart makine unsurlarının (delik ve çıkıntılar) katı modelleri başarılı bir şekilde oluşturulabilmektedir.

6.2. Amaç

Tasarım ve imalatla insan faktörü, kaliteli ürün ve imalat maliyetine doğrudan etki etmektedir. Yapılan tüm çalışmalar tasarım veya imalat sürecinde çalışanların bilgi, beceri ve yeteneğini arttırmaya yöneliktir. Tasarımı gerçekleştiren kullanıcının mesleki bilgisi ve tasarımı gerçekleştirdiği araçları kullanmadaki kabiliyeti parçanın doğru bir şekilde ve en kısa zamanda tasarımı ile doğrudan ilişkilidir. Böyle bir tasarım, şüphesiz bilgisayar desteği ile mümkün olmaktadır.

Doğru ve hızlı tasarımı gerçekleştirmek için kullanıcı yerine 2B'lu teknik resim girdilerini yorumlayan, katı model oluşturmada kullanılan grafik ortamın yeteneklerine sahip bir bilgisayar programı şu avantajlara sahip olabilir:

- Kullanıcı yorumundan kaynaklanan hatalar ortadan kalkabilir,
- Katı model oluşturma zamanı en aza indirilebilir,
- Katı modelle ilgili verilerin bir veri tabanında saklanması mümkün olabilir.

Burada yapılan çalışmanın amacı, katı model oluşturma sürecine yukarıda sayılan avantajları sağlayacak bir yaklaşım geliştirmektir.

6.3. Kapsam

Bu tezde tek, iki ve üç görünüşle ifade edilen silindirik ve prizmatik parçaların katı modellerini oluşturmaya yönelik bir çalışma tanıtılmaktadır. Çalışmada öncelikle kapsamlı bir kaynak araştırması verilmektedir. Konuyla ilgili yaklaşık kırk yıllık bir zaman diliminde yapılan çalışmalar tanıtılmakta ve bu çalışmalarla ilgili bir

değerlendirmede bulunmaktadır. Çalışmanın bu bölümünde sınıflandırma yöntemine yeni bir bakış açısı getirilmiştir.

Araştırmanın uygulamaya yönelik kısmı ise AutoCAD altında çalışan bir bilgisayar programının hazırlanmasıdır. Burada, AutoCAD altında çizilen 2B teknik resim girdilerini yorumlayıp katı modellerini oluşturan bir bilgisayar programı hazırlanmıştır. Hazırlanan bilgisayar programı aşağıdaki özelliklere sahiptir:

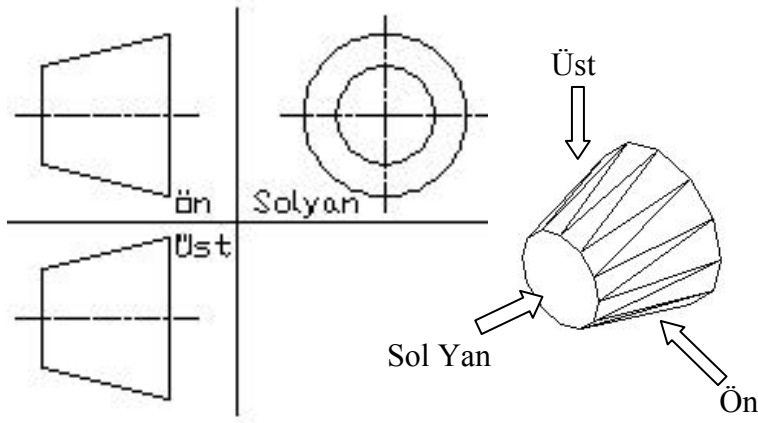
- Hazırlanan program, tek görünüşle ifade edilen dönel simetrik, iki ve üç görünüşle ifade edilen silindirik ve prizmatik parçaların katı modellerini oluşturmaktadır,
- Katı model oluşturma işlemi Hacimsel bir yaklaşımla gerçekleştirilmektedir.
- Katı model oluşumunda kullanılan temel elemanlar hacme dayalı teknikler kullanılmaktadır. Bu çalışmada temel elemanlar; klasik YKG yaklaşımı ve süpürme teknikleri ile oluşturulmaktadır. Son katı ise hacimsel kesitirme ile oluşturulmaktadır. Çalışma, hacim tabanlı yöntemlerin üçünü bir arada kullanmasıyla, probleme yeni bir bakış açısı getirmektedir.
- Katı temel elemanın biçimine ve buna bağlı olarak uygulanacak boolean operasyonlarına karar verme aşamasında yapay zeka tekniklerinden uzman sistem kullanılmaktadır.
- Çalışmada, 2B girdi, iki ve üç görünüşlü parçaları ait oldukları görünüş katmanlarına atma ve uzman sistem kural tabanına kural ekleme ile ilgili kısımları dışındaki bütün aşamalar kullanıcı katkısız gerçekleştirilmektedir.
- Çalışma, tek kalınlıklı parçaların tam ve yarım kesitlerini yorumlayıp katı modellerini oluşturabilmektedir.

6.3.1. Girdi çeşitleri

Katı model, 2B izdüşümlerin yorumlanması ile oluşturulmaktadır. Bu nedenle girdinin yapılış tarzı önemlidir. Çalışmada, 2B girdinin aşağıdaki tarz ve özellikte yapılması istenmektedir.

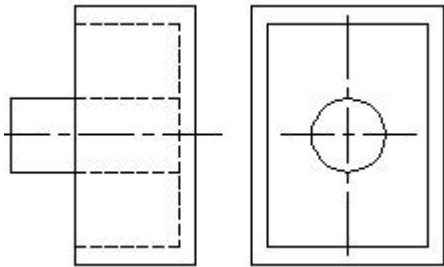
- a. Tek görünüşle ifade edilen dönel simetrik elemanların katı modelini oluşturabilmek için girdinin aşağıdaki özellikleri taşıması gerekmektedir:
 1. Program, girdinin doğru olduğunu kabul ederek işlem yapmaktadır. Hazırlanan programın hatalı çizimleri düzeltme fonksiyonu olmadığı için çizimin teknik resim kurallarına uygun ve doğru bir şekilde yapılması gerekmektedir.
 2. Programın bu modülü, simetrik silindirik parçalara hitap etmektedir. Bu yüzden program tek görünüşle ifade edilip kayıt işlemi gerçekleştirilen izdüşümü silindirik parça gibi işleme alınır.
 3. Bu modülde kullanılacak girdi line, pline ve arc'lerden oluşturulmalıdır. Parça eksenini mutlak surette belirtilmelidir.
 4. AutoCAD ortamında yapılan girdide çizgi türü olarak görünmez kenarlarda *"Hidden"*, görünür kenarlarda *"Bylayer"* veya *"Byblock"*, eksenlerde *"Center"* çizgi türleri kullanılmalıdır.
 5. Program, standart çizim kuralları çerçevesinde oluşturulan tam ve yarım kesitli parçaların katı modellerini de oluşturabilmektedir. Kesit tarama yöntemi olarak makinecilik alanında kullanılan tarama yöntem ve standartları tercih edilmelidir. Burada AutoCAD'de mekanik çizimlerin taranmasında kullanılan ANSI31 tarama biçimi referans alınmıştır.
- b. İki ve üç görünüşle ifade edilen silindirik ve prizmatik elemanların katı modelini oluşturabilmek için girdinin aşağıda belirtilen özelliklere uygun oluşturulması gerekmektedir:
 1. Dönel simetrik elemanlarda olduğu gibi 2B girdinin doğru ve teknik resim kurallarına uygun olması gerekmektedir.
 2. Girdi görünüşleri TS 88'de tasrif edilen kurallara uygun, birinci izdüşüm (Avrupa) yöntemine göre oluşturulmalıdır (Şekil 6.1) [148].
 3. 2B izdüşüm line, pline, arc ve circle çizim elemanları ile oluşturulmalıdır.
 4. Program, görünür veya görünmez (görünüşe göre arka tarafta kalmış) bölgeleri komşu izdüşümlerine göre yorumlayarak katı temel elemanları

oluşturabilmektedir. Bu yüzden prizmatik ve silindirik bölgelerin izdüşümleri tam uyumlu olmalıdır.



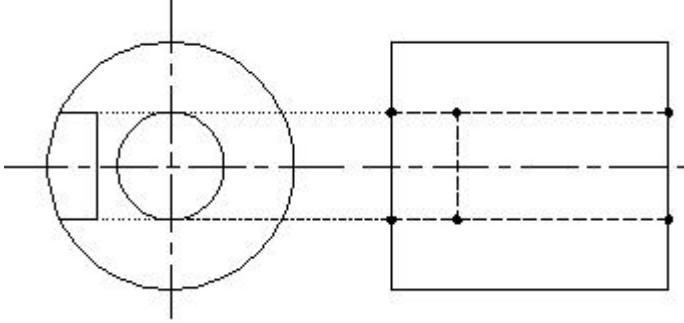
Şekil 6.1. Birinci izdüşüm yöntemi

5. Program tek kalınlıklı nesnelerin katı modellerini büyük oranda oluşturmaktadır. Çok parçalı resimlerde ise orta karmaşıklık düzeyindeki parçalarda başarılı sonuçlar vermektedir.
6. Program üst üste gelen görünüşleri (kapalı bölgeler) yorumlayabilmektedir.
7. İzdüşümü oluşturmada kullanılan 2B elemanın çizgi türü tek olmalıdır. Yani, bir çizgi görünür veya görünmez olmalıdır. 2B elemanın bir bölümü görünür, bir bölümü görünmez olmamalıdır (Şekil 6.2). Yani bir unsur, bir parçanın içinden başlayıp dışına taşmamalıdır. Böyle bir durumda, temel elemanın boyu yanlış hesaplanır.



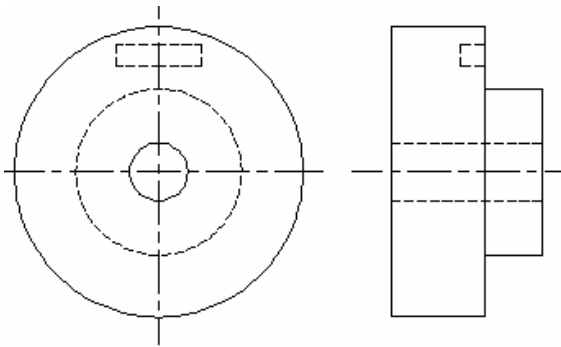
Şekil 6.2. Bir doğru parçasının iki farklı çizgi türüne sahip olma durumu

8. Algoritma son noktaları çakışan izdüşümleri yanlış yorumlayabilir. Bu tür durumlar katı temel elemanın boylarının yanlış hesaplanmasına yol açmaktadır. Bu yüzden bu tür parçalardan kaçınılmalıdır (Şekil 6.3).



Şekil 6.3. İzdüşümlerin çakışması

9. Bazı temel elemanlar bakış doğrultusuna göre arka tarafta kalabilirler. Bu tür parçalar 2B izdüşüm görünüşlerinde görünmez çizgilerle ifade edilir. Çalışmada bu tür parçaların da katı modelleri oluşturulabilmektedir (Şekil 6.4).

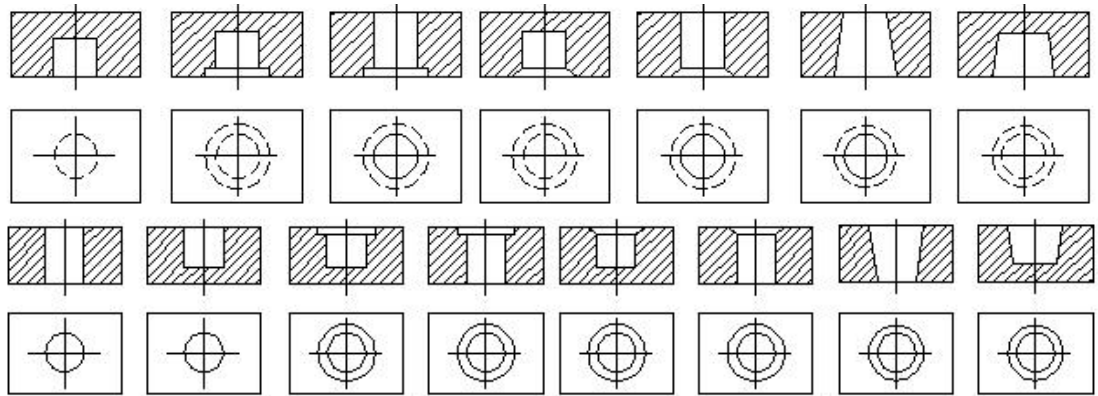


Şekil 6.4. Ana parçanın arkasında kalan temel elemanlar

Program, yukarıda anlatılan sınırlar dahilinde oluşturulan silindirik ve prizmatik parçaların katı modellerini oluşturmaktadır.

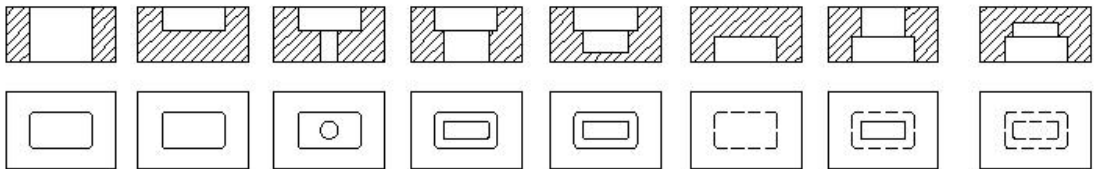
6.3.2. Çalışmada ele alınan parça türleri

Yapılan çalışmada delik ve çıkıntı unsurları oluşturulabilmektedir. Silindirik unsurlara boydan boya delik, kör delik, havşabaşlı delik, kademeli delik, düz silindirik çıkıntı konik çıkıntılar örnek verilebilir (Şekil 6.5).



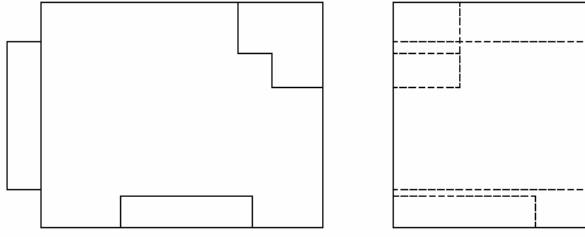
Şekil 6.5. Algoritmada ele alınan çeşitli silindirik delik unsurları

Geliştirilen katı model oluşturma algoritması ile kör, boydan boya, kademeli ve konik prizmatik delik unsurları oluşturulabilmektedir (Şekil 6.6).



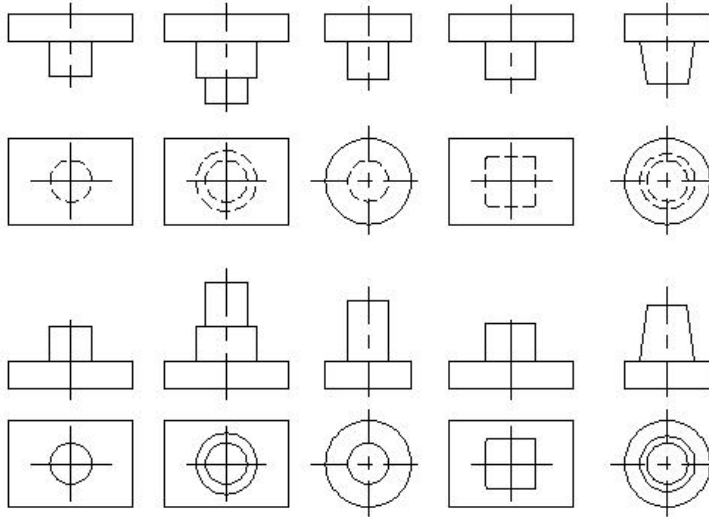
Şekil 6.6. Algoritmada ele alınan çeşitli prizmatik delik unsurları

Katı model oluşturma çalışmalarında en büyük problem, köşe ve kenar cep, kanal ve çıkıntılar gibi 2B izdüşümleri kapalı bir bölge oluşturmayan prizmatik temel elemanların oluşturulmasında ortaya çıkmaktadır (Şekil 6.7). Araştırma kapsamında geliştirilen Açık Kenar Kapatma Algoritması (AKKA) ile bu tür parçalarda büyük başarı sağlanmaktadır.



Şekil 6.7. Köşe, kenar cepler ve çıkıntılar

Araştırma kapsamında geliştirilen algoritma ile prizmatik ve silindirik çıkıntı unsurları da başarılı bir şekilde oluşturulabilmektedir (Şekil 6.8).



Şekil 6.8. Algoritmada ele alınan prizmatik ve silindirik çıkıntılar

7. 2B GİRDİ VE VERİ TABANI OLUŞTURMA

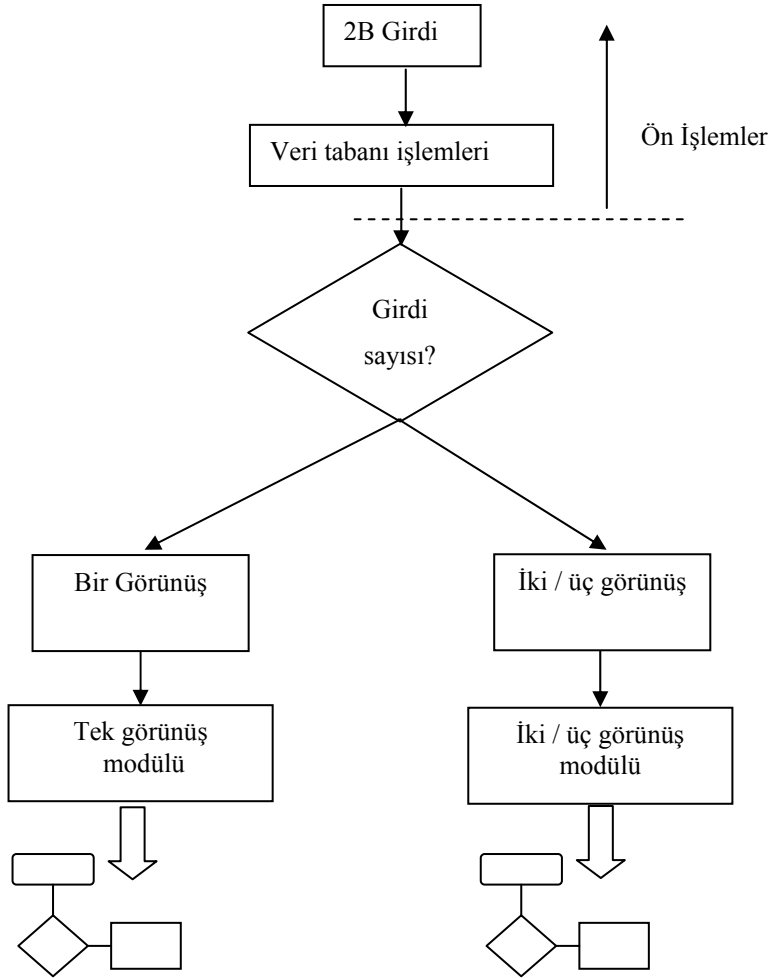
7.1. Giriş

Araştırma kapsamında silindirik ve prizmatik nesnelerin 2B'lu görüşleri kullanılarak katı modelleri oluşturulmaktadır. 2B'lu girdilerin katı modellerini oluşturmak için geliştirilen yaklaşımda nesnelerin bir, iki veya üç görüş olarak tarif edilen girdileri yorumlanarak 3B nesne yorumları elde edilmekte ve katı model oluşturulmaktadır. Çalışma kapsamında geliştirilen yöntem bir bilgisayar programında test edilmiştir. Bu program, girdinin ifade edildiği görüş sayılarına bağlı olarak şu iki ana modülden oluşur;

- 3) Tek görüşlü parça katı model oluşturma modülü ve,
- 4) İki veya üç görüşlü parça katı model oluşturma modülü.

Bu program modüllerin katı model oluşturma algoritmaları farklılık göstermektedir. Fakat, 2B girdi oluşturma, kayıt ve 2B'lu temel elemanlarla ilgili veri tabanı işlemleri ortaktır. Bu modüllerin seçim ve kullanımı, program içinde bulunan kullanıcı arabirimi aracılığı ile yapılmaktadır.

Ön işlemler olarak adlandırılan 2B girdi ve veri tabanı işlemleri sonrası girdi türüne göre ilgili program modülü çalıştırılmaktadır. Şekil 7.1'de programın ilgili modüllere nasıl yönlendirildiği şematik olarak gösterilmektedir.



Şekil 7.1. Program modüllerine geçiş

7.2. 2B Girdi

2B girdi AutoCAD ortamında yapılmaktadır. Girdiler, AutoCAD 2B temel çizim elemanları olan line, arc, pline ve circle kullanılarak yapılmaktadır. Katı modeli oluşturulacak girdi, dik izdüşüm metodu, görünüş düzenleri TS 88’de belirtilen birinci izdüşüm metodu kurallarına uygun şekilde yapılmaktadır [147].

Kullanıcı, girdiyi kapsam bölümünde anlatılan sınırlandırmalar çerçevesinde AutoCAD ortamında oluşturmaktadır.

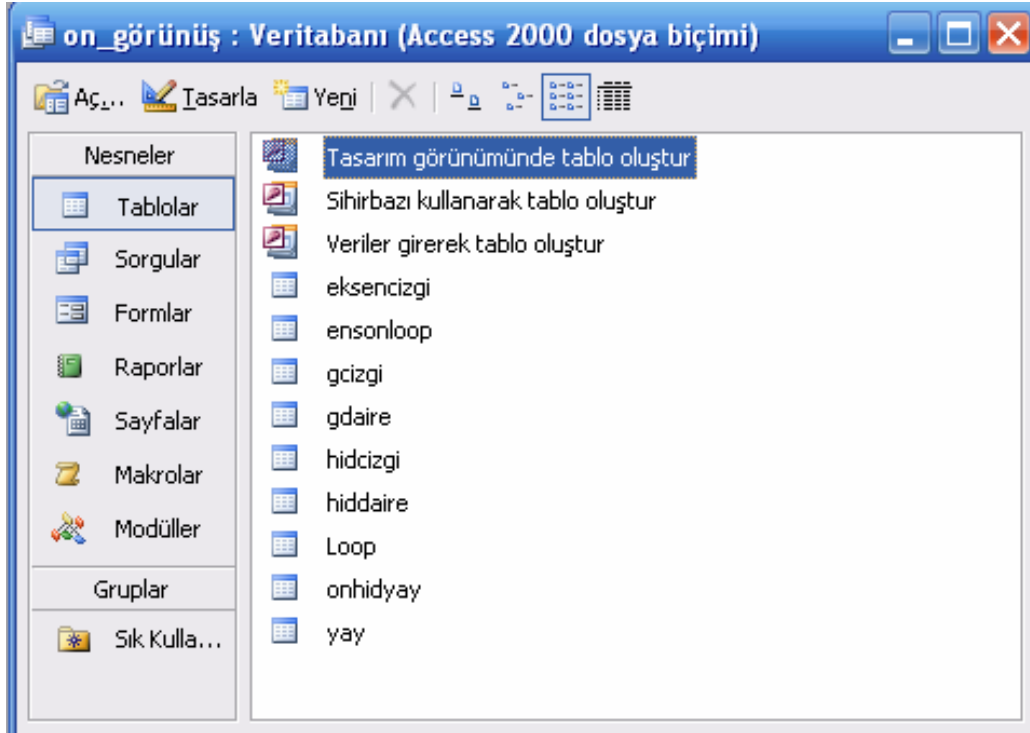
7.3. Veri Tabanı Oluşturma

Hazırlanan bilgisayar programında veri tabanı oluşturma, güncelleme gibi işlemler katı model oluşturma işleminin tüm adımlarında gerçekleştirilir. Katı model oluşturma işlemi boyunca etkili bir veri tabanı işlemi söz konusudur. Bu bölümde 2B girdi ile ilgili veri tabanı oluşturulur. Burada, DXF veri formatına kaydedilen 2B izdüşümle ilgili veriler düzenlenerek katman, eleman türü ve çizgi türüne göre veri tabanına aktarılır.

Veri tabanı oluşturma işleminde iki tür veri tabanı formatı kullanılmıştır. Bunlar, Microsoft Access ve text veri tabanlarıdır. AutoCAD ortamında yapılan işlemler için kullanılan programlama dili olan VisualLISP, Microsoft Access veri tabanındaki verileri yorumlayamamaktadır. Bu sebeple çizimle ilgili son veriler *text* veri tabanına aktarıldıktan sonra VisualLISP ile yapılan işlemlerde kullanılmaktadır. Bunun yanında, Access veri tabanı daha düzenli ve kolay işlenebilir bir formatta olduğu için veri düzenleme ve görüntüleme işlemlerinde tercih edilmiştir.

Kayıt işleminden sonra DXF veri tabanına kaydedilen bilgiler sınıflandırılmakta ve Microsoft Access ortamında hazırlanan veri tabanına aktarılmaktadır. Sınıflandırma 2B'lu veri türü (doğru, yay ve çember), verilerin çizgi türü (görünür, görünmez ve eksen) ve katmanlarına (ön, yan, üst ve tek) göre yapılmaktadır.

Microsoft Access ortamında hazırlanan veri tabanında; her görünüş katmanı için ayrı veri tabanı oluşturulur (ön_görünüş.mdb, yan_görünüş.mdb, üst_görünüş.mdb ve tek_görünüş.mdb). Görünüşlerle ilgili veri tabanı altında ise çizgi türlerine göre (görünür, görünmez ve eksen) ayrılmış 2B elemanlara (doğru, yay ve çember) ait kayıtlar (gorunurcizgi, gorunmezçizgi, gorunurdaire, gorunmezdaire vb.) oluşturulur (Şekil 7.2).

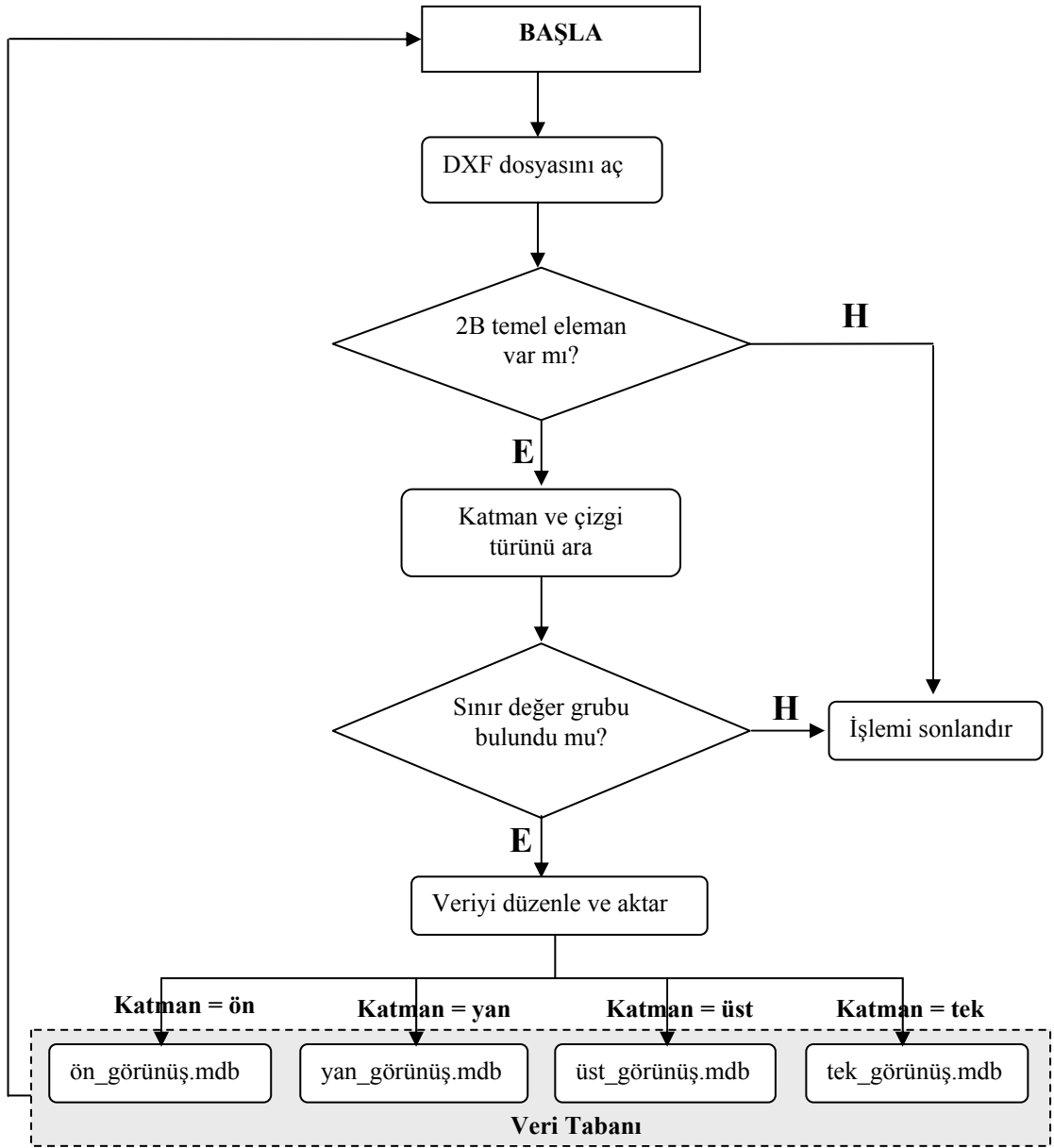


Şekil 7.2. Microsoft Access ortamında oluşturulan veri tabanı (ön görünüş için)

Text veri dosyalarına kayıt işleminde ise her bir katman için ayrı klasörler oluşturulur. Her bir görünüş için oluşturulan klasörlere doğru (görünür, görünmez, eksen), çember (görünür, görünmez) ve yay (görünür, görünmez) verileri aktarılır.

7.3.1. DXF'ten veri aktarma

DXF'teki verilerin veri tabanına aktarılmasında hiyerarşik bir yöntem takip edilmektedir. 2B izdüşümlerin var olduğu katmanlar sırayla araştırılarak ilgili veriler veri tabanına aktarılır (Şekil 7.3).



Şekil 7.3. Veri aktarım işlemi

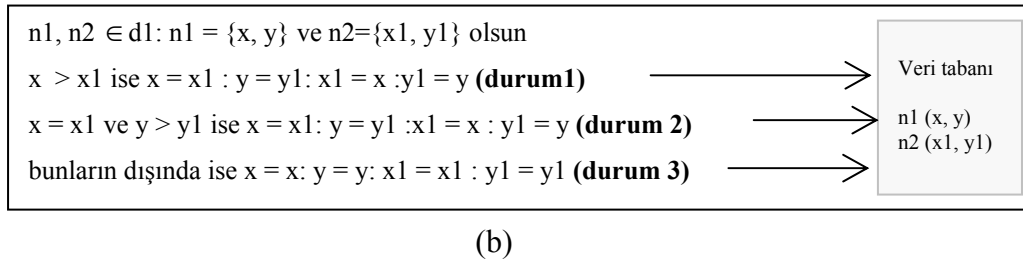
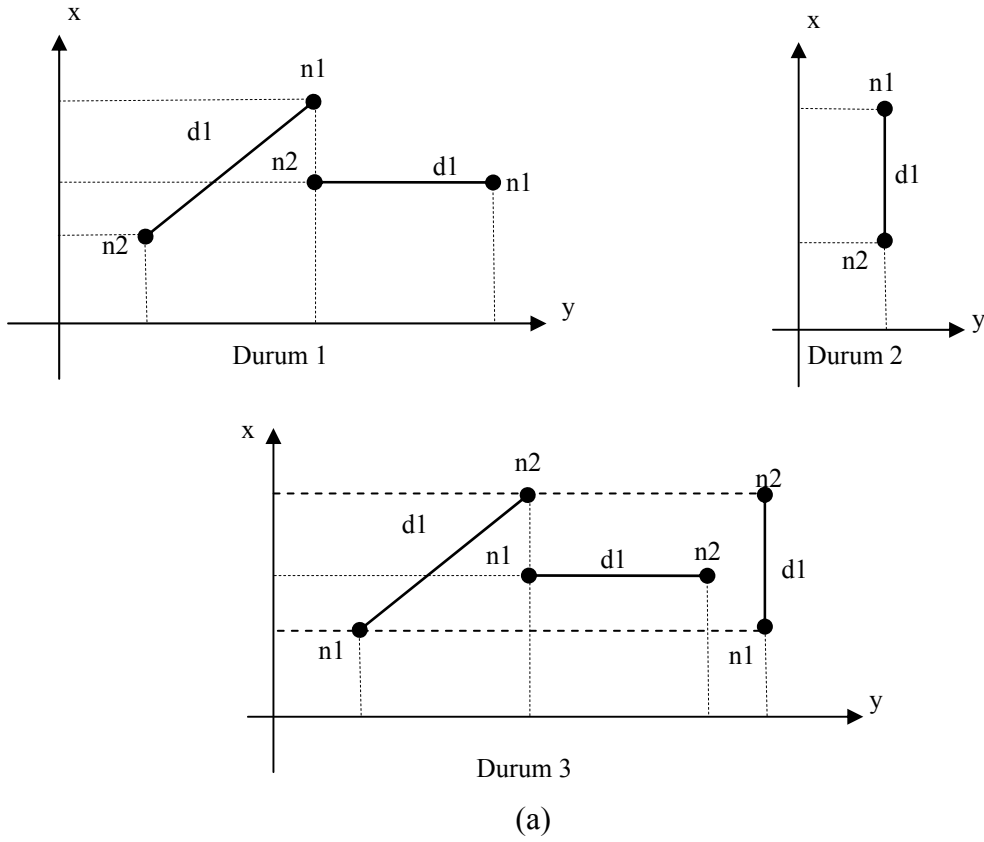
Arama işleminde önce 2B eleman türü araştırılır. Eğer veri tabanında 2B elemanlardan doğru, yay veya daire bulunursa bu elemanın katmanı araştırılır. Katman bulunduktan sonra, 2B elemanın çizgi türü tespit edilir. Çizgi türü tespit edilen elemanın son olarak sınır değer grupları aranmaktadır. Bu grupta, 2B'lu elemanın fiziksel biçimini tanımlayan koordinat verileri vardır. Bu veriler DXF'te x , y ve z koordinatları olarak saklanmaktadır.

DXF'teki 2B'lu elemanlarla ilgili veriler veri tabanına olduğu gibi aktarılmaz. Bunlar bazı düzenlemelere tabi tutulurlar. Bu işlem, doğruların çizim yönlerini değiştirme, daire ve yay tanımlarından yeni veri oluşturma şeklindedir. Veri tabanına aktarılan 2B temel elemanların özellikleri ve bunlarla ilgili yapılan düzenlemeler aşağıda verilmiştir.

Doğru: Doğru, iki noktanın sınırladığı bölgedir. Her bir doğru parçası düzlemsel yüzeyleri oluşturan kenarlardır. Bir doğru parçasının başlangıç ve bitiş noktası x ve y koordinat çiftleri ile ifade edilir. Buna göre 2B bir doğru parçası, x ve y koordinatlarından oluşan başlangıç ve bitiş köşe noktalarından oluşur. Bu köşe noktalarının koordinat değerleri veri tabanına aktarılır.

Aktarma sırasında, köşe noktaları soldan sağa veya aşağıdan yukarıya doğru sıralanır. Bu işlem için köşe sıralama algoritması geliştirilmiştir. Burada bir doğru için üç durum tanımlanmıştır (Şekil 7.4). Bir doğru ve yay parçasının köşe noktaları veri tabanına aşağıda verilen yöntemle aktarılır:

- 1) Doğruya ait başlangıç ve bitiş nokta koordinatlarından y değerleri eşitse ilk önce x koordinatı küçük olan köşe noktası, daha sonra diğer köşe noktası veri tabanına aktarılır (Durum 1).
- 2) Eğer x koordinatları eşitse önce y koordinatı küçük olan, daha sonra diğer köşe noktası aktarılır (Durum 2).
- 3) x veya y koordinatları eşit değilse önce x koordinatı küçük olan köşe noktası daha sonra diğer köşe noktası aktarılır (Durum 1 ve 3).



Şekil 7.4. Bir doğruya ait köşe noktalarının birbirlerine göre konumları:

- a) Köşe nokta konumları
 b) köşe sıralama kuralları

Köşe sıralama işlemi ile doğru ve yay parçalarının belirli bir düzende veri tabanında tutulması sağlanır. Böylece programın ileriki aşamalarındaki işlemler daha düzenli ve kolay yapılması sağlanmaktadır.

Daire: Daireler DXF veri formatında merkez x, y koordinatları ve yarıçap değerleri ile tanımlanırlar. Program içinde bu değerlere ilave olarak dairenin üst uç ve yan uç noktalarının koordinat değerleri de hesaplanır. Şekil 7.5’de görülen merkez ve r

değerlerinden oluşturulan $d1$ dairesinde;

$Merkez = \{x1, y1, z1\}$, $tp1 = \{tp1_x, tp1_y\}$: $tp2 = \{tp2_x, tp2_y\}$: $yn1 = \{yn1_x, yn1_y\}$:
 $yn2 = \{yn2_x, yn2_y\}$ ve $r = \text{yarıçap}$ olmak üzere üst uç ve yan uç nokta koordinatları

$$tp1_x = x1 \quad (7.1)$$

$$tp1_y = y1 + r \quad (7.2)$$

$$tp2_x = x1 \quad (7.3)$$

$$tp2_y = y1 - r \quad (7.4)$$

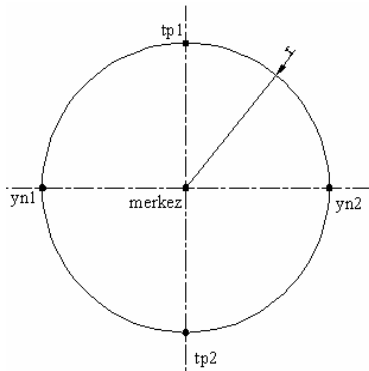
$$yn1_x = x1 - r \quad (7.5)$$

$$yn1_y = y1 \quad (7.6)$$

$$yn2_x = x1 + r \quad (7.7)$$

$$yn2_y = y1 \quad (7.8)$$

eşitlikleri ile hesaplanır.



Şekil 7.5. Daire ve ilgili veriler

Yay: Bir yay, DXF veri formatında merkez x, y koordinatları, yay yarıçapı, yay başlangıç ve bitiş açıları ile temsil edilir (Şekil 7.6). Ayrıca yay başlangıç ve bitiş nokta koordinatları hesaplanarak veri tabanına aktarılmaktadır. Bu değerler aşağıdaki şekilde hesaplanmaktadır.

Merkez = $\{x, y, z\}$ ve yarıçapı = r başlangıç açısı α ve bitiş açısı β olan bir yayın

başlangıç (n_1) ve bitiş nokta (n_2) koordinatları $n_1 = (x_1, y_1, z_1)$: $n_2 = (x_2, y_2, z_2)$: $z_1 = z_2 = 0$ olmak üzere:

n_1 başlangıç noktası:

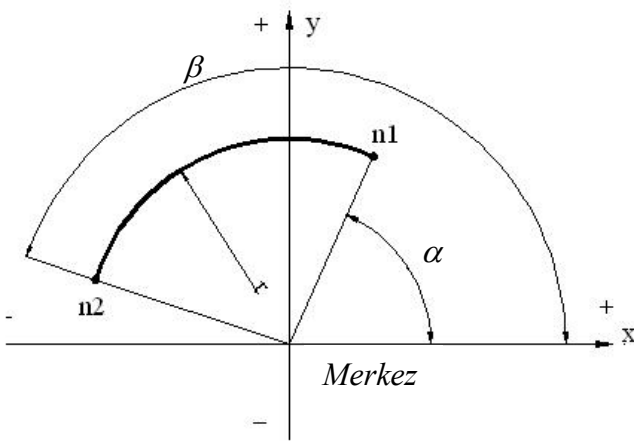
$$x_1 = x \pm (r \times \cos \alpha) \quad (7.9)$$

$$y_1 = y \pm (r \times \sin \alpha) \quad (7.10)$$

n_2 bitiş noktası:

$$x_1 = x \pm (r \times \cos \beta) \quad (7.11)$$

$$y_1 = y \pm (r \times \sin \beta) \quad (7.12)$$



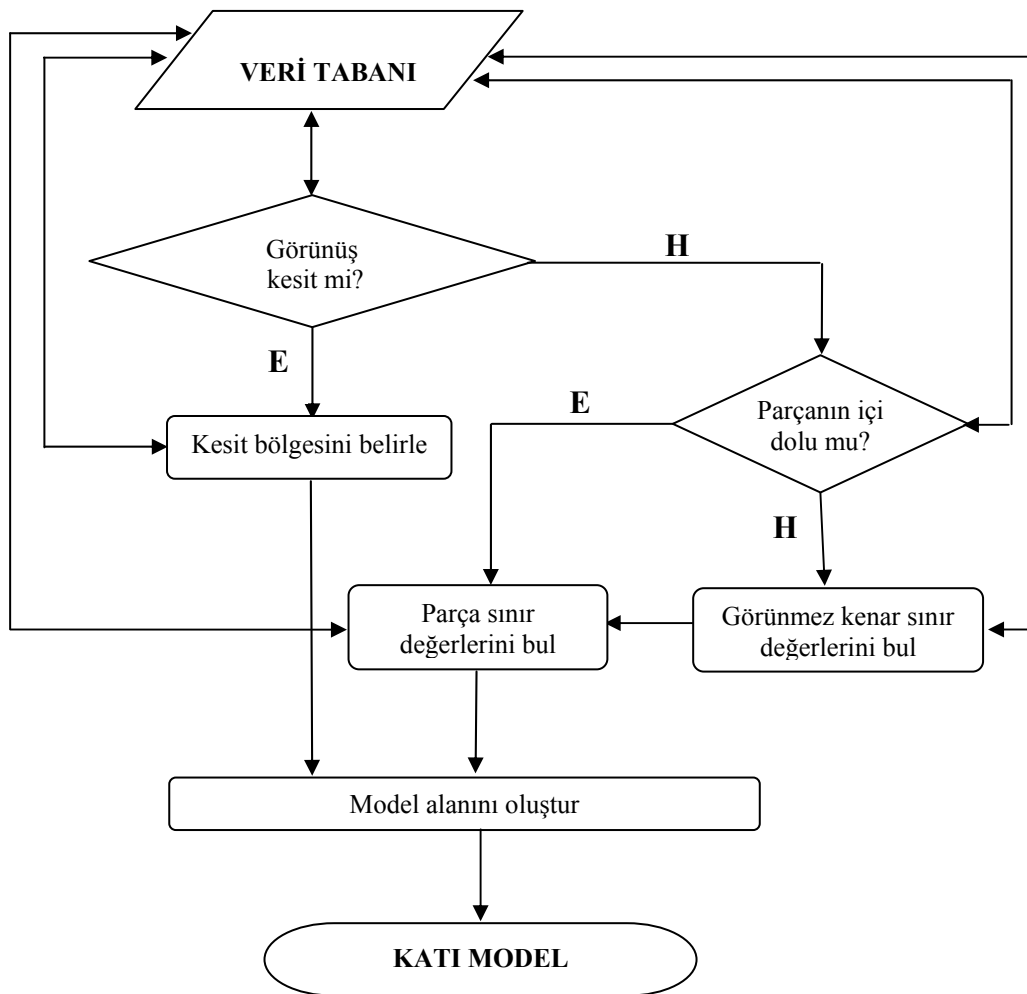
Şekil 7.6. Yay ve ilgili veriler

2B temel eleman verileri düzenleme işlemleri sonrası görünüş sayısına göre ilgili program modülüne geçiş yapılır. Geçiş, kullanıcı arabirimi aracılığı ile yapılır.

8. DÖNEL SİMETRİK PARÇALARIN KATI MODELLERİNİ OLUŞTURMA

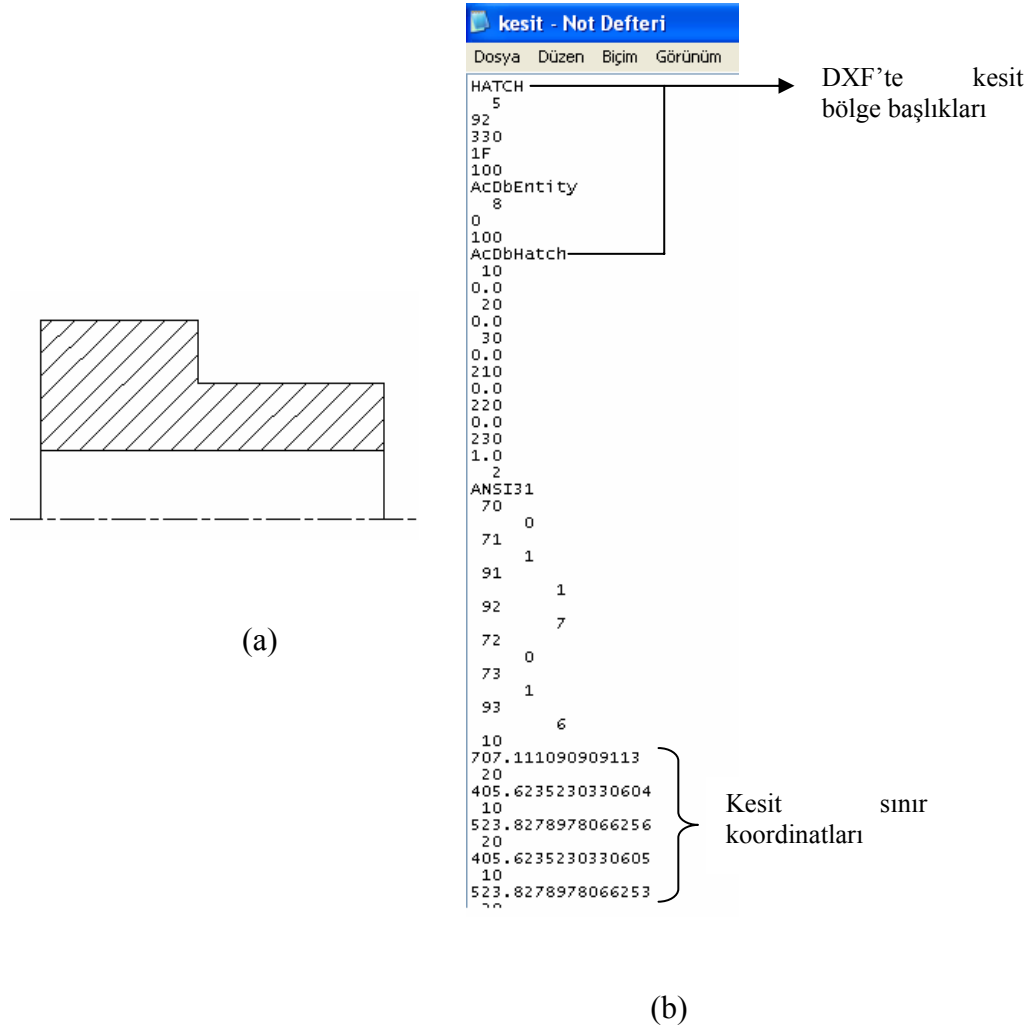
Dönel simetrik parçalar, genelde tek görünüşle ifade edilmektedir. Programın bu modülü kullanılarak tek görünüşle ifade edilebilen dönel simetrik parçaların katı modelleri oluşturulmaktadır. Burada girdinin simetrik olma özelliği dışında herhangi bir sınırlama yoktur.

Dönel simetrik parça oluşturma modülünde kesit görünüşlerle ifade edilen parçaların da katı modelleri oluşturulabilmektedir. Burada ana işlem adımları, 2B izdüşümde kesit bölgesi ve görünmez kenar araştırma, model alanı oluşturma ve eksenel döndürme ile katı model oluşturma şeklindedir (Şekil 8.1).



Şekil 8.1. Dönel simetrik parça oluşturma işlem akış şeması

Burada öncelikle parçanın kesit görünüşle ifade durumu araştırılmaktadır. Parçanın kesit olup olmadığı veri tabanı oluşturma aşamasında belirlenmektedir. Bir çizimin kesit içerip içermediği, kesit tarama türü ve kesit bölgesinin sınırları DXF veri tabanından elde edilmektedir. DXF veri tabanı, bir çizimin kesit olup olmadığı ile ilgili bilgiyi kaydeder. DXF’te kesit bölgeleri ile ilgili bilgiler AutoCAD kesit tarama komutu olan “Hatch” başlığı altında tutulmaktadır (Şekil 8.2). Bu bilgiler Hazırlanan program aracılığı ile veri aktarım işlemi sırasında, DXF ortamından alınıp veri tabanına aktarılmaktadır.

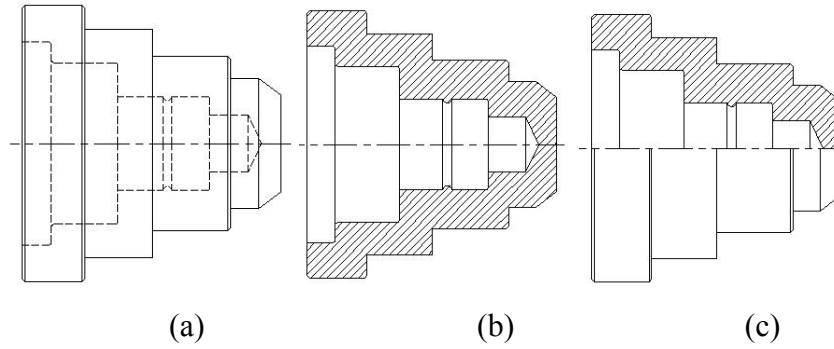


Şekil 8.2. Kesit alınmış bir parçanın ifadesi:

- Kesit alınmış parça
- DXF’te kesit bölge başlığı ve kesit sınır koordinatları

8.2. Kesit Alınmış Dönel Simetrik Parçaların Katı Modellerini Oluşturma

Kesit görünüşler teknik resimleri çizilen parçaların iç ayrıntılarının daha açık ifade edilebilmesi için kullanılır. Pek çok kesit türü vardır. Tam kesit, bir parçanın simetrisinden kesilmiş halini gösterir; yarım kesitler ise parça geometrisinin yarısının kesilmiş halidir (Şekil 8.3). Ayrıca kademeli ve kısmi kesitler de teknik resim çizimlerinde kullanılmaktadır. Araştırma kapsamında bu kesit türlerinden tam ve yarım kesitle ifade edilen parçalar incelenmiştir.



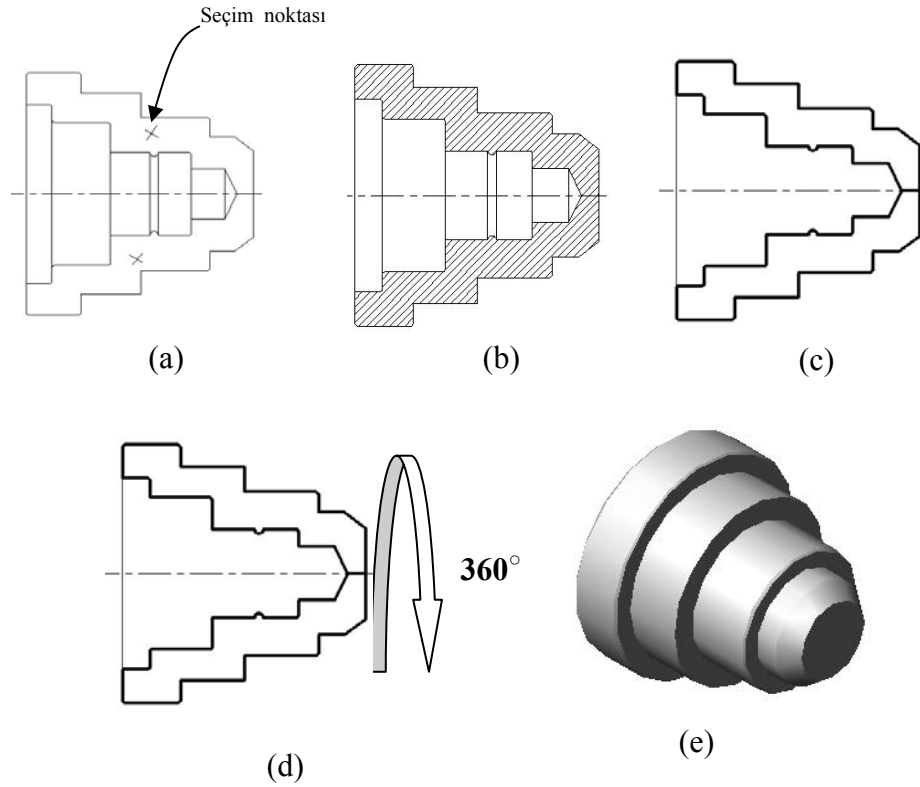
Şekil 8.3. 2B görünüşlerin tam ve yarım kesitle ifadesi:

- a) 2B görünüş
- b) tam kesit
- c) yarım kesit

Kesit görünüşlerle ifade edilen parçaların model alanının oluşturulmasında basit olmakla birlikte etkili bir yöntem kullanılmaktadır. Burada kesit parçaların oluşturulmasında kesit bölgesinin seçiminde kullanılan seçim noktasından (*pick point*) faydalanılmaktadır.

Seçim noktası yöntemi ile belirlenen kesit bölgesi aynı zamanda model alanıdır. Hazırlanan program ile DXF verilerinden kesit oluşturmak için kullanılan seçim nokta koordinatı bulunmaktadır. Daha sonra bu noktanın etrafını sınırlayan kapalı alan model alanına dönüştürülmektedir. Bu işlem, AutoCAD “*boundary*” komutu kullanılarak kolayca gerçekleştirilir. Komut, tespit edilen bir seçim noktası etrafındaki doğru ve yayları birleşik çizgi haline dönüştürerek model alanını oluşturur (Şekil 8.4. (c)). Daha sonra model alanı döndürme eksenini etrafında 360°

döndürülerek katı model oluşturulmaktadır (Şekil 8.4. (d)). Tam kesitlerde iki seçim noktası olduğu için aynı modelin iki kez oluşturulma durumu ortaya çıkabilecektir. Bu gibi işlem tekrarlarını önlemek için kesit parçalarda eksen üstündeki profil döndürülmektedir. Model alanının eksenel döndürme işlemi AutoCAD komutlarından “Revolve” komutuyla gerçekleştirilmektedir.



Şekil 8.4. Kesit alınmış dönel simetrik parçanın katı modelini oluşturma:

- a) Seçim noktası yöntemi ile kesit bölgesi tespiti
- b) tam kesiti alınmış parça
- c) model alanı
- d) eksenel döndürme
- e) katı model

Bu aşamada son işlem, ekrandaki 2B’lu temel elemanların temizlenmesidir. Bunun için katı model yeni bir katmana atılır. Bu katman kapatılarak ekrandaki bütün 2B temel elemanlar temizlenir. Daha sonra kapalı 3B katman açılarak katı model ekranda görüntülenir. Yeni katman oluşturma, katmanı açma ve kapatma işlemler

AutoCAD “layer” komut fonksiyonlarının (thaw, freze, new vb.) program içerisinde kullanılmasıyla gerçekleştirilir.

AutoCAD komutları, VisualLISP ortamında hazırlanan program parçasıyla otomatik olarak çalıştırılmaktadır.

Bu yöntem kullanılarak tam ve yarım kesitle ifade edilen dönel simetrik parçaların katı modelleri oluşturulmaktadır. Yöntem, 2B izdüşümün oluşturulması dışında kullanıcı katkısı içermemektedir.

8.2. Kesit Alınmamış Parça Katı Modellerini Oluşturma

Kesit alınmadan çizilen dönel silindirik parçaların katı modellerini oluşturma, kesit parçalara göre farklı adımlar içermektedir. Burada önce 2B veriler düzenlenmekte, daha sonra model alanı oluşturulmaktadır. Katı model kesit görünüşlerle ifade edilen parçalarda olduğu gibi model alanının dönme eksenini boyunca döndürülmesiyle oluşturulur. Bu modülün ana işlem adımları şu şekilde belirtilebilir:

- a) Parçanın iç ayrıntı durumu saptanır (dolu veya boş parça),
- b) Döndürme eksenini altındaki kenar silinir,
- c) Eksen üstündeki doğru ve yay parçaları düzenlenir (sıralanır),
- d) Eksik çizgiler tamamlanır,
- e) Model alanı oluşturulur,
- f) Eksenel döndürme ile katı model oluşturulur.

8.2.1. Parçanın iç ayrıntı durumunu saptama

Dönel simetrik bir parçanın iç ayrıntısı görünmez çizgilerle ifade edilen delik unsurlarıdır. Kesit görünüşlerle ifade edilmeyen bir 2B’li izdüşüm eğer görünmez kenarlara sahip değilse *içi dolu parça* olarak tanımlanır. Bu durum veri tabanı incelenerek tespit edilmektedir. İçi dolu ve boş parçalar aynı modül içerisinde

değerlendirilir. İçi dolu ve boş parça ayrımı işlem zamanını azaltma açısından önemlidir. Parçanın içi doluysa, görünmez kenar düzenlemeleri yapılmadan model alanı oluşturma işlemine geçilir (Şekil 8.1).

8.2.2. Döndürme eksenini altında kalan kenarları silme

Burada yapılan işlem, 2B izdüşümü oluşturan köşelerden eksenin üzerinde kalan köşe noktaları dışındaki köşelerin silinmesidir. Veri tabanında kayıtlı olan izdüşüm verileri incelenerek eksen üzerindeki köşe noktalarından yeni 2B nokta listesi oluşturulmaktadır. Eksen üzeri köşe nokta listesi hazırlama işlemi aşağıdaki şekilde ifade edilebilir.

Şekil 8.4’de d_e söndürme eksenini, k_I 2B izdüşümün herhangi bir kenarını ve m_k model alanını oluşturacak kenarların liste ismini temsil etmek üzere;

$$d_e = \{(n_e, n_{e+1}) \mid n_e = (x_e, y_e, z_e), n_{e+1} = (x_{e+1}, y_{e+1}, z_{e+1})\}$$

Eksen üzerinde n_1 ve n_2 köşe noktalarından oluşan bir k_I kenarı,

$$k_I = \{(n_1, n_2) \mid n_1 = (x_1, y_1, z_1), n_2 = (x_2, y_2, z_2) : x_1 < x_2, x_1 < x_2, z_1 = z_2 = 0\}$$

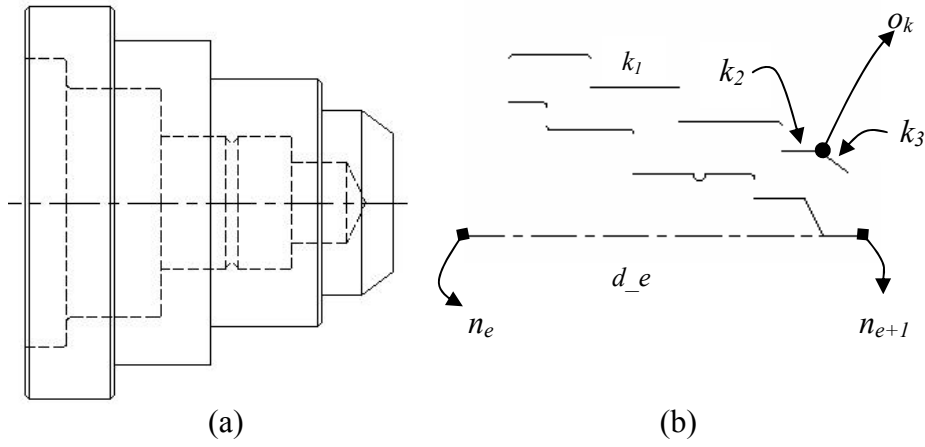
şeklinde tanımlanmış olsun,

Eğer $y_1 > y_e$ ve $y_2 > y_e$ veya $y_1 > y_{e+1}$ ve $y_2 > y_{e+1}$ ise $k_I \in m_k$.

Eğer $y_1 = y_{e+1}$ ve $y_2 > y_{e+1}$ veya $y_1 > y_{e+1}$ ve $y_2 = y_{e+1}$ ise $k_I \in m_k$.

olur.

Şekil 8.5.(a)’da görünüş ve Şekil 8.5.(b)’de eksen üzerinde kalan kenarlar görülmektedir. Görünmez kenarların eksen altında kalan kısımları da aynı işlemle silinir.



Şekil 8.5. Eksen altında kalan kenarları silme:

a) 2B izdüşüm

b) eksen üzerindeki nokta, kenar ve yay elemanları

8.2.3. Eksen üstündeki doğru ve yay köşe noktalarını düzenleme

Eksen üzerinde kalan kenar ve yaylara ait 2B köşe nokta listesi en küçük x koordinatına sahip köşe noktasından başlayarak bir zincir oluşturacak şekilde sıralanmaktadır. Sıralama işleminde, her bir kenarın son noktasına eşit köşe noktasına sahip kenar bulunarak yeni kenar listesine eklenir.

Şekil 8.5.(b)'de görünen k_2 ve k_3 kenarlarının x ve y koordinatlarının eşit olduğu (kesiştigi) nokta ortak köşe (o_k) noktalarıdır ve bu iki kenar birbirini izleyen iki kenar olarak yeni kenar listesine konur:

$$k_2 = \{(n_3, n_4) \mid n_3 = (x_3, y_3, z_3), n_4 = (x_4, y_4, z_4) : x_3 < x_4, z_3 = z_4 = 0\}$$

$$k_3 = \{(n_5, n_6) \mid n_5 = (x_5, y_5, z_5), n_6 = (x_6, y_6, z_6) : x_5 < x_6, z_5 = z_6 = 0\}$$

ve,

$x_2 = x_3$ ve $y_2 = y_3$ ise, k_2 ve k_3 kenarları birbirini izleyen iki kenardır.

Burada, eksen altında kalan kenarlar işleme alınmadığı için bazı durumlarda kenarların son noktasına eşit köşe noktası bulunmayacaktır. Bu durumda sadece bir

koordinatın (x veya y) eşit olması yeterlidir. Böylece birbiriyle ilişkili olmayan fakat birbirini izleyen kenarlar sıralanmış olacaktır (Şekil 8.5.(b)).

8.2.4. Eksik çizgileri tamamlama

Burada önemli bir işlem açık kalan kenarların kapatılmasıdır. Birbirini izleyen kenarlar arasında bağlantılı olmayan kenarlar vardır. Bu durumda model alanı kapalı bir bölge oluşturamaz. Model alanını kapalı bir bölge haline getirebilmek için, bu tür kenarlar arasında bağlantı kenarları oluşturulmaktadır. Bu tür kenarlar sanal kenar olarak adlandırılmaktadır. Sanal kenarlar, birbirleriyle komşu kenarların ortak koordinat değerlerinden oluşturulur. Bunlar iki türde incelenmektedir:

- 1) Parçanın eksen üzerinde kalan kenarları arasında ara sanal kenarlar (as_n^k)
- 2) Model alanının ilk ve son noktasını kapatan son sanal kenarlar (ss_n^k)

Ara sanal kenarların oluşturulması (as_n^k): Ara sanal kenarlar, eksen üzerindeki kenarlar arasında birbirleri ile bağlantılı olmayan kenarlar arasında bağlantı oluşturur (Şekil 8.6). Ara sanal kenar (as_n^k) birbirini izleyen ve sadece x koordinatları eşit olan iki kenar arasında oluşturulur. Kenarın köşe nokta koordinatları bu iki kenarın aynı x koordinatına sahip köşe noktalarının koordinatlarından oluşturulur. Şekil 8.6'daki k_1 ve k_2 kenarları arasında kesik çizgi ile çizilen ara sanal kenar aşağıdaki şekilde oluşturulur:

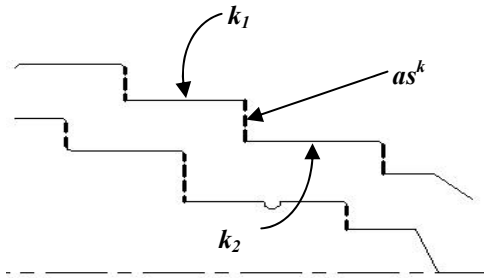
k_1 ve k_2 kenarları,

$$k_1 = \{(n_1, n_2) \mid n_1 = (x_1, y_1, z_1), n_2 = (x_2, y_2, z_2) : x_1 < x_2, z_1 = z_2 = 0\}$$

$$k_2 = \{(n_3, n_4) \mid n_3 = (x_3, y_3, z_3), n_4 = (x_4, y_4, z_4) : x_3 < x_4, z_3 = z_4 = 0\}$$

olarak tanımlı, her iki kenarın çizgi türü aynı ve $(as_n^k : n = 1 \dots n)$ 'e kadar kenar sayısını temsil etmek üzere;

$x_2 = x_3$ ise ve $y_2 \neq y_3$ ise $as_n^k = (n_2, n_3)$ olur (Şekil 8.6).



Şekil 8.6. Ara sanal kenarlar

Son sanal kenarların oluşturulması (ss_n^k): Son sanal kenarlar parçanın en küçük ve en büyük köşe noktalarının arasında oluşturulan kenarlardır. Bu kenarlar model alanını kapatan kenarlardır. İçi dolu ve boş (görünmez kenar içeren) parçaların son sanal kenarlarının oluşturulmasında farklı durumlar vardır.

a) *İçi dolu parçalarda son sanal kenarların oluşturulması:* En küçük ve en büyük köşe noktalarının koordinatları ile eksen koordinat değerlerinden oluşturulur. Burada ucu konik ve düz parçalar için iki durum tanımlanmıştır (Şekil 8.7).

$n_1 = (x_1, y_1, z_1)$ ve 2B görünüşün en küçük x koordinatına sahip köşe noktası olarak, $n_3 = (x_3, y_3, z_3)$ ve 2B görünüşün en büyük x koordinatına sahip köşe olarak, döndürme eksenini $d_e = \{ (x_e, y_e, z_e) (x_{e+1}, y_{e+1}, z_{e+1}) \}$, $z_1 = z_2 = z_e = z_{e+1} = 0$ olarak tanımlansın. ss_n^k ($n = 1 \dots n$)'e kadar kenar sayısını temsil etmek üzere

1. Durum: Düz parçalar için (Şekil 8.7.(a)):

$y_1 > y_e$ ve $y_3 > y_e$ ise

$sn_1 = (x_1, y_e) : sn_2 = (x_3, y_e) \Rightarrow ss_1^k = (n_1, sn_1) : ss_2^k = (n_3, sn_2)$

$ss_3^k = (sn_1, sn_2)$ olur.

2. *Durum*: Ucu Konik parçalar için (Şekil 8.7.(b))

$y_1 > y_e$ ve $y_2 = y_e$ ise

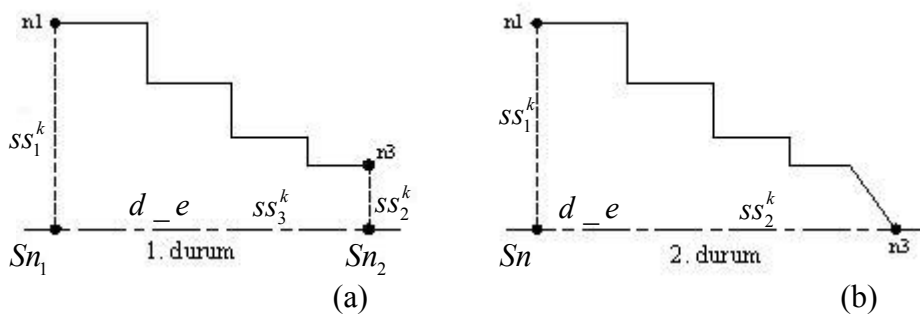
$$sn = (x_l, y_e) \Rightarrow ss_1^k = (n_l, sn): ss_2^k = (n_3, sn)$$

olur.

Farklı durumlarda bu kuralların kombinasyonları kullanılabilir. Şekil 8.7.(b)'deki temas durumunun en küçük köşe noktası (n_l) için geçerli olduğunu varsayarsak sanal nokta benzer tarzda ilgili köşe noktasının x ve eksenin y koordinatından oluşturulur.

Bu şekilde, dönel simetrik elemanın döndürülecek profilini oluşturan kenarlar için döndürme eksenine göre dört çeşit konum tespit edilmiş ve ilgili kurallar oluşturulmuştur. Bu konumlar aşağıdaki şekildedir:

- iki köşesi de eksenin üzerinde,
- en büyük köşesi eksene temas halinde, en küçük köşesi eksen üzerinde,
- en küçük köşesi eksene temas halinde, en büyük köşesi eksen üzerinde,
- her iki köşesi de eksene temas halinde,

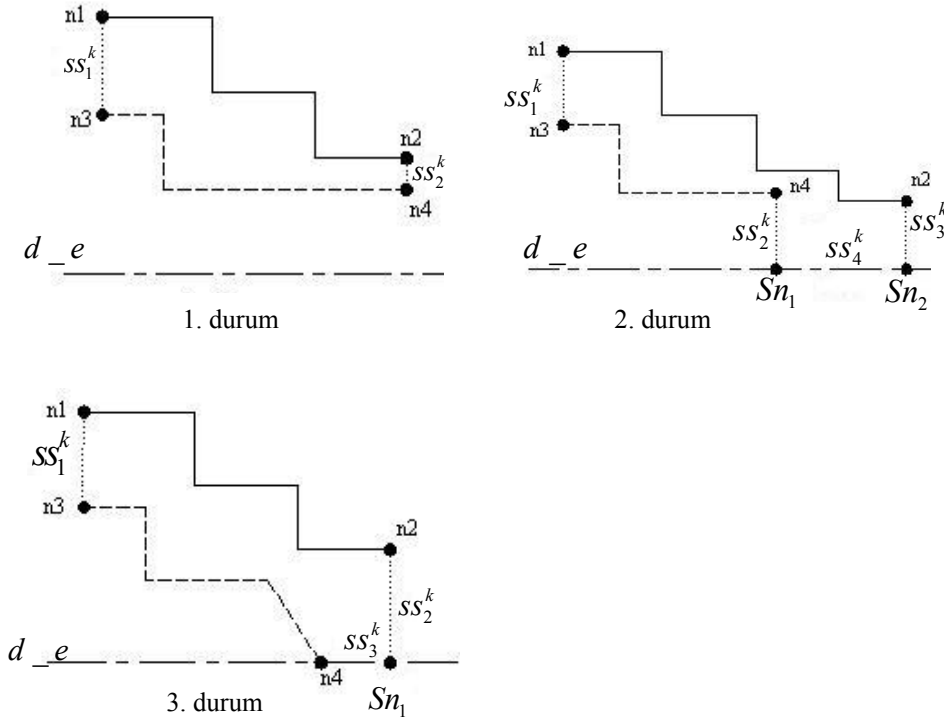


Şekil 8.7. İçi dolu parçalarda son sanal kenarlar:

- Ucu düz
- Ucu konik

b) *İçi boş parçalarda son sanal kenarların oluşturulması*: Bu tür parçaların son sanal kenar oluşturma işlemi içi dolu parçalarda kullanılan yöntemle benzer. Fakat burada görünmez ve görünür kenarların birbirlerine göre konumları da dikkate alınmaktadır.

Burada, görünür ve görünmez kenarların uç noktalarının (en büyük veya en küçük köşe noktaları) konumuna göre üç farklı durum tanımlanmıştır (Şekil 8.8).



Şekil 8.8. Görünmez kenar içeren parçalarda son sanal kenarlar

Şekil 8.8’de tanımlanan üç farklı durumdaki 2B’lu görünüşe ait köşe noktaları aşağıdaki şekilde tanımlanmış olsun:

$n_1 = (x_1, y_1, z_1)$ ve görünür kenarların en küçük x koordinatına sahip köşe noktası,
 $n_2 = (x_2, y_2, z_2)$ ve görünür kenarların en büyük x koordinatına sahip köşe noktası,
 $n_3 = (x_3, y_3, z_3)$ ve görünmez kenarların en küçük x koordinatına sahip köşe noktası,
 $n_4 = (x_4, y_4, z_4)$ ve görünmez kenarların en büyük x koordinatına sahip köşe noktası,
 $d_e = \{ (x_e, y_e, z_e) (x_{e+1}, y_{e+1}, z_{e+1}) \}$ döndürme eksenini tanımlansın. Buna göre;

1. Durum: Eğer görünür ve görünmez kenarların en küçük veya en büyük köşe noktalarının $(n_1 - n_3 / n_2 - n_4)$ x koordinatları birbirine eşitse son sanal kenarların (ss_n^k) köşe nokta koordinatları bu köşe noktalarının koordinat değerlerine eşittir (Şekil 8.8).

$x_1 = x_3$ ise $ss_1^k = (n_1, n_3)$: Eğer $x_2 = x_4$ ise $ss_2^k = (n_2, n_4)$ olur.

2. *Durum*: Eğer, görünür ve görünmez kenarların en küçük veya büyük köşe noktalarının $(n_1 - n_3 / n_2 - n_4)$ x koordinatları birbirinden farklı ise son sanal kenarlar (ss_n^k) bu köşe noktalarının ve eksenı oluşturan köşe noktalarının koordinat değerlerinden oluşturulur (Şekil 8.8).

$$x_1 = x_3 \text{ ise } ss_1^k = (n_1, n_3)$$

$$x_2 > x_4 \text{ ve } y_4 = y_e \text{ ise}$$

$$sn_1 = (x_4, y_e) : sn_2 = (x_2, y_e) \Rightarrow ss_2^k = (n_4, sn_1) : ss_3^k = (n_2, sn_2)$$

$$ss_4^k = (sn_1, sn_2)$$

olur.

Tersi durumda, yani;

$$x_1 > x_3 \text{ ve } y_3 = y_e \text{ ise}$$

$$sn_1 = (x_1, y_e) : sn_2 = (x_1, y_e) \Rightarrow ss_1^k = (n_3, sn_1) : ss_2^k = (sn_1, sn_2)$$

$$ss_3^k = (sn_2, n_1)$$

$$x_2 = x_4 \text{ ise } ss_4^k = (n_2, n_4)$$

olur.

3. *Durum*: Eğer, görünür ve görünmez kenarların en küçük veya büyük köşe noktalarının $(n_1 - n_3 / n_2 - n_4)$ x koordinatları birbirinden farklı ve bu köşe noktalarından biri eksen üzerindeyse (konik) ise son sanal kenarlar (ss_n^k) bu köşe noktalarının ve eksenı oluşturan köşe noktalarının koordinat değerlerinden oluşturulur (Şekil 8.8).

$$x_1 = x_3 \text{ ise } ss_1^k = (n_1, n_3)$$

$$x_2 > x_4 \text{ ise } sn_1 = (x_2, y_e) \Rightarrow ss_2^k = (n_2, sn_1): ss_3^k = (n_4, sn_1) \text{ olur.}$$

Tersi durumda, yani:

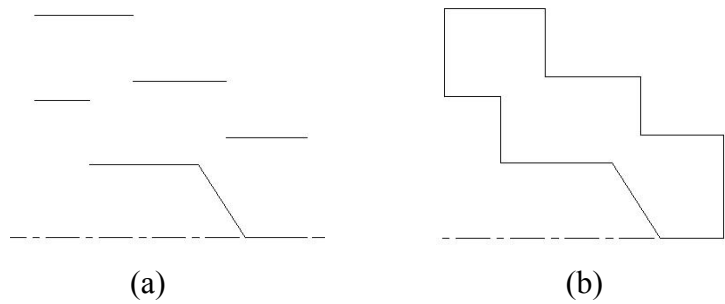
$$x_1 = x_3 \text{ ise } ss_1^k = (n_1, n_3)$$

$$x_2 > x_4 \text{ ise } sn_1 = (x_2, y_e) \Rightarrow ss_2^k = (n_3, sn_1): ss_3^k = (n_1, sn_1) \text{ olur.}$$

Burada görünür ve görünmez kenarların en büyük ve küçük köşe noktalarının yedi çeşit birleşim türü ele alınmış ve ilgili kurallar hazırlanmıştır.

8.2.5. Tekrar çizim ve model alanı oluşturma

2B'lu görünüşü oluşturan kenarlarla ilgili düzenleme ve sıralama işlemi tamamlandıktan sonra model alanı oluşturulur. Model alanı, izdüşümün eksen üzerinde kalan kenarlarından oluşturulmaktadır. Burada önce, eksen üzerinde kalan doğru ve yaylar çizilmektedir. Daha sonra, sanal kenarlar çizilmektedir (Şekil 8.9).



Şekil 8.9. Tekrar çizim ve model alanı oluşturma:

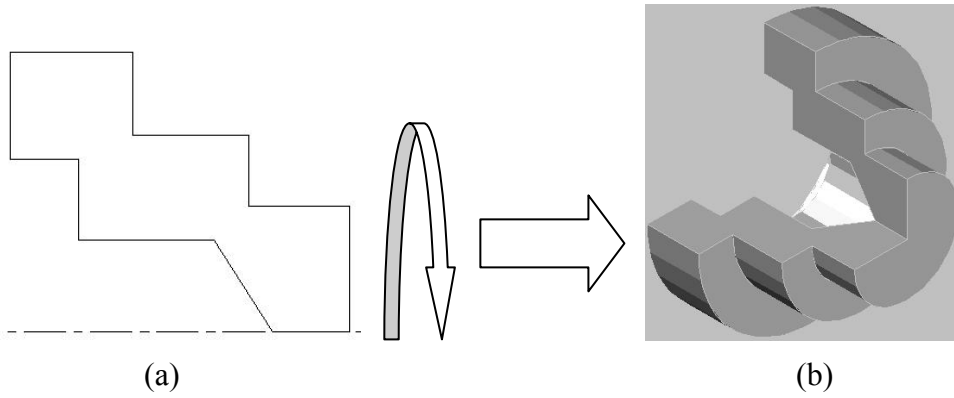
a) tekrar çizim

b) Model alanı

Çizim daha sonra birleşik çizgi (pline) haline dönüştürülerek model alanı oluşturulmaktadır. Çizim işleminde AutoCAD line ve arc çizim elemanları kullanılmaktadır. Birleşik çizgi haline dönüştürme işlemi ise “pedit” komutuyla yapılır.

8.2.6. Katı model

Katı model, model alanının döndürme eksenini etrafında 360 derece süpürülmesi ile oluşturulur. Şekil 8.10'da katı modelin iç kısmını görüntüleyebilmek için profil süpürme açısı 270 derece olarak alınmıştır.



Şekil 8.10. Katı model oluşturma:

- a) aksenal döndürme
- b) katı model

Eksenel döndürme AutoCAD “Revolve” komutu yardımıyla gerçekleştirilmektedir. Bütün bu çizim ve katı model işlemleri için kullanılan AutoCAD komutları, VisualLISP kodunda yazılan program parçacıkları yardımıyla otomatik olarak çalıştırılmaktadır.

9. UZMAN SİSTEM YARDIMIYLA KATI MODELLER OLUŞTURMA

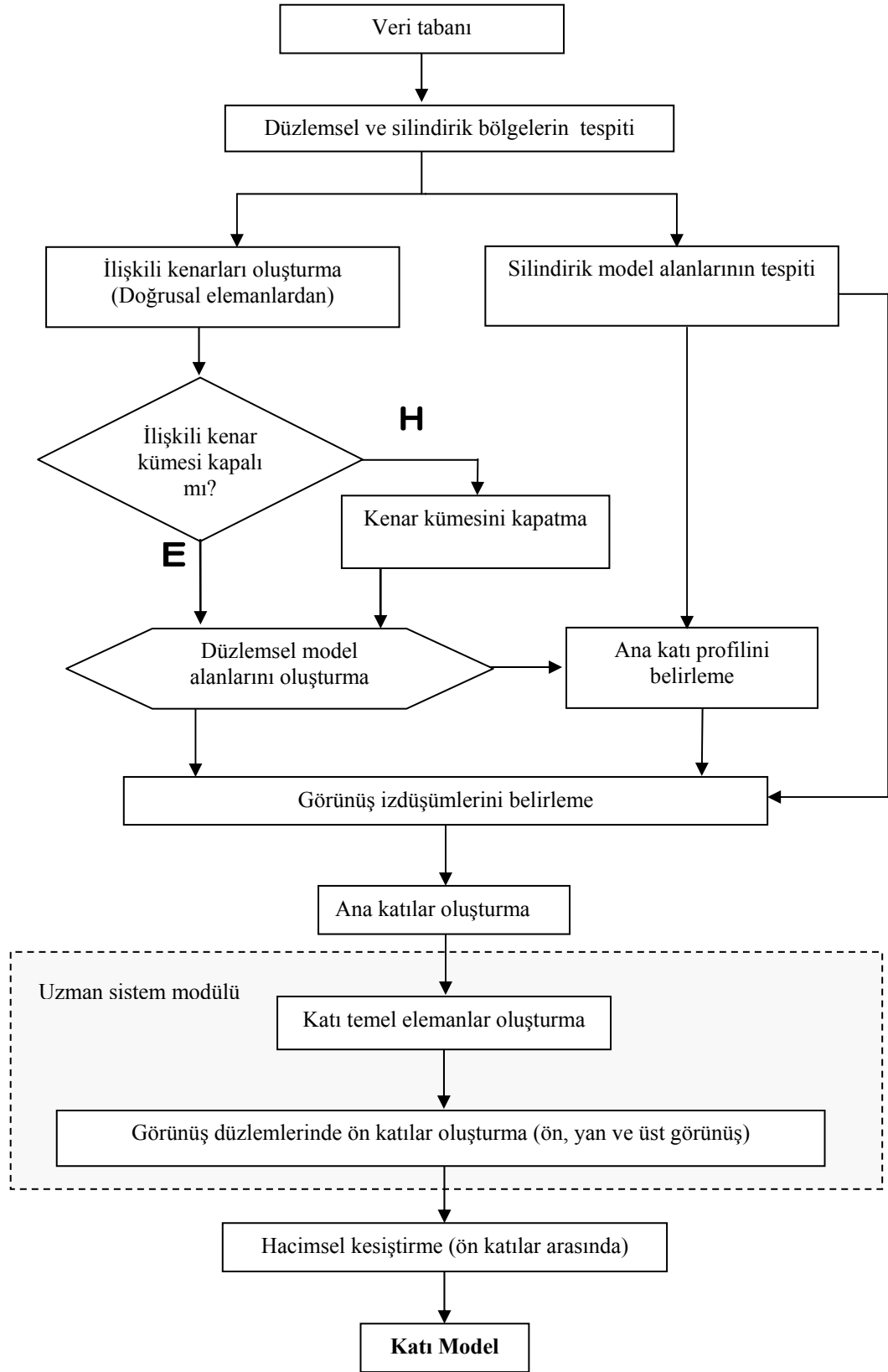
Bu bölümde iki veya üç görünüş girdisi ile ifade edilen silindirik ve prizmatik parçaların katı modellerinin oluşturma işlemi açıklanmaktadır. Hacim tabanlı bir yöntemin benimsendiği çalışmada silindirik ve prizmatik parçalar bir arada ele alınmaktadır. Ayrıca çalışmada iki ve üç görünüş girdisi bir arada kullanılabilir.

Geliştirilen yaklaşımda katı model oluşturma işleminde kullanılan tekniklerin çoğu bir arada kullanılmaktadır. Bunlar, YKG temel elemanları, süpürme teknikleri ve hacimsel kesiştirme olarak sayılabilir. Ayrıca katı temel elemanların biçimine karar verme aşamasında bir uzman sistem kullanılmıştır.

Programın bu modülünün ana işlem adımları şu şekilde belirtilebilir:

- Düzlemsel ve silindirik bölgelerin tespiti,
- İlişkili kenar kümelerini ve bunların türlerini belirleme (düzlemsel yüzeylerde),
- Silindirik alanları belirleme,
- Model alanları oluşturma ve bunların izdüşümlerini tespit etme
- Ana katıyı tespit etme,
- Silindirik ve prizmatik katı temel elemanları oluşturma,
- Katı temel elemanlar ve ana katı arasında boolean (ekleme ve çıkartma) işlemleri ile görünüş bölgelerinde ön katılar oluşturma,
- Ön katılar arasında hacimsel kesiştirme ile katı modeller oluşturma.

Yukarıda özetlenen yaklaşıma ait işlemler, Şekil 9.1’de görülen akış şemasında gösterilmiştir. Bu şemada da görüldüğü gibi, katı temel eleman ve ön katılar oluşturma aşamasında kullanılmak üzere uzman sistem geliştirilmiştir.

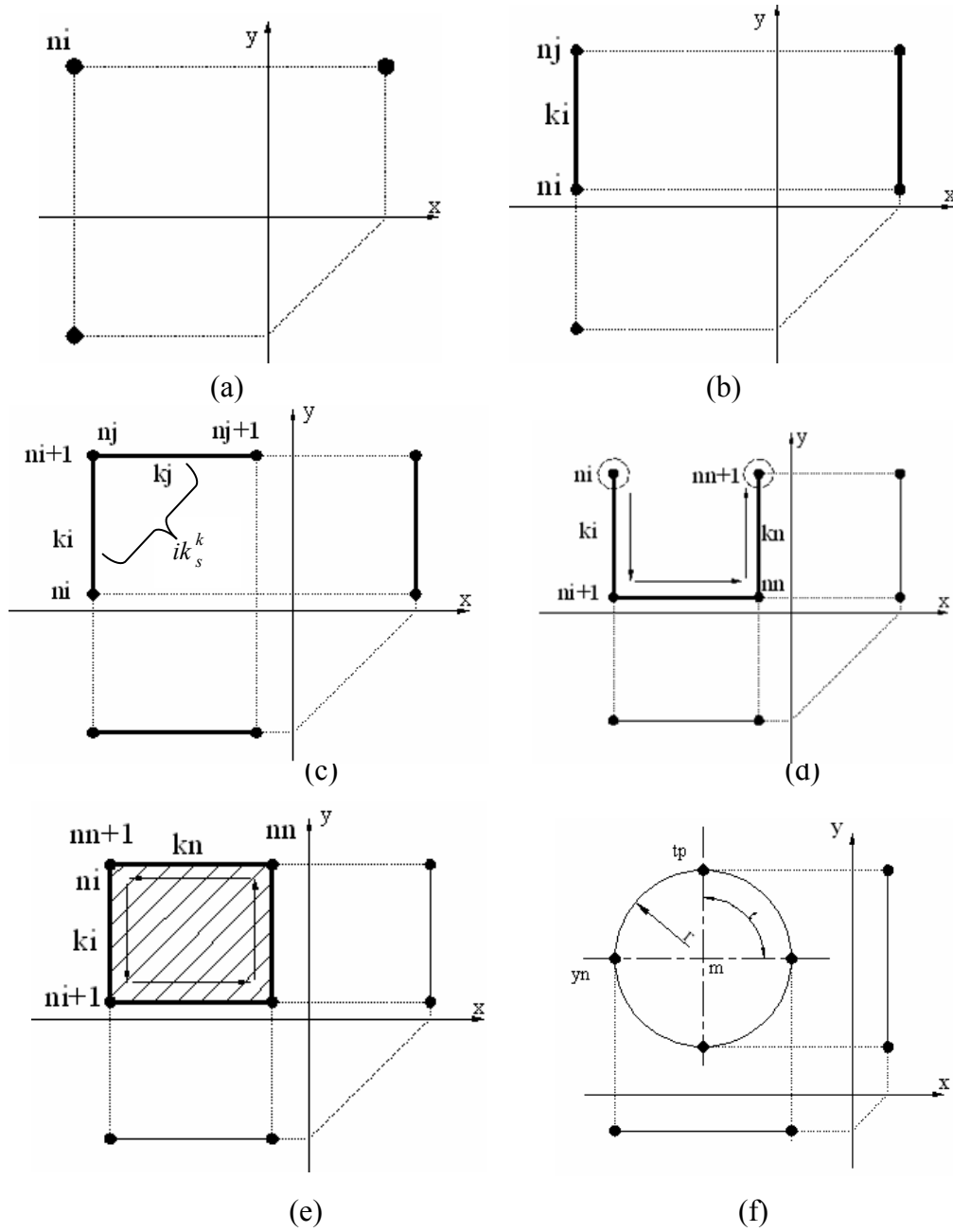


Şekil 9.1. İki ve üç görünüş program modülü ana işlem adımları

9.1. Temel Kavram ve Tanımlamalar

Bu kısımda katı model oluşturma işleminin anlatımı esnasında kullanılacak temel kavram ve tanımlamalar açıklanmaktadır. Bunlar şu şekilde tanımlanabilir.

- i. *Köşe Noktası* (n_i): Bir köşe noktası üç temel koordinatla tanımlanır. n köşe noktasını, “ i ” ilgili köşe noktasının indisini temsil etmek üzere (Şekil 9.2.(a));
 $n_i = \{ (x_i, y_i, z_i) : 1 \leq i \leq n ; n, i \in \mathbb{Z} \}$ olmalıdır.
- ii. *Kenar* (k_i): Bir kenar, iki köşe noktasını birleştiren doğru parçasıdır. k kenarı, i ilgili kenarın indisini temsil etmek üzere (Şekil 9.2.(b));
 $k_i = \{ (n_i, n_j) : i \neq j : 1 \leq i \leq n ; 1 \leq j \leq n ; n, i, j \in \mathbb{Z} ; n_i \not\subset n_j \Leftrightarrow n_j \not\subset n_i \}$ olmalıdır.
- iii. *İlişkili Kenar Kümesi* (ik_s^k): Görünüşte yer alan 2B doğru ve yay gibi kenarların bir veya daha çok ortak köşe noktası ile birbirine bağlanmışsa, bu kenarlara ilişkili kenar, bu kenarların oluşturduğu kümeye ilişkili kenar kümesi denir (Şekil 9.2.(c)).
- iv. *Kapalı kenar* (k_s^k): İlişkili kenar kümesindeki ilk kenar ve son kenarın son nokta koordinatları birbirine eşit ise bu kenar kümesi kapalı kenar kümesi olarak isimlendirilir (Şekil 9.2.(d)). s , kenar küme numarasını göstermektedir.
- v. *Açık kenar Kümesi* (a_s^k): Bir açık kenar kümesinde ilk ve son kenar koordinatları farklı ise, yani ilişkili kenar kümesi kapalı bir bölge oluşturmuyorsa bu kenar kümesine açık kenar kümesi denir (Şekil 9.2.(e)). s , kenar küme numarasını göstermektedir.
- vi. *Dairesel Bölge* (db): Dairesel bölgeler tam silindirik unsurların 2B’lu görünüşüdür ve doğrusal ayrıntı içermezler (Şekil 9.2.(f)).



Şekil 9.2. 2B izdüşümlerle ilgili temel kavramlar:

- a) köşe noktası
- b) Kenar
- c) İlişkili kenar
- d) kapalı kenar kümesi
- e) açık kenar kümesi
- f) dairesel bölge

vii. *Ana katı*: Genellikle en büyük alana sahip düzlemsel veya silindirik kapalı bölgeden, her görünüş bölgesinde oluşturulan katıdır. Temel katılar, ana katılar üzerine eklenip çıkarılır.

- viii. *ön katı*: Ön katı, ilgili görünüş bölgesindeki (ön, yan veya üst) temel katıların, boolean operasyonları ile ana katıdan eklenip çıkarılması ile oluşturulur.
- ix. *Ortak Koordinat Eksen*i: Girdi görünüşü ile izdüşümünün paylaştığı ortak koordinat eksenidir. Ortak koordinat eksenini ön-yan görünüşler arasında “y” eksenini, ön – üst görünüşler arasında ise “x” eksenidir (Şekil 9.2).

9.2. Düzlemsel ve Silindirik Bölgeleri Saptama

Silindirik ve prizmatik katı temel elemanlar, düzlemsel ve dairesel kapalı bölgelerden oluşturulur. Bu amaçla öncelikle düzlemsel ve silindirik bölgelerin saptanması gerekmektedir.

Silindirik bölgeler: Silindirik bölgeler, dairelerden oluşur (Şekil 9.2.(f)). Her bir dairesel bölge (*db*), silindirik katı temel elemanın oluşturulacağı kapalı bölgedir. Bu bölgelerle ilgili bilgiler, 2B’lu daire verilerinin aktarıldığı veri tabanından elde edilir.

Düzlemsel bölgeler: Düzlemsel bölgeler ise, doğru (*line*) ve yayların (*arc*) kapalı bölgelerinden oluşur. Düzlemsel bölgeleri oluşturan 2B elemanlar veri tabanında düzensiz olarak kayıtlıdır. Düzlemsel bölgelerin düzenli bir şekilde oluşturulabilmesi için bu verilerin düzenlenmesi gerekir. Bu düzen, veri tabanında kayıtlı yay ve doğruların bir zincir oluşturacak şekilde sıralanmasıyla gerçekleştirilir. Bunun için, 2B’lu yay ve doğru verilerinin birbirleri ile ilişkilendirilerek düzenli 2B eleman kümelerine dönüştürülmesi gerekmektedir. Çalışmada bu işleme *ilişkili kenar oluşturma* ismi verilmiştir.

9.3. İlişkili Kenar Kümesi Oluşturma (ik_s^k)

Birbirini takip eden iki kenarın ilişkili olup olmadığını tespit etmede temel amaç, kenar zincirlerini ortaya çıkarmaktır. Böylece 2B izdüşüm görüntüsünü oluşturan kapalı ve açık kenar kümeleri düzenlenmektedir. Şekil 9.3’te verilen 2B izdüşümde

birbirini takip eden k_i ve k_j kenarları için aşağıdaki şekilde tanımlansın:

$$k_i = \{ (n_i, n_{i+1}) : n_i < n_{i+1} \} : k_j = \{ (n_j, n_{j+1}) : n_j < n_{j+1} \} : i < j$$

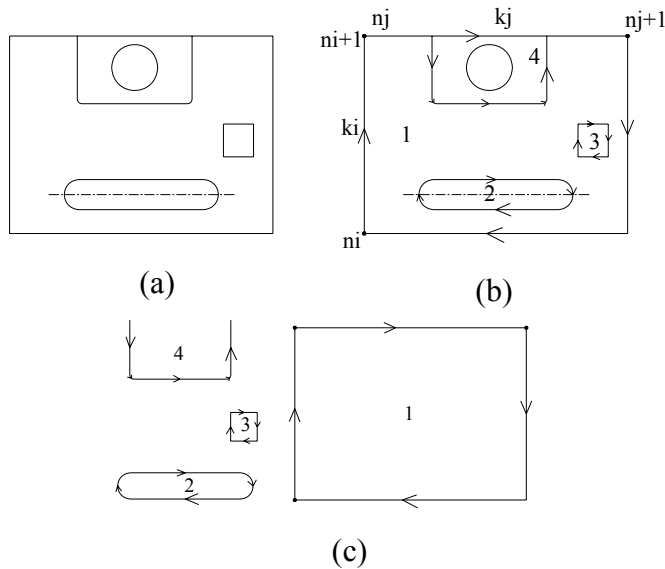
$$n_{i...m} = (x_{i..m}, y_{i..m}, z_{i..m}) : n_{j...m} = (x_{j..m}, y_{j..m}, z_{i..m}) : 1 \leq i \leq m : 1 \leq j \leq m :$$

$m, j \in \mathbb{Z}$ olmak üzere;

eğer $n_{i+1} = n_j$ ise $k_i \in \mathbf{ik}_s^k$, $k_j \in \mathbf{ik}_s^k$ olur.

Bu durum, k_i ve k_j kenarlarının birbirleri ile ilişkili ve ilişkili kenar kümesinin birer üyesi olduğunu gösterir.

İlişkili kenar kümesi oluşturma işlemine, her bir çizim grubuna ait en küçük x koordinatlı kenardan başlanır (k_i). Daha sonra bu kenarın ilk veya son noktasının koordinat nokta değerlerine (x ve y) eşit kenar aranır. Bu kenar (k_j) bulunduğunda ilk kenar (k_i), ilişkili kenar listesine kaydedilir. İlk kenara eşit olan kenar (k_j) için işlem tekrar edilerek ilişkili kenar kümesi oluşturulur. İlişkilendirilecek eleman kalmadığında ise son eleman ilişkili kenar kümesine kaydedilerek küme kapatılır. Küme numarası (s) bir arttırılarak yeni ilişkili kenar kümesi için işlem tekrarlanır (Şekil 9.4).



Şekil 9.3. İlişkili kenarlar:

- a) Temel görünüş
- b) İlişkili (1, 2, 3)
- c) kapalı (1, 2, 3) ve açık kenar kümeleri (4)

```

S = 1 ' ilişkili kenar küme numarası
If n > 1 then 'eleman sayısı birden fazla ise

For i = 1 to n

For j = i+1 to n

If xi+1 = xj and yi+1 = yj and zi+1 = zj
or xi = xj and yi = yj and zi = zj then
    ki ∈ isk : ki = kj
else: end if
next j: next i
if i = n then kj ∈ isk
s = s+1
else : end if

```

Şekil 9.4. İlişkili kenar oluşturma algoritması

İlişkili kenar oluşturma işleminde yaylar çizgilerle birlikte işleme alınmaktadır. Yayların başlangıç ve bitiş köşe nokta koordinatları doğru köşe noktaları ile birlikte işleme alınır. Yay ilk veya son noktalarına eşleşen köşe bulunduğu ilişkili kenar kümesine doğrularda olduğu gibi ilk ve son nokta koordinatları ile birlikte dahil edilir.

Kapalı kenar kümesi (k_s^k)

Kapalı kenar kümesi ilişkili kenar kümelerinin kapalı döngüsünden oluşur. İlişkili kenar kümesine ait ilk köşe noktası ile son köşe noktasının *x* ve *y* koordinatlarının her ikisi de aynı değere sahipse bu kümeye kapalı kenar kümesi denir. 9.3.(c)'de görülen (1, 2 ve 3) numaralı kenar kümesinden herhangi biri aşağıdaki şekilde tanımlanmış olsun:

$$i_s^k = \{ k_i, k_{i+1}, \dots, k_n \}; k_i = (n_i, n_{i+1}); k_n = (n_n, n_{n+1}); n_i < n_{i+1} \text{ ve } n_n < n_{n+1}$$

olmak üzere

eğer $n_i = n_{n+1}$ ise $i_s^k \in k_s^k$ 'dir.

Açık kenar kümesi (a_s^k)

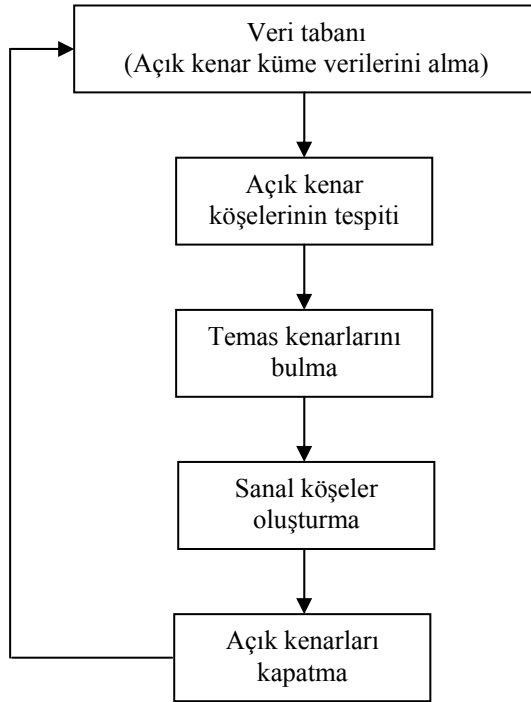
Eğer ilişkili kenar kümesi kapalı bir alan oluşturamıyorsa bu tür kümelere açık kenar kümesi denir. Açık kenar kümelerinde, ilişkili kenar kümesine ait ilk köşe noktası ile son köşe nokta koordinatları birbirine eşit değildir. Bir ilişkili kenar kümesinin açık kenar kümesi olarak değerlendirilmesi için köşe noktalarını oluşturan x veya y koordinatlarından birisinin veya her ikisinin de farklı olması gerekmektedir. Şekil 9.3.(c)'de görülen 4 numaralı kenar kümesinin açık kenar kümesi olarak tanımlanabilmesi için aşağıdaki şartları sağlaması gerekir:

$$i_s^k = \{ k_i, k_{i+1}, \dots, k_n \}; k_i = (n_i, n_{i+1}); k_n = (n_n, n_{n+1}); n_i < n_{i+1} \text{ ve } n_n < n_{n+1} \\ \text{olmak üzere} \\ \text{eğer } n_i \neq n_{n+1} \text{ ise } i_s^k \in a_s^k \text{ 'dır.}$$

Açık ve kapalı kenar kümeleri tespit edildikten sonra açık kenar kümelerinin kapalı kenar kümelerine dönüştürülme işlemi gerçekleştirilmektedir. Açık kenar kümeleri, 3B nesnelerin köşe cep, kapalı kanal vb. unsurlarını oluşturur. Açık kenar kümelerini kapalı kenar kümelerine dönüştürmek amacıyla Açık Kenar Kümesi Kapatma Algoritması (AKKA) ismi verilen bir algoritma geliştirilmiştir.

9.4. Açık Kenar Kümesi Kapatma Algoritması (AKKA)

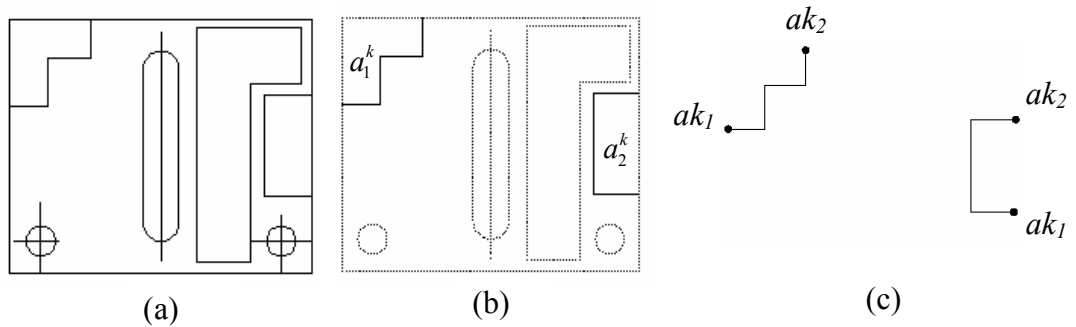
Açık kenar kümelerini kapatma algoritması dört aşamalı bir işlemi içerir. Bunlar, açık kenar kümesinin açık köşelerinin tespiti, temas kenarlarının bulunması, sanal köşeler oluşturma ve açık kenarın kapatılması (Şekil 9.5)



Şekil 9.5. Açık kenar kümesi kapatma işlemi

9.4.1. Açık kenar kümesinin açık köşe noktalarının tespiti

Açık kenar kümesinin açık köşeleri, açık kenar kümesinde herhangi bir köşe noktası ile kesişmeyen köşe noktalarıdır (Şekil 9.6). Şekil 9.6.(b)'de açık kenar kümeleri, 9.6.(c)'de ise bu kenar kümelerine ait açık köşe noktaları görülmektedir.



Şekil 9.6. Açık köşe noktalarının tespiti:

- a) 2B görünüş
- b) açık kenar kümeleri
- c) açık köşe noktaları

9.4.2. Temas kenarlarının (tk) bulunması

Açık köşe noktaları tespit edildikten sonra, bu köşe noktalarının temas ettiği kenar veya kenarlar bulunur. Bu kenarlar, açık köşe koordinatının herhangi birine (“ x ” veya “ y ”) eşit olan kenarlardır. Şekil 9.7.(a)’da ak_i ve ak_j olarak verilen açık köşelerin temas ettiği kenarlar (tk_s) aşağıdaki şekilde bulunur:

$$ak_i = \{ (x_i, y_i, z_i) : 1 \leq i \leq m : m, i \in \mathbb{Z} \text{ ve } ak_i \in \mathbf{a}_s^k$$

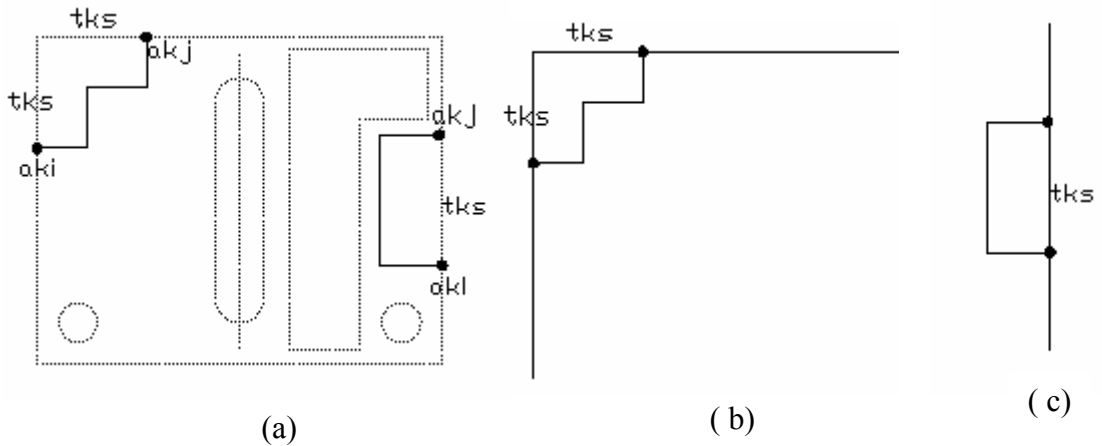
$$ak_j = \{ (x_j, y_j, z_j) : 1 \leq j \leq m : m, j \in \mathbb{Z} \text{ ve } ak_j \in \mathbf{a}_s^k \text{ ve}$$

$$tk_s = \{k_s, k_{s+1}\} ; k_s = (x_s, y_s, z_s) ; k_{s+1} = (x_{s+1}, y_{s+1}, z_{s+1}) : k_s < k_{s+1} : 1 \leq s \leq n:$$

$n, i \in \mathbb{Z}$ ve $i \neq j$ olsun

$x_j = x_s$ veya $y_j = y_s$ veya $x_j = x_{s+1}$ veya $y_j = y_{s+1}$ ise tk_s açık köşenin temas kenarıdır.

Bu şekilde, açık köşelerin temas ettiği kenarlar bulunur. Eğer bir açık kenar kümesinin açık köşe koordinatları ($ak_i \neq ak_j$) birbirinden farklı ise iki temas kenarı vardır (Şekil 9.7.(c)). Eğer açık köşe noktalarının koordinatlarından herhangi biri (x veya y) eşitse tek temas kenarı vardır (Şekil 9.7.(b)).

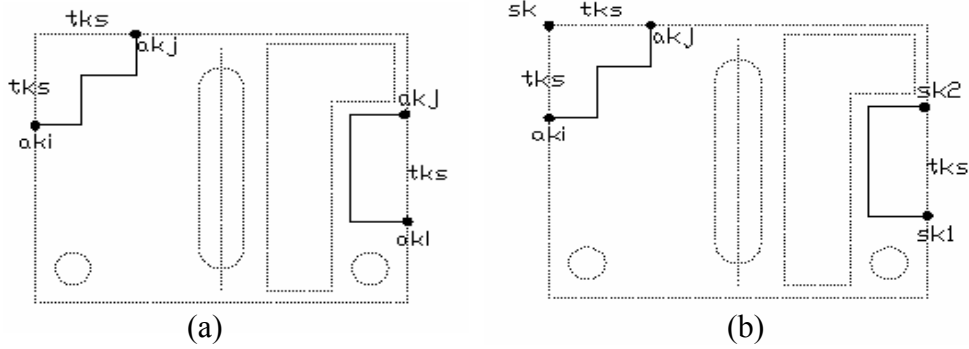


Şekil 9.7. Temas kenarlarını bulma:

- a) temas kenarları
- b) iki temas kenarı
- c) tek temas kenarı

Sanal köşeler oluşturma: Sanal köşeler (s_k), iki temas kenarlarının kesiştiği noktadır.

Sanal köşeler, açık kenarın türüne göre iki farklı yapıdadır. Bir açık kenar kümesinin tek veya iki sanal köşesi olur. Eğer açık kenar kümesi Şekil 9.8.(a)'deki gibi iki temas kenarına sahip ise tek sanal köşeye sahiptir (Şekil 9.8.(b)). Burada oluşan sanal köşe, temas kenarlarının ortak kesişim noktasıdır.



Şekil 9.8. Sanal köşe oluşturma

Eğer tek temas kenarına sahipse, yani açık köşe noktalarının herhangi bir koordinatları eşit ise (x veya y) iki sanal köşe oluşur. Bu durumda sanal köşe koordinatları açık köşelerin koordinatlarından oluşturulur (Şekil 9.8.(b)). Sanal köşe oluşturma algoritması aşağıda verilmiştir.

$ak_i \in a_s^k$ ve $ak_j \in a_s^k$ ise

$x_i <> x_j$ ve $y_i <> y_j$ ise

$s_k = (x_i, y_j)$ olur.

$x_i = x_j$ ve $y_i <> y_j$ veya $x_i <> x_j$ ve $y_i = y_j$ ise

$s_{k1} = (x_i, y_i) : s_{k2} = (x_j, y_j)$ olur.

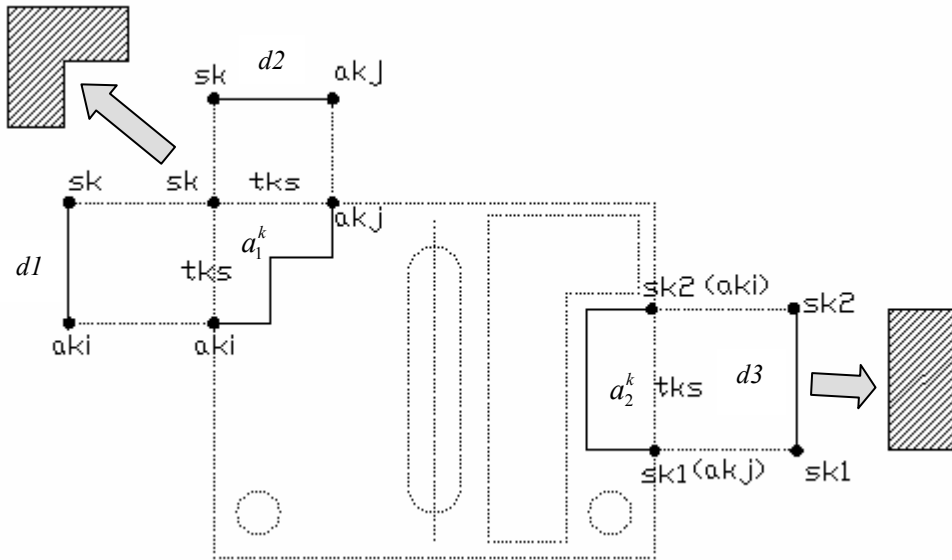
9.4.3. Açık kenar kümelerinin kapatılması

AKKA'da son adım açık kenar kümelerinin kapatılmasıdır. Burada işlem açık kenar kümelerinin yer aldığı veri tabanının tekrar düzenlenmesidir. Bu çerçevede veri tabanına sanal köşe ve açık köşe koordinatlarından oluşturulan yeni kenar koordinatları dahil edilir.

Sanal köşeler veri tabanına dahil edilirken açık kenar kümelerini kapatacak doğru parçaları şekline aktarılmaktadır. Bir doğru iki köşe noktasından oluşur. Açık kenar kümelerini kapatacak doğrular, sanal köşe koordinatları ve açık köşe koordinatlarından oluşturulur. Bu işlem, açık köşelere ait koordinat çiftleriyle sanal köşe koordinatlarının bir doğru parçası oluşturacak şekilde bir araya getirilmesi ile yapılır.

Eğer bir açık kenarın (a_1^k) tek sanal köşesi varsa bu açık kenar iki doğru parçası oluşturacak şekilde ($d1, d2$), eğer iki sanal köşeye sahipse (a_2^k) tek doğru parçası oluşturacak şekilde ($d3$) veri tabanına veri aktarımı gerçekleştirilir (Şekil 9.9). Kenarların köşe koordinat çiftleri aşağıdaki şekilde belirlenir:

$$d_1 = \{ak_i, sk\}; d_2 = \{ak_j, sk\}; d_3 = \{sk_1, sk_2\}$$

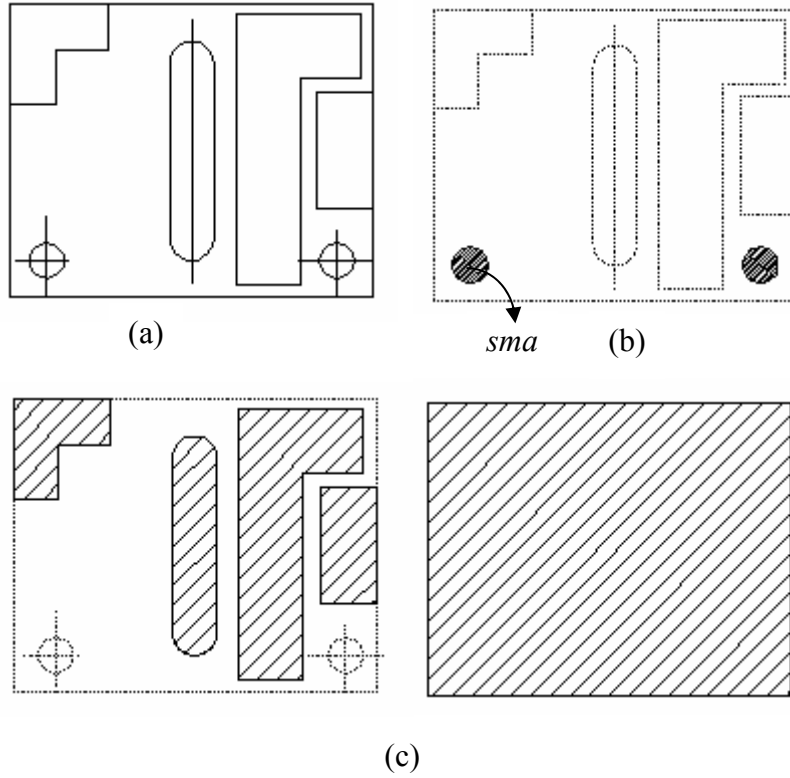


Şekil 9.9. Açık kenar kümelerini kapatma

9.5. Düzlemsel Model Alanı (*dma*) Oluşturma

Düzlemsel model alanları (*dma*), kapalı kenar kümeleri ve kapalı kenar kümelerine dönüştürülmüş açık kenar kümelerinden oluşturulmaktadır (Şekil 9.10.(c)). Sadece

doğrulardan oluşan düzlemsel bölgelerin yanında, düzlemsel unsurlar içeren yarı silindirik bölgeler de (kama kanallı silindirik parçalar vb.) düzlemsel model alanı olarak değerlendirilmektedir.



Şekil 9.10. Düzlemsel ve silindirik model alanları:

- a) 2B izdüşüm
- b) silindirik model alanları
- c) düzlemsel model alanları

9.6. Silindirik Model Alanlarının (*sma*) Tespiti

Girdi görüşlerindeki dairesele bölgeler silindirik model alanları (*sma*) olarak belirlenmektedir (Şekil 9.10.(b)). Bu bölgelerin, merkez koordinatları, yarıçap değerleri, üst uç ve yan uç nokta koordinatları oluşturulan veri tabanına aktarılmaktadır. Bu aşamada silindirik bölgelerin alanları da hesaplanıp veri tabanına kaydedilmektedir.

9.7. Ana Katı Profilini Belirleme

Bu aşamada düzlemsel ve silindirik model alanlarından görünüş bölgesinde oluşturulacak ana katı profiline karar verilir. Ana katı, görünüş bölgelerinde oluşturulacak ön katıların oluşturulmasında referans katılardır. Ön katılar, ana katı üzerine temel katıların eklenmesi veya çıkarılmasıyla oluşturulur. Ana katı profiline üç aşamalı bir işlemle karar verilir. Bunlar;

1. Düzlemsel model alanlarından düzlemsel ana katı profilini seçme
2. Silindirik model alanlarından silindirik ana katı profilini seçme
3. Düzlemsel ve silindirik ana katı profillerinden ana katı profiline karar verme

9.7.1. Düzlemsel model alanlarından düzlemsel ana katı profili belirleme

Düzlemsel ana katı profili, kapalı kenar kümelerinden oluşturulan *dma*'lar arasında en büyük yüz ölçüme sahip model alanıdır. Düzlemsel bölgelerin alan yüzölçümleri en küçük köşe noktasının x ve y koordinatı ile, en büyük köşe noktasının x ve y koordinatlarının farklarının çarpımları ile hesaplanır.

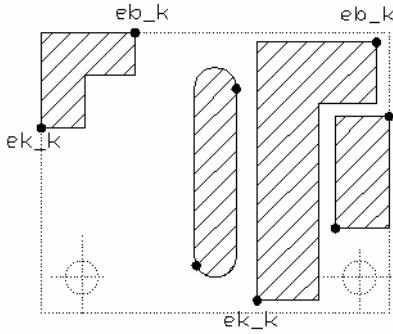
Şekil 9.11'de verilen düzlemsel model alanlarının en küçük köşe noktaları ek_k , en büyük köşe noktaları eb_k olarak gösterilsin. Bu iki köşe noktası arasında kalan düzlemsel bölgenin alanı (d_a) aşağıdaki şekilde hesaplanır.

$$ek_k = \{ (x_i, y_i, z_i) : 1 \leq i \leq m : m, i \in \mathbb{Z},$$

$$eb_k = \{ (x_j, y_j, z_j) : 1 \leq j \leq n : n, j \in \mathbb{Z} \text{ ve } x_j > x_i, y_j > y_i \text{ olmak üzere.}$$

$$x_s = x_j - x_i : y_s = y_j - y_i : d_a = x_s \times y_s$$

Yukarıdaki eşitlikle bütün düzlemsel model alanları hesaplanır. Düzlemsel ana katı profili bu düzlemsel model alanları arasındaki en büyük yüzölçüme sahip model alanıdır (Şekil 9.12). Düzlemsel ana katı profili belirlendikten sonra, ana katı profili ile ilgili bilgiler veri tabanına aktarılır.



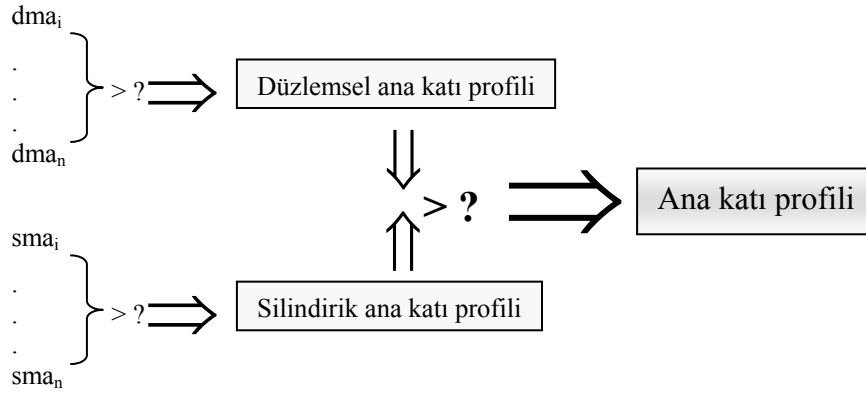
Şekil 9.11. Düzlemsel model alanlarının (*dma*) en küçük ve en büyük köşe noktaları

9.7.2. Silindirik model alanlarından silindirik ana katı profilini seçme

Silindirik ana katı profili seçme işlemi, düzlemsel ana katı profili seçme işlemine benzer şekilde silindirik model alanları (*sma*) içerisinde en büyük yüz ölçüme sahip model alanının tespit edilmesidir (Şekil 9.12). Silindirik ana katı profili belirlendikten sonra, geometrik biçimini ve alan yüzölçümünü veren değerlerle birlikte veri tabanına aktarılır. Bu değerler, merkez x ve y koordinatları, üst uç (tp) ve yan uç (yn) nokta koordinatlarıdır.

9.7.3. Ana katı profiline karar verme

Düzlemsel ve silindirik ana katı profilleri seçildikten sonra, bu iki profil arasından görünüş bölgesindeki ana katının oluşturulacağı ana katı profiline karar verilir. Ana katı profili, düzlemsel ve silindirik ana katı profilleri arasındaki en büyük alana sahip katı profilidir (Şekil 9.12).

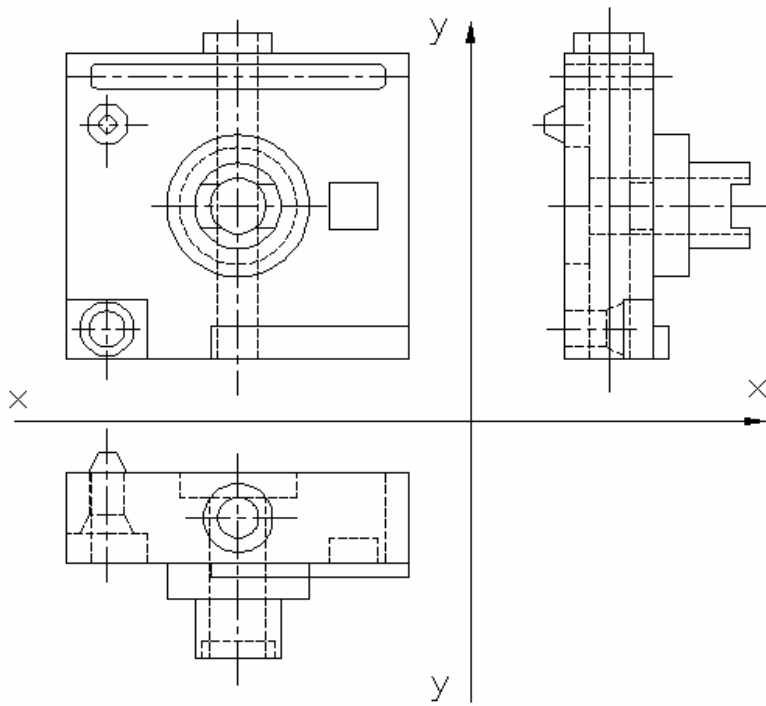


Şekil 9.12. Düzlemsel, silindirik ve ana katı profillerine karar verme

9.8. Görünüş İzdüşümlerini Belirleme

Düzlemsel ve silindirik model alanları ile ana katı profillerinin 3B yorumlarını oluşturabilmek için izdüşümlerini belirlemek gerekir. Model alanını oluşturan doğru veya yay son noktalarının herhangi bir koordinatı (x veya y) ile komşu görünüşte eşleşen doğru parçası model alanının izdüşümü olarak belirlenir. İzdüşümler, model alanlarının oluşturacağı temel ve ana katıların kalınlık, 3B’lu nesne türü, 3B’lu konumlarının belirlenmesi vb. işlemlerde kullanılmaktadır.

İzdüşümlerin belirlenmesi aşamasında öncelikle görünüşlerin ortak koordinat eksenlerinin belirlenmesi gerekmektedir. Ortak koordinat eksen, model alanının (düzlemsel veya silindirik) izdüşümleri bulunurken görünüş ve izdüşümü arasında eşitliği aranacak koordinat eksenidir. Yani, girdi görünüşü ile izdüşümünün paylaştığı ortak koordinat eksenidir. Ortak koordinat eksen, ön-yan görünüşler arasında “ y ” eksen, ön – üst görünüşler arasında ise “ x ” eksenidir (Şekil 9.13).



Şekil 9.13. Ortak koordinat eksenleri

Ortak koordinat eksenleri belirlendikten sonra görünüş izdüşümleri araştırılır. Bir düzlemsel veya silindirik model alanının izdüşümü, komşu görünüşteki ortak koordinat eksen değerine eşit köşe noktalarıdır. Arama işlemi ön-yan, yan-ön ve üst-ön görünüş çiftleri arasında yapılır. Eğer model alanı ön görünüşte ise, izdüşümü yan görünüşte, model alanı yan görünüşte ise izdüşüm ön görünüşte, model alanı üst görünüşte ise izdüşüm ön görünüşte aranır. İzdüşüm arama işlemi düzlemsel ve silindirik model alanlarında farklıdır.

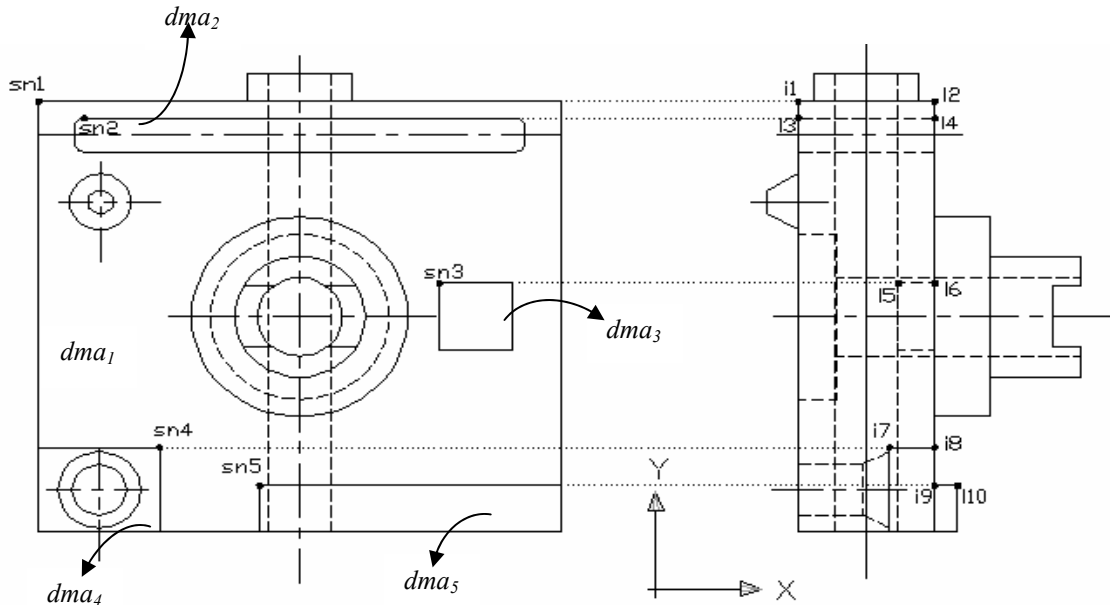
9.8.1. Düzlemsel model alanlarının izdüşümlerinin bulunması

Düzlemsel model alanlarının izdüşümleri belirlenirken model alanını oluşturan daire ve yay parçalarının son nokta koordinatlarına eşit doğru parçaları aranır. Eşit olması istenen koordinatlar görünüşler arasındaki ortak koordinat eksenlerine göre belirlenir. Ortak koordinat eksenine göre izdüşümler aşağıdaki şekilde aranır:

- Eğer düzlemsel model alanı ön görünüşte ise, yan görünüşte düzlemsel model alanını (dma) oluşturan doğru ya da yay parçalarının son nokta y koordinatına eşit köşe noktaları aranır.
- Eğer düzlemsel model alanı yan görünüşte ise, ön görünüşte dma 'nı oluşturan doğru ya da yay parçalarının son nokta y koordinatına eşit köşe noktaları aranır.
- Eğer düzlemsel model alanı üst görünüşte ise, ön görünüşte dma 'nı oluşturan doğru ya da yay parçalarının son nokta x koordinatına eşit doğru parçası aranır.

Arama işleminde öncelikle düzlemsel model alanını oluşturan doğru parçalarının son noktaları kullanılır. Eğer bu doğruların eşleştiği izdüşüm bulunamazsa yay parçalarının tepe noktalarının izdüşümleri aranır.

Şekil 9.14'de görülen bir numaralı düzlemsel model alanının (dma_1) $sn1$ olarak gösterilen köşe noktasının yan görünüşteki izdüşümü, kendisi ile aynı y koordinatına sahip doğru parçasıdır. Burada izdüşüm $i1$ ve $i2$ köşe noktalarından oluşan doğru parçasıdır. Benzer şekilde dma_2 ile gösterilen alanın yan görünüşteki izdüşümü, $sn2$ köşe noktası ile aynı y koordinatına sahip $i3$ ve $i4$ köşe noktalarıdır.



Şekil 9.14. dma 'ların izdüşümleri

Bu işlemle ilgili program algoritması şekil 9.15'te verilmiştir.

```

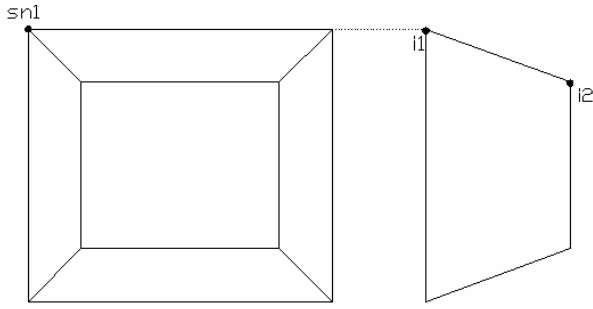
son= model alanını oluşturan doğru ve yay parça sayısı
m= komşu görünüşteki doğru ve yay parça sayısı
dzlm= model alanı izdüşüm kümesi

bas=1
For i = bas to son
  Sn1 ∈ dma1: sn1 = (x1, y1, z1): z1=0
  For j= 1 to m
    i1 = (xi1, yi1, zi1): i2=(xi2, yi2, zi2): zi1=zi2=0
    if x1 = xi1 and x1 = xi2 then
      i1 ∈ dzlm: i2 ∈ dzlm : end
    elseif x1=xi1 or x1=xi2 then
      i1 ∈ dzlm: i2 ∈ dzlm: end
    end if
  bas=son +1
next j
next i

```

Şekil 9.15. dma izdüşümlerini bulma algoritması

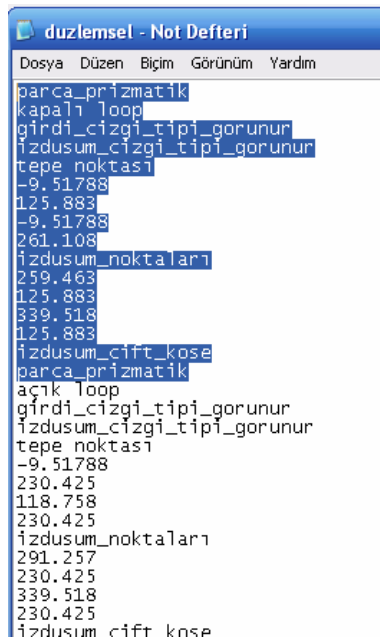
dma'nı oluşturan doğru veya yay parçalarının son noktaları bazı durumlarda komşu izdüşümlerin tek köşe noktasıyla eşleşebilirler. Bu tür parçalar, konik parçalardır (Şekil 9.16).



Şekil 9.16. *dma*'ların tek köşe noktası ile eşleşme durumu

dma'ların izdüşümleri bulunduğundan sonra elde edilen veriler veri tabanına kaydedilir. Bunlar (Şekil 9.17);

- izdüşümlerin bulunmasında kullanılan doğru parçasının köşe noktası (sn_1, \dots, sn_n) koordinatları
- ilgili *dma*'nın izdüşümlerinin köşe noktası koordinatları
- izdüşümün çizgi türü,
- *dma*'nın çizgi türü.
- *dma*'nın ait olduğu kenar küme grup ismi (açık veya kapalı küme).



Şekil 9.17. Veri tabanında kayıtlı *dma* verileri

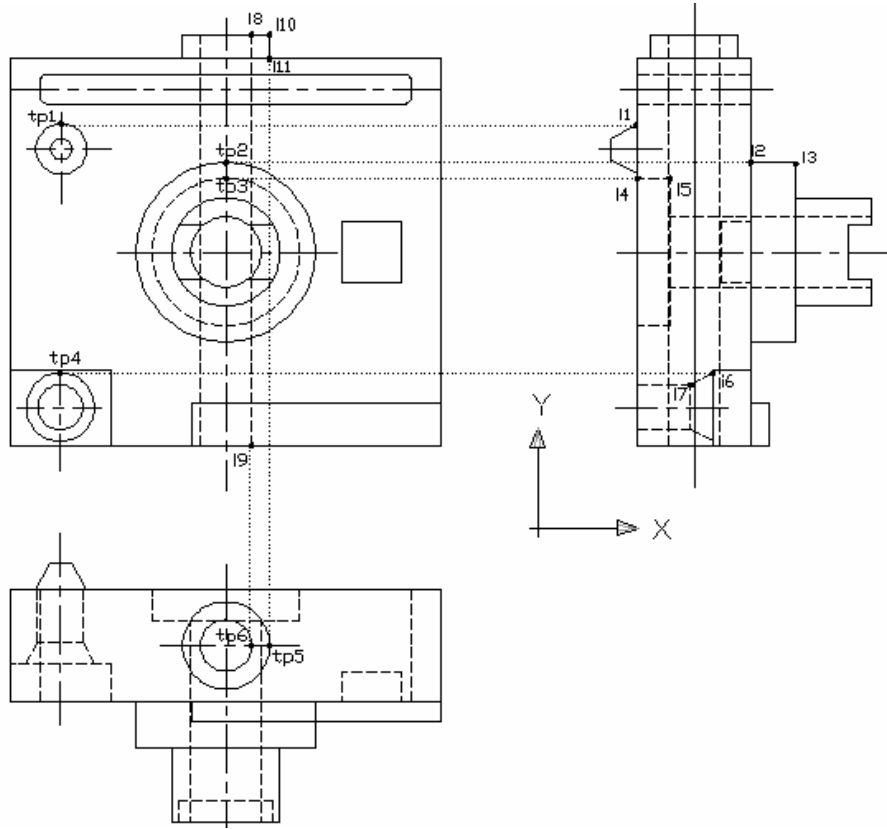
9.8.2. Silindirik model alanlarının (*sma*) izdüşümlerinin bulunması

sma izdüşümleri *dma* izdüşümlerinin bulunmasına benzer tarzda gerçekleştirilir. Burada, silindirik elemanın uç noktası ile aynı koordinat değerine sahip köşe noktaları aranmaktadır. Şekil 9.18’de ön görünüşte görülen *tp2* üst uç noktasına sahip *sma*’nın izdüşümü, *tp2* üst uç noktası ile aynı *y* koordinatına sahip *i2* ve *i3* köşe noktaları arasında kalan doğru parçasıdır. Benzer şekilde, üst görünüşte *tp5* yan uç noktasına sahip *sma*’nın ön görünüşteki izdüşümü, *tp5* yan uç noktası ile aynı *x* koordinatına sahip *i10* ve *i11* köşe noktaları arasında kalan doğru parçasıdır.

sma’nın tepe noktası *dma*’larda olduğu gibi tek köşe noktası ile de eşleşebilir. Bu tür parçalar konik çıkıntılar veya havşalı deliklerdir. Şekil 9.17’de *tp1* üst uç noktasına sahip *sma*’nın yan görünüşteki izdüşümünün sadece bir köşesi *tp1* ile aynı *y* koordinatına sahiptir. Bu durumda sadece tek köşe noktasının eşit olması durumu kabul edilmektedir. İzdüşüm köşe noktaları veri tabanına aktarılırken, tepe noktası ile eşleşen köşe noktasının ait olduğu doğru parçasının diğer köşe noktası da veri tabanına kaydedilir.

sma’lara ait izdüşümler bulunduktan sonra, izdüşüm verileri ve diğer biçim bilgileri veri tabanına aktarılır. Bu bilgiler aşağıdaki şekildedir:

- *sma*’nın tepe ve yan tepe nokta koordinatları (*tp1*...*tpn*),
- *sma* izdüşümlerinin köşe nokta koordinatları,
- izdüşümün çizgi türü,
- *sma*’nın çizgi türü.
- *sma*’nın 2B temel eleman türü (daire).



Şekil 9.18. Silindirik model alanlarının izdüşümleri

9.9. Süpürme Miktar ve Açılarını Hesaplama

Düzlemsel ve silindirik model alanların süpürme miktarları izdüşümleri boyu hesaplanarak bulunur. Süpürme açısı ise model alanının izdüşümünün yatayla yaptığı açıdır. Şekil 9.18’de $Sn1$ noktasının üzerinde bulunduğu model alanının yan görünüşteki izdüşümünün boyu aşağıdaki şekilde hesaplanır;

$i1$ ve $i2$ köşe noktaları dma’nın izdüşümüne ait köşe noktaları

$i1 = \{ (x_1, y_1, z_1) \}$; $i2 = \{ (x_2, y_2, z_2) \}$: $z_1 = z_2 = 0$ olmak üzere,

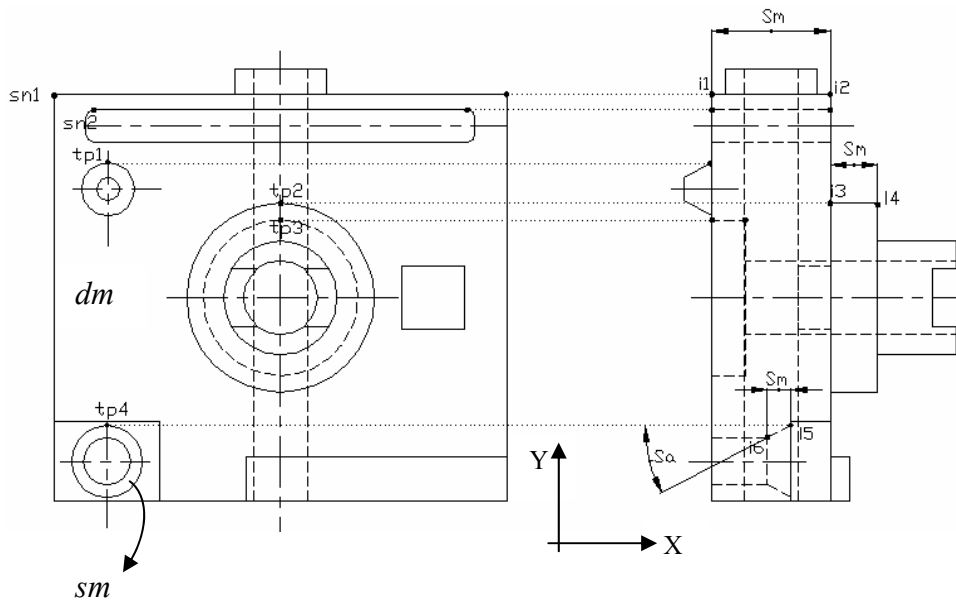
Süpürme miktarı (S_m);

$$S_m = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (9.1)$$

eşitliği ile hesaplanır.

Silindirik model alanlarının süpürme miktarları düzlemsel model alanları ile aynı şekilde hesaplanır. Şekil 9.19'da $tp4$ üst uç noktasının üzerinde bulunduğu sma 'nın süpürme miktarı eşitlik 9.2 yardımıyla hesaplanır. Fakat burada havşa başın oluşturulabilmesi için süpürmenin açılı yapılması gerekmektedir. Süpürme açısı (Sa) aşağıdaki eşitlik yardımıyla hesaplanır:

$$s_a = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right) \quad (9.2)$$



Şekil 9.19. Süpürme miktar ve açıları

9.10. Ana Katı Oluşturma

Model alanlarının süpürme miktar ve açıları belirlendikten sonra ana katılar oluşturulur. Ana katılar ana katı profillerinden oluşturulur. Ana katının oluşturulması için ana katı profilinin üzerinde bir seçim noktasının belirlenmesi, süpürme miktarı, açısı ve yönü gereklidir. Süpürme yönü dışındaki veriler daha önceki adımlarda elde edilip veri tabanına kaydedilmiştir. Burada, veri tabanında kayıtlı olmayan süpürme yönünün belirlenmesi gerekmektedir.

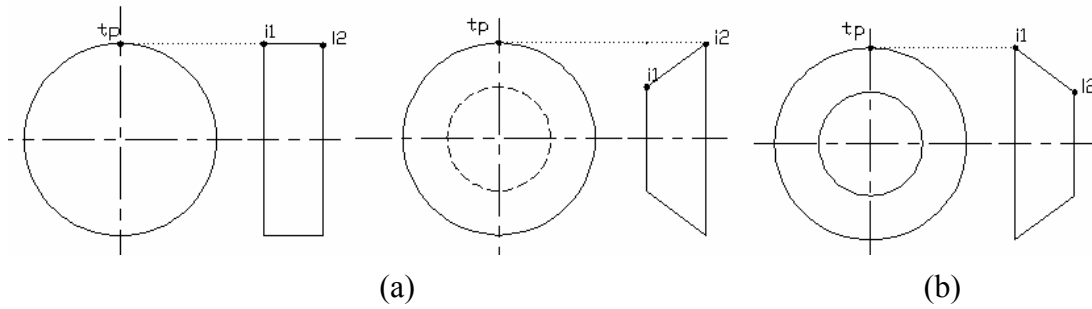
Şekil 9.20’de görülen ön görünüşteki 2B’lu girdi ve yan görünüşteki izdüşümleri aşağıdaki şekilde tanımlansın;

üst uç noktası: $tp = \{x_b, y_b, z_b\}; z_b=0\}$

İzdüşüm köşe noktaları: $i1=\{x_1, y_1, z_1\}; i2=\{x_2, y_2, z_2\}; z_1 = z_2 = 0$ olsun,

Ön ve yan görünüşte oluşturulacak ana katıların süpürme yönleri aşağıdaki kurallara göre belirlenir:

- 1) $y_1 \leq y_2$ ise süpürme yönü negatif (-) (Şekil 9.20.(a))
- 2) $y_1 > y_2$ ise süpürme yönü pozitif (+) (Şekil 9.20.(b))



Şekil 9.20. Ön ve yan görünüşte süpürme yönü belirleme:

- a) negatif yönde süpürme
- b) pozitif yönde süpürme

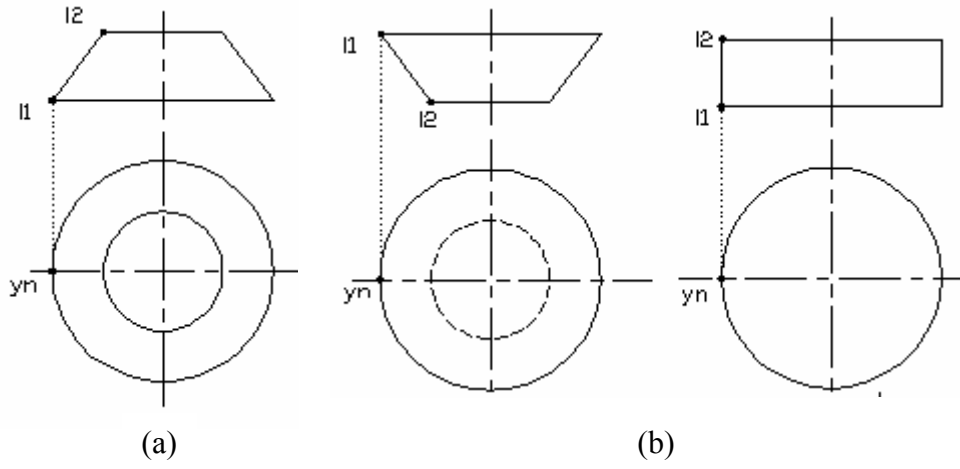
Şekil 9.21’de görülen üst görünüşteki 2B’lu girdi ve ön görünüşteki izdüşümleri aşağıdaki şekilde tanımlansın;

Yan uç noktası: $yn = \{x_y, y_y, z_y\}; z_y=0\}$

İzdüşüm köşe noktaları: $i1=\{x_1, y_1, z_1\}; i2=\{x_2, y_2, z_2\}; z_1 = z_2 = 0$ olsun,

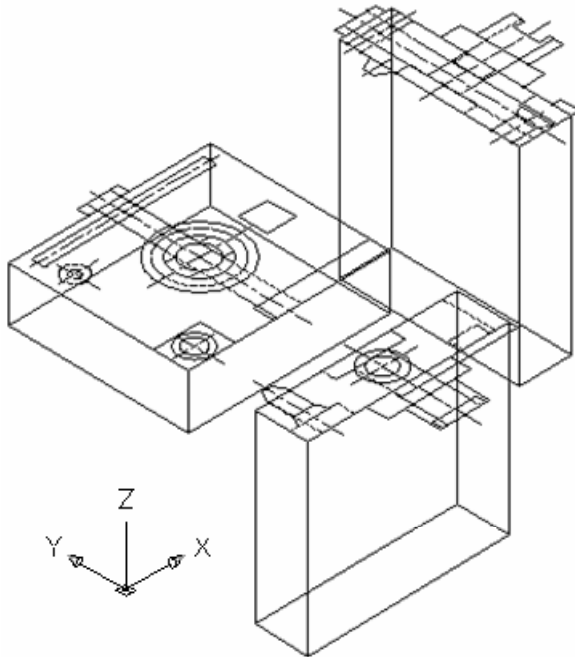
Üst görünüşte oluşturulacak ana katının süpürme yönleri aşağıdaki şekildedir:

- 1) $y_1 < y_2$ ve $x_1 \neq x_2$ ise, süpürme yönü pozitif (+) (Şekil 9.21.(a))
- 2) $y_1 < y_2$ ve $x_1 = x_2$ veya $y_1 > y_2$ ve $x_1 \neq x_2$ ise, süpürme yönü negatif (-) (Şekil 9.21.(b)).



Şekil 9.21. Üst görünüşte süpürme yönü belirleme
a) pozitif
b) negatif

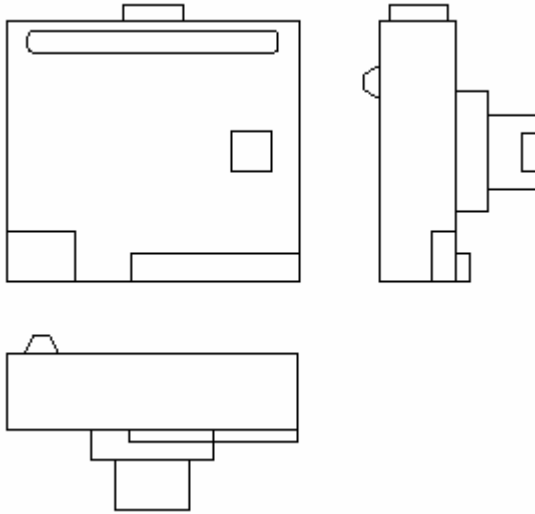
Süpürme miktarı, yönü ve açısı belirlenen ana katı profili, Şekil 9.22’de verilen koordinat ikonunda oklar normal vektör yönünü göstermek üzere, normal vektörü doğrultusunda (+) veya tersi yönünde (-) doğrusal süpürme işlemine tabi tutularak ana katılar oluşturulur (Şekil 9.21).



Şekil 9.22. Görünüş bölgelerinde oluşturulan ana katılar

dma'ların tekrar çizilmesi

Ana katılar oluşturulduktan sonra *dma* ve *sma*'lar tekrar çizilir. AKKA algoritması ile açık kenarları kapatılan açık kenar kümelerinin tekrar çizilmesi gerekmektedir. Tekrar çizim işlemi ekrandaki orijinal çizimin silinerek 2B izdüşümlerin tekrar oluşturulmasından ibarettir (Şekil 9.23). Burada, ilişkili kenar kümesine dahil olmayan çizgi ve yaylar tekrar çizilmez. Ayrıca yan ve üst görünüşteki, görünmez çizgilerle tanımlanan açık kenar kümeleri de tekrar çizilmez.



Şekil 9.23. *dma*'ların tekrar çizilmesi

Silindirik model alanlarının tekrar çizim işlemi *sma* ve *dma*'lardan US yardımıyla katı temel eleman oluşturma işlemi sırasında gerçekleştirilir. Çizim işlemi, VisualLISP programlama dili ile hazırlanan program parçası ile gerçekleştirilir. Program parçası, AutoCAD çizim çizim (line, circle, arc) komutlarını otomatik olarak çalıştırarak çizim işlemini gerçekleştirir.

9.11. Uzman Sistem Modülü

Katı temel elemanların oluşturulması aşamasında çok sayıdaki karmaşık işlemin sistematik ve daha kolay yapılabilmesi için yapay zeka tekniklerinden uzman sistem kullanılmıştır. US, pek çok karmaşık karar verme sürecinde güvenilir ve işlem hızını arttırıcı bir fonksiyon taşır. Bunun yanında karar verme adımlarında kullanılan karmaşık kural yapılarının daha sistematik bir yapıya kavuşmasını sağlar.

Klasik kural tabanlı sistemlerde kural yapısına müdahale edilemeyişi önemli bir kısıtlılıktır. US, program çalışırken dahi kural eklenebilen dinamik yapısı ile temel eleman biçimine karar verme adımlarında çözümü kolaylaştırıcı bir özelliğe sahiptir.

Bu sistemde US'un tercih edilmesinin en önemli sebebi, yazılıma müdahale edilmeden çözüm uzayının genişletilebilmesi imkânıdır.

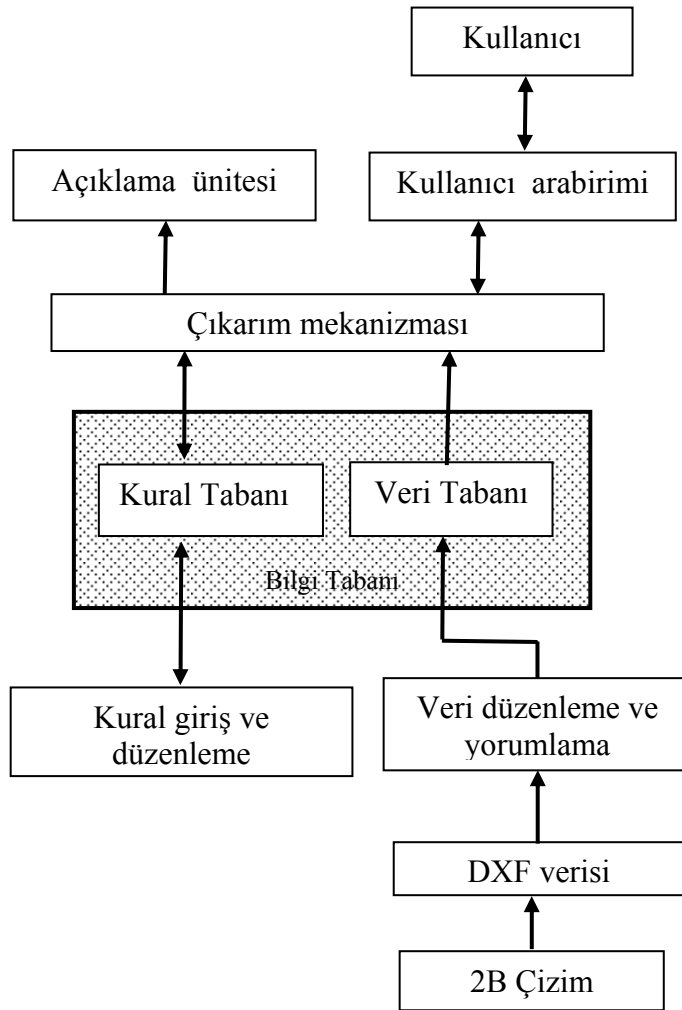
Katı model oluşturma algoritmasında US, tekrar çizim aşamasından sonra devreye girmektedir. Temel katı oluşturma aşamasında temel katının biçimine ve buna bağlı olarak boolean operasyonlarına karar verme işlemleri US modülü aracılığı ile gerçekleştirilir.

US karar destek birimi, ana katı oluşturma aşamasından görüşlerde ön katı oluşturma aşamasına kadar geçen katı model oluşturma sürecini kontrol eder ve yönlendirir (Şekil 9.24).

Bu bölümde geliştirilen uzman sistem destekli bir karar mekanizması ile temel elemanın türüne (delik veya çıkıntı), biçimine (düz veya konik) ve yapılacak boolean operasyonuna (ekleme veya çıkartma) karar verilmektedir. Geliştirilen US yaklaşımı, 2B görüşlerin yorumlanmasından 3B nesne oluşturma çalışmalarında ilk defa kullanılması yönüyle yenilik içermektedir.

US bilgi tabanında bilgiler kural tabanlı sunulmaktadır. Bilgi tabanı “eğer – O

halde” yapısı ile oluşturulmuştur. Eğer o halde yapısı bilginin sunulmasında en çok kullanılan yöntemdir. Bu yapıda kurallar şart cümleleri şeklindedir. *eğer – o halde* yapısı modüler yapısı, kural eklemeye uygun olması, şeffaf olması ve kolay değiştirilebilir olması yönleriyle çalışmada bilgi sunum tekniği olarak tercih edilmiştir. US Çıkarım mekanizması ise “*ileriye zincirleme*” yöntemi ile tasarlanmıştır. Şekil 9.24’te geliştirilen US modülünün genel yapısı görülmektedir.



Şekil 9.24. US modülünün genel yapısı

9.11.1. US veri tabanı

US veri tabanı, temel katı oluşturma aşamasında kullanılacak süpürme yönü, temel elemanın türü, temel elemanların 3B'lu koordinat bilgileri vb. bilgilerden oluşur. Bu bilgiler, programın daha önceki aşamalarında DXF formunda elde edilen verilerin düzenlenmesi ile oluşturulur. US veri tabanı aşağıdaki veri çeşitlerinden oluşur:

- katı temel eleman biçimi (*silindirik / prizmatik*)
- model alanı türü (*açık veya kapalı katı / dairesel*)
- model alanı görünüş bölgesi (*ön, yan, üst*)
- model alanı çizgi türü
- izdüşüm çizgi türü
- süpürme miktarı
- süpürme açısı
- temel katı izdüşümünün ana katı izdüşümüne göre konumu
- model alanı tepe veya son noktasının eşleştiği izdüşüm köşe sayısı (*tek veya çift*)
- model alanı izdüşüm boyu ile ana katı izdüşüm boyu karşılaştırması
- model alanının z koordinatı
- kesit durumu (*model alanında kesit olma hali*)

Burada belirtilen veriler şu iki çeşitte olabilir;

1. 2B'lu girdi bölgesi, çizgi tipi gibi AutoCAD ortamında tanımlanan sistem ve biçim bilgileri ve
2. bu verilerin yorumlanması ve bazı hesaplamalar sonucu elde edilen bilgilerdir.

Bu verilerin bir kısmı daha önceki işlem adımlarında oluşturulmaktadır (girdi ve izdüşüm çizgi tipi, süpürme miktarı, kesit durumu vb.). Diğer veriler, girdi görünüşü, izdüşümler ve daha önce oluşturulan verilerin yorumlanması sonucu elde edilirler.

Yukarıda verilen US veri tabanında kayıtlı veriler listesindeki süpürme açısından sonraki bilgiler bazı karşılaştırma ve hesaplamalar sonucu oluşturulmakta ve veri tabanına aktarılmaktadır.

Model alanlarının z koordinatları, süpürme yönleri gibi veriler model alanı ve ana katı profili izdüşümlerinin yorumlanması sonucu elde edilirler.

Model alanı “z” koordinat ve süpürme yönleri

Düzlemsel ve silindirik model alanlarından oluşturulan temel katı elemanların 3B düzlemdeki gerçek konumlarında olması gerekir. AutoCAD ortamında yapılan çizimlerde 2B girdi “ x, y ” koordinat çiftleri ile yapıldığı için, bütün model alanları aynı düzlemde görünür (Şekil 9.22). Bu durumda oluşturulacak katı temel elemana boolean operasyonu uygulamak yanlış model oluşmasına neden olur. Buna engel olmak için, her bir model alanına “ z ” koordinatları atanarak model alanı 3B’lu konumuna getirilmektedir.

Model alanlarına z koordinatı atama işlemi, ana katı profilinin biçimi ve model alanı izdüşümlerinin ana katı profilinin izdüşümü ile olan ilişkileri dikkate alınarak yapılır. Şekil 9.20 ve şekil 9.21’de görüldüğü gibi ana katı profili düz veya konik olabilir. Buna göre, Şekil 9.25’de görülen ana katı profilinin izdüşüm noktaları:

$$\begin{aligned}
 a_iz1 &= \{(ax_1, ay_1, az_1), a_iz2 = (ax_2, ay_2, az_2): az_1 = az_2 = 0\} \text{ ve} \\
 tp1 &= \{(xt_2, yt_2, zt_2), zt_2 = 0\}, \text{ tepe noktası ile ifade edil } sma \text{ 'nın izdüşüm noktaları,} \\
 t_iz1 &= \{(tx_1, ty_1, tz_1), t_iz2 = (tx_2, ty_2, tz_2): tz_1 = tz_2 = 0\} \\
 tp &= \{(xt_1, yt_1, zt_1), zt_1=0\}, \text{ tepe noktası ile ifade edilen } sma \text{ 'nın izdüşüm noktaları,} \\
 t_iz1 &= \{(tx_1, ty_1, tz_1), t_iz2 = (tx_2, ty_2, tz_2): tz_1 = tz_2 = 0\} \\
 sn1 \text{ ve } sn2 &\text{ son noktalarının üzerinde bulunduğu } dma \text{ 'nın izdüşüm noktaları,} \\
 t_iz1 &= \{(tx_1, ty_1, tz_1), t_iz2 = (tx_2, ty_2, tz_2): tz_1 = tz_2 = 0\} \text{ olsun,}
 \end{aligned}$$

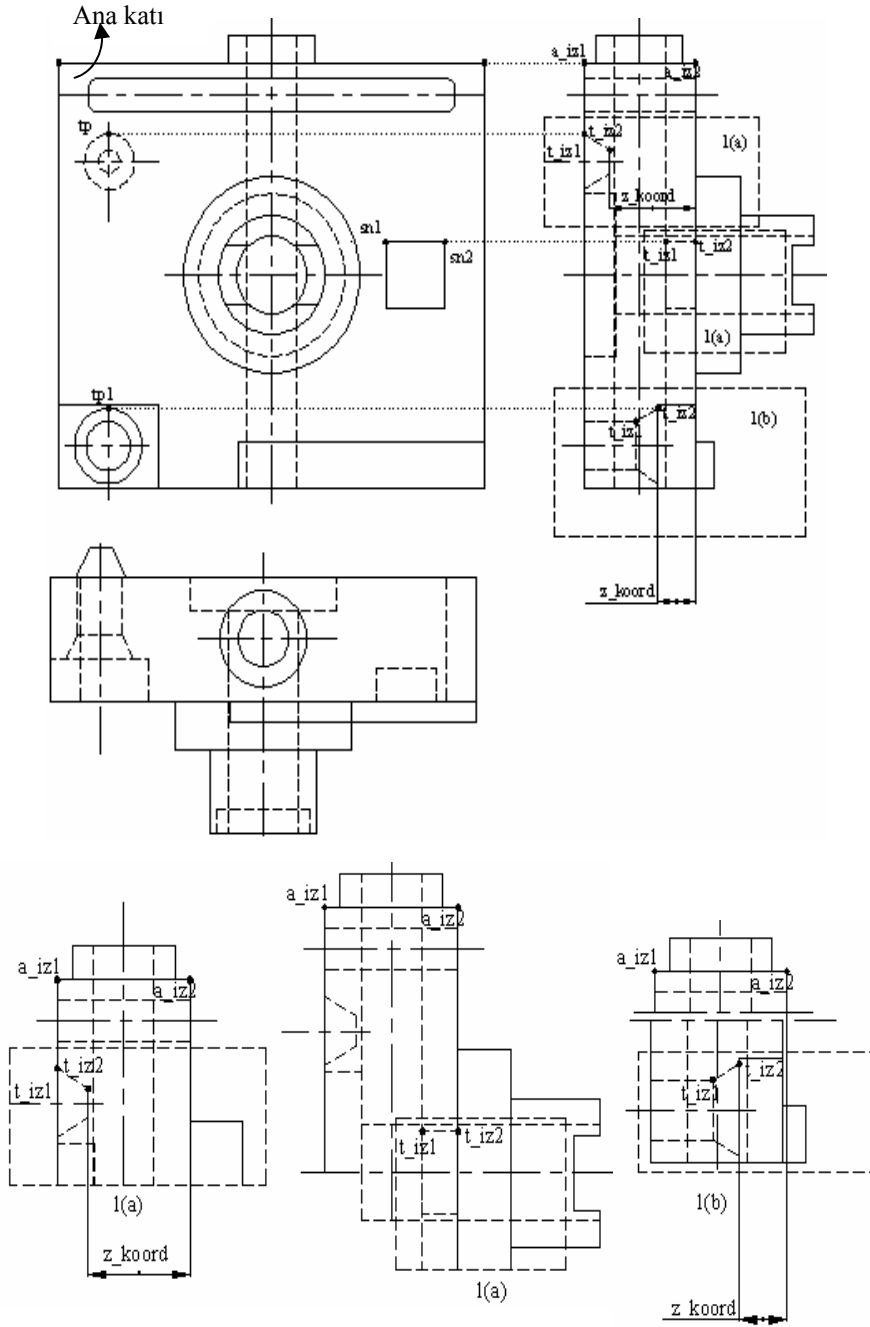
1. Eğer ana katının izdüşümü düz veya sola konik ise (Şekil 20.(a))

$$(ay_1 \leq ay_2)$$

Temel elemanların z koordinatları, süpürme miktar ve yönleri aşağıdaki şekilde hesaplanır:

a) Şekil 9.25'te (1a) olarak gösterilen durumda da görüldüğü gibi ise;

$ty_1 \leq ty_2$ ise $z_koord = tx_2 - ax_2$ olur ve süpürme yönü negatif (-) alınır.



9.25. Model alanı z koordinat değerlerini belirleme

b) Şekil 9.25'te (1b) olarak gösterilen durumda da görüldüğü gibi ise,
 Eğer $ty_1 > ty_2$ ise $z_koord = tx_1 - ax_1$ olur ve süpürme yönü pozitif (+) alınır.

2. Şekil 9.26'da tp üst uç noktasının üzerinde bulunduğu ana katı profili izdüşüm noktaları,

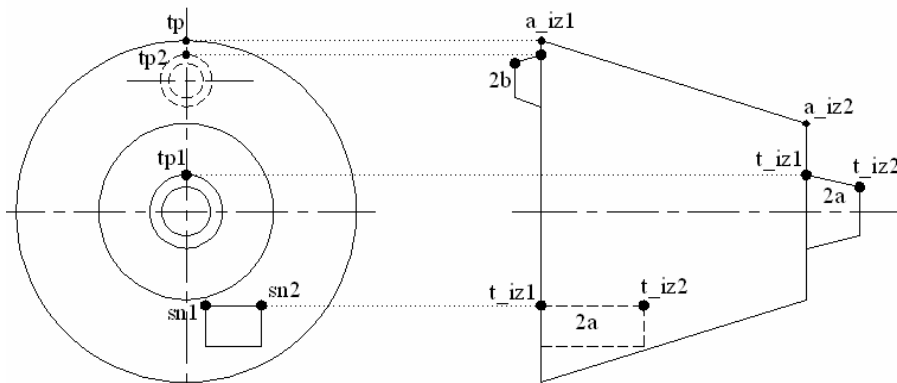
$a_iz1 = \{(ax_1, ay_1, az_1), a_iz2 = (ax_2, ay_2, az_2): az_1 = az_2 = 0: ay_1 > ay_2\}$ olarak tanımlı ise temel elemanların z koordinatları, süpürme miktar ve yönleri aşağıdaki şekilde belirlenir:

a) Şekil 9.26'da (2a) olarak gösterilen durumda da görüldüğü gibi,

$ty_1 \geq ty_2$ ise $z_koord = tx_1 - ax_1$ olur ve süpürme yönü pozitif (+) alınır.

b) Şekil 9.26'da (2b) olarak gösterilen durumda da görüldüğü gibi,

Eğer $ty_1 < ty_2$ ise $z_koord = tx_2 - ax_1$ olur ve süpürme yönü negatif (-) alınır.



Şekil 9.26. Model alanı z koordinat değerlerini belirleme

Model alanı izdüşümünün ana katı izdüşümüne göre konumu

Temel katı elemanın biçimine karar vermede önemli kriterlerden birisi de silindirik veya prizmatik model alanının izdüşümünün, ana katı izdüşümüne göre konumudur.

Burada belirlenen konum temel elemanın çıkıntı olup olmadığına karar vermede önemli bir veridir. Model alanı izdüşümünün ana katı izdüşümüne göre üç farklı konumu tanımlanmıştır:

- 1) ana katı sınırları içinde
- 2) ana katıya soldan çıkıntı
- 3) ana katıya sağdan çıkıntı

ana katı sınırları içinde: Şekil 9.25'teki model alanı ve ana katı profili tanımlamaları ışığında model alanı izdüşümünün ana katı sınırları içerisinde sayılabilmesi için aşağıdaki şartları sağlamalıdır:

“ izdüşüm çizgi tipi görünmez” ve

$tx_1 \geq ax_1$ ve $tx_2 = ax_2$ veya $tx_2 < ax_2$ ise, *“izdüşüm ana parça sınırları içinde”*

ana katıya soldan çıkıntı: Şekil 9.25'teki model alanı ve ana katı profil tanımlamalarına göre model alanı izdüşümünün ana katıya soldan çıkıntı olarak kabulü için aşağıdaki şartları sağlamalıdır:

“ izdüşüm çizgi tipi görünür” ve

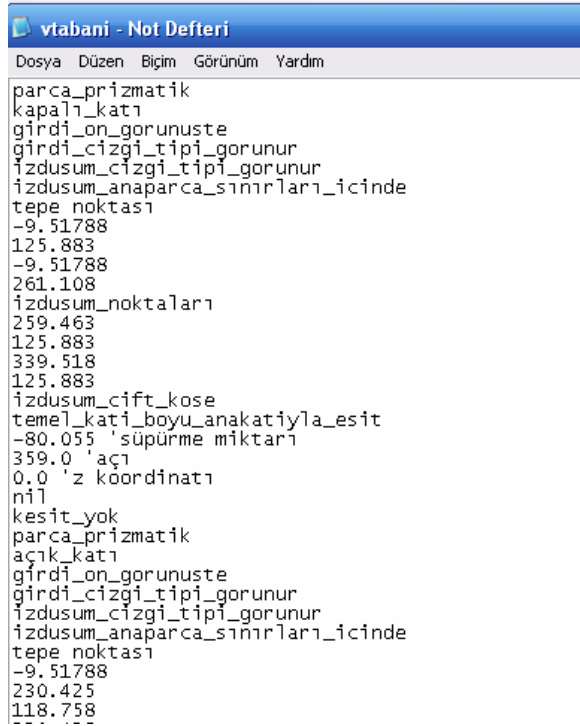
$tx_2 \leq ax_1$ ve $tx_1 < ax_1$ ise, *“izdüşüm ana parçaya soldan çıkıntı”*

ana katıya sağdan çıkıntı: Şekil 9.25'teki model alanı ve ana katı profil tanımlamalarına göre model alanı izdüşümünün ana katıya sağdan çıkıntı olarak kabulü için aşağıdaki şartları sağlamalıdır:

“ izdüşüm çizgi tipi görünür” ve

$tx_1 \geq ax_2$ ve $tx_2 > ax_2$ ise, *“izdüşüm ana parçaya sağdan çıkıntı”*

US veri tabanı tüm bu bilgilerin elde edilmesi sonucu oluşturulur. Veri tabanına kullanıcı tarafından doğrudan veri girişi yoktur. 2B'lu girdi görünüşüne ve izdüşümüne ait bilgilerin yorumlanması sonucu oluşturulur (Şekil 9.27).



```

vtabanı - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
parca_prizmatik
kapalı_katı
girdi_on_gorunuste
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sınırları_ıcinde
tepe_noktası
-9.51788
125.883
-9.51788
261.108
izdusum_noktaları
259.463
125.883
339.518
125.883
izdusum_cift_kose
temel_kati_boyu_anakatiyla_esit
-80.055 'süpürme miktarı
359.0 'açı
0.0 'z koordinatı
nil
kesit_yok
parca_prizmatik
açık_katı
girdi_on_gorunuste
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sınırları_ıcinde
tepe_noktası
-9.51788
230.425
118.758

```

Şekil 9.27. US veri tabanında kayıtlı veriler

9.11.2. Kural tabanı

Kural tabanı, temel katı eleman biçimlerine karar vermek amacıyla hazırlanan kurallardan oluşur. Kurallar, temel katı elemana ait veri tabanında kayıtlı bilgilerini kullanarak çıkarım mekanizması aracılığı ile katı temel elemanın biçimine karar vermede kullanılır.

Bir temel elemanın biçimi silindirik veya prizmatik olabilir. Silindirik veya prizmatik biçimli bir temel eleman delik veya çıkıntı olabilir. Bu delik ve çıkıntılar ise düz veya konik olabilir. Kural tabanındaki tanımlamalar, temel elemanın bu sınırların hangisine dahil olduğunu belirlemeye yardımcı olur. Bu tanımlamalar aşağıdaki şekildedir:

- katı temel eleman biçimi (*silindirik / prizmatik*)
- model alanı türü (*açık veya kapalı katı / dairesel*)

- model alanı görünüş bölgesi (*ön, yan, üst*)
- model alanı çizgi tipi
- izdüşüm çizgi tipi
- temel katı izdüşümün ana katı izdüşümüne göre konumu
- kesit durumu (*model alanının kesitle ifade hali*)
- model alanı tepe veya son noktasının eşleştiği izdüşüm köşe sayısı (*tek veya çift*)
- model alanı izdüşüm boyu ile ana katı izdüşüm boyu karşılaştırması

Kurallar “*eğer – o halde*” yapısı ile oluşturulmuştur. Bu yapı, anlaşılabilirlik ve sisteme esneklik katması yönü ile tercih edilmektedir. *Eğer – o halde* yapısında kural ekleme, çıkarma gibi düzenleme işlemleri daha kolay bir şekilde gerçekleştirilmektedir.

Kuralların oluşturulması

US’te kurallar, konuyla ilgili uzman bilgisi veya temel kaynak olan kitaplardan edinilen bilgilerle oluşturulur. Kural cümlesini oluşturan kriterlerin belirlenmesi ve bu kriterlerle ilgili hükümler uzman bilgisi ve değişik kaynaklar dikkate alınarak oluşturulmuştur. Uzman bilgisi konuyla ilgili mesleki tecrübe ve bilgi birikiminden oluşmaktadır. Uzman bilgisi yanında kullanılan kaynaklar aşağıdaki şekilde sayılabilir:

1. Teknik resimle ilgili temel kurallar
2. Daha önceki çalışmalar sonucu ortaya çıkan tecrübeler.
3. Problemlerle ilgili tanımlanan özel durumlar

Teknik resimle ilgili temel kurallar: Bu tür kurallar, görünüş ve izdüşüm ilişkileri ile ilgili kurallardır. Bunlar teknik resimle ilgili temel prensiplerden meydana gelir. Aşağıda verilen kriterler örnek olarak verilebilir:

- Bir deliğe ait izdüşümün kesit görünüşlerde görünür çizgilerle, kesit alınmayan durumlarda görünmezle ifade edilmesi,
- Bir çıkıntının izdüşümünün görünür olması.

Daha önceki çalışmalar sonucu ortaya çıkan tecrübeler: Bunlar, konuyla ilgili daha önceki çalışmalar sonucu ortaya konulan bilgilerdir. Kural cümlelerindeki bu tür kriterlere aşağıdaki örnekler verilebilir:

- İzdüşümlerin belirlenmesinde ortaya konan kriterler: Bir görünüşe ait izdüşümler bulunurken kullanılan ortak koordinat eksen bilgisi,
- İzdüşüm köşe sayısına göre temel elemanın biçimini belirleme: İzdüşüm tek köşe ise katı temel elemanın konik, çift köşe ise düz olması durumu.

Problemlerle ilgili özel durumlar: Tez çalışması çerçevesinde yeni tanımlanan durumlardan oluşturulan kriterler. Bunlar;

- Bir 2B temel eleman kümesinin açık ve kapalı kenar küme olma durumu
- 2B temel eleman türüne göre temel eleman türünü tespit etme
- İzdüşüm boy ve konumuna göre çıkıntı olup olmama durumunu belirleme

şeklinde belirtilebilir. Tüm bu kaynaklar yardımıyla tespit edilen kriterler çerçevesinde oluşturulan kural zincirinin hangi temel elemanı tanımlayacağı ile ilgili nihai karar uzman tarafından,

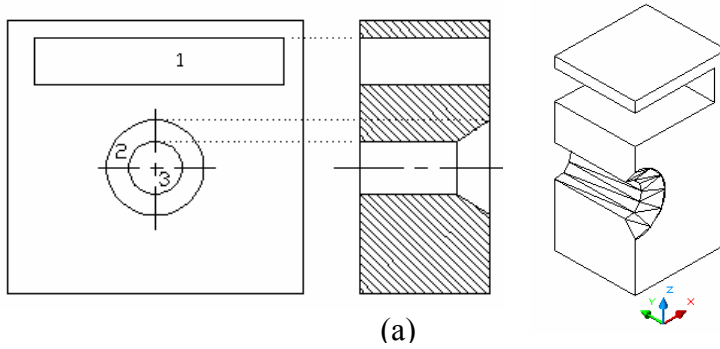
- Mesleki bilgi ve tecrübesi ve
- 3B nesne tanıma ile ilgili geçmiş çalışmalar sonucunda ortaya çıkan bilgiler çerçevesinde karar verilmektedir.

US'in bilgi tabanındaki diğer işlemler (kural ekleme, çıkartma, düzenleme vb.) tamamıyla uzman kullanıcı tarafından gerçekleştirilmektedir.

Şekil 9.28.(a)'da verilen 2B görünüş ve izdüşümlerinin temsil ettiği katı temel eleman türleri 9.28.(b)'de verilen örnek kurallar kullanılarak belirlenebilir. Buna göre;

1 numaralı eleman düzlemsel model alanına sahip olduğu için parça türü prizmatik tanımlanmaktadır, model alanı kapalı kenar kümesi olduğu için model alanı türü kapalı katı olarak tanımlanır. Girdi görünüşü ön görünüşte, girdi çizgi türü görünür, izdüşüm çizgi türü görünür olarak tanımlanmıştır. Parça kesit görünüşle ifade edilmiş ve izdüşümü eş iki y koordinat değerine sahiptir (izdüşüm iki köşe). Bu tanımlamalar ışığında bir numaralı parça boydan boya prizmatik delik olarak tanımlanır. Eğer kesit alınmamış olmasaydı bu elemanın delik olarak tanımlanabilmesi için izdüşümünün görünmez kenarlarla ifade edilmesi gerekirdi.

2 ve 3 numaralı parçalar silindirik model alanına sahip olduğu için kural tabanında silindirik parça olarak tanımlanır. Burada parça türü ve model alanı dışında bütün tanımlamalar benzerdir. İki numaralı parça tek izdüşüm köşe noktasına sahip olduğu için havşa başlı delik olarak tanımlanır.



(a)

Kural_n (1 numaralı parça)**EĞER** parca tipi = parca_prizmatik

ve model alanı türü = kapalı_katı

ve girdi görünüş bölgesi= girdi_on_gorunuste

ve model alanı çizgi tipi= girdi_cizgi_tipi_gorunur

ve izdüşüm çizgi tipi= izdusum_cizgi_tipi_gorunur

ve izdüşümün konumu= izdusum_anaparca_sınırları_icinde

ve kesit durumu = kesit_var

ve izdüşüm köşe sayısı = izdusum_cift_kose

ve temel ana katı boyu = temel_kati_boyu_anakatiya_esit

İSE katı temel eleman = boydanboya_prizmatik_delik**Kural_n (2 numaralı parça)****EĞER** parca tipi = parca_silindirik

ve model alanı türü = daire

ve girdi görünüş bölgesi= girdi_on_gorunuste

ve model alanı çizgi tipi= girdi_cizgi_tipi_gorunur

ve izdüşüm çizgi tipi= izdusum_cizgi_tipi_gorunur

ve izdüşümün konumu = izdusum_anaparca_sınırları_icinde

ve kesit durumu = kesit_var

ve izdüşüm köşe sayısı = izdusum_tek_kose

ve temel ana katı boyu = temel_kati_boyu_anakatidan_kucuk

İSE katı temel eleman = silindirik_kor_havsali_delik**Kural_n (3 numaralı parça)****EĞER** parca tipi = parca_silindirik

ve model alanı türü = daire

ve girdi görünüş bölgesi= girdi_on_gorunuste

ve model alanı çizgi tipi= girdi_cizgi_tipi_gorunur

ve izdüşüm çizgi tipi= izdusum_cizgi_tipi_gorunur

ve izdüşümün konumu = izdusum_anaparca_sınırları_icinde

ve kesit durumu = kesit_var

ve izdüşüm köşe sayısı = izdusum_cift_kose

ve temel ana katı boyu = temel_kati_boyu_anakatidan_kucuk

İSE katı temel eleman = silindirik_kor_delik

(b)

Şekil 9.28. Kural tabanından örnek kurallar:

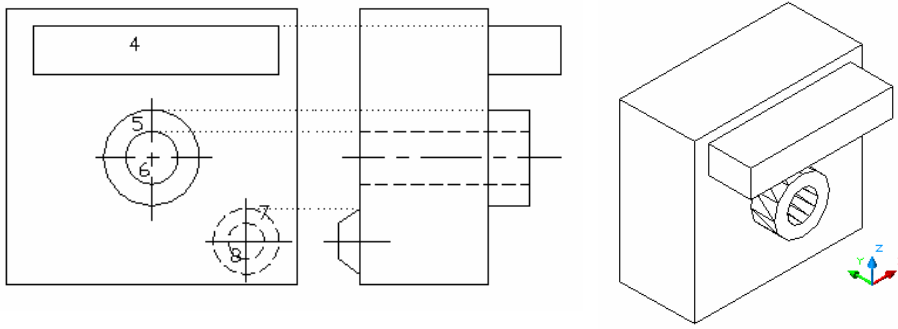
a) 2B görünüş ve izdüşümleri

b) Kurallar

Bir temel elemanın çıkıntı olarak tanımlanabilmesi için ana katıya göre konumu belirleyicidir. Şekil 9.29.(a)'de 5 ve 7 numaralı silindirik model alanlarının izdüşümlerinin konumu ana parça izdüşümüne göre solda ve sağda oldukları için çıkıntı olarak tanımlanırlar. Kesit alınmayan parçalarda bir elemanın çıkıntı unsuru olarak tanımlanabilmesi için model alanı izdüşümünün görünür olması gereklidir. Kesit alınmayan parçalarda bir temel elemanın delik unsuru olarak tanımlanabilmesi için ise 6 numaralı elemenda olduğu gibi, model alanı izdüşümünün çizgi türünün görünmez olması gerekmektedir.

7 numaralı model alanında olduğu gibi girdi görünüşü görünmez olduğu durumlarda temel eleman, görünür çizgi türüne sahip elemanlar gibi oluşturulur. Bir elemanın girdi görünüşünün çizgi türü onun konumunu belirlemektedir. Unsur türünde belirleyici olan izdüşümün çizgi türü, ana parçaya göre konumu gibi kriterlerdir. Girdi görüntüsü görünmez çizgilerle oluşturulan model alanlarından oluşturulan temel elemanların z koordinat değerleri temel elemanın 3B'lu düzlemdeki yerinin belirlenmesinde en önemli parametredir.

US kural tabanında yukarıda belirtilen kriterler çerçevesinde 570 kural geliştirilmiştir. US kural tabanı, uzman kullanıcının ihtiyaç duyduğu anda yeni kurallar ekleme ve düzenleyebilmesine imkân sağlayacak şekilde tasarlanmıştır. Ek – 2'de geliştirilen kural cümlelerinden örnekler verilmiştir.



(a)

Kural_n (4 numaralı parça)**EĞER** parca tipi = parca_prizmatik

ve model alanı türü = kapalı_katı

ve girdi görünüş bölgesi= girdi_on_gorunuste

ve model alanı çizgi tipi= girdi_cizgi_tipi_gorunur

ve izdüşüm çizgi tipi= izdusum_cizgi_tipi_gorunur

ve izdüşümün konumu= izdusum_anaparcaya_sagdan_cikinti

ve kesit durumu = kesit_yok

ve izdüşüm köşe sayısı = izdusum_cift_kose

ve temel ana katı boyu = temel_kati_boyu_anakatidan_kucuk

İSE katı temel eleman = prizmatik_cikinti**Kural_n (5 numaralı parça)****EĞER** parca tipi = parca_silindirik

ve model alanı türü = daire

ve girdi görünüş bölgesi= girdi_on_gorunuste

ve model alanı çizgi tipi= girdi_cizgi_tipi_gorunur

ve izdüşüm çizgi tipi= izdusum_cizgi_tipi_gorunur

ve izdüşümün konumu = izdusum_anaparcaya_sagdan_cikinti

ve kesit durumu = kesit_yok

ve izdüşüm köşe sayısı = izdusum_cift_kose

ve temel ana katı boyu = temel_kati_boyu_anakatidan_kucuk

İSE katı temel eleman = silindirik_cikinti**Kural_n (7 numaralı parça)****EĞER** parca tipi = parca_silindirik

ve model alanı türü = daire

ve girdi görünüş bölgesi= girdi_on_gorunuste

ve model alanı çizgi tipi= girdi_cizgi_tipi_gorunmez

ve izdüşüm çizgi tipi= izdusum_cizgi_tipi_gorunur

ve izdüşümün konumu = izdusum_anaparcaya_sagdan_cikinti

ve kesit durumu = kesit_yok

ve izdüşüm köşe sayısı = izdusum_tek_kose

ve temel ana katı boyu = temel_kati_boyu_anakatidan_kucuk

İSE katı temel eleman = silindirik_havsalı_cikinti

(b)

Şekil 9.29. Kural tabanından örnek kurallar:

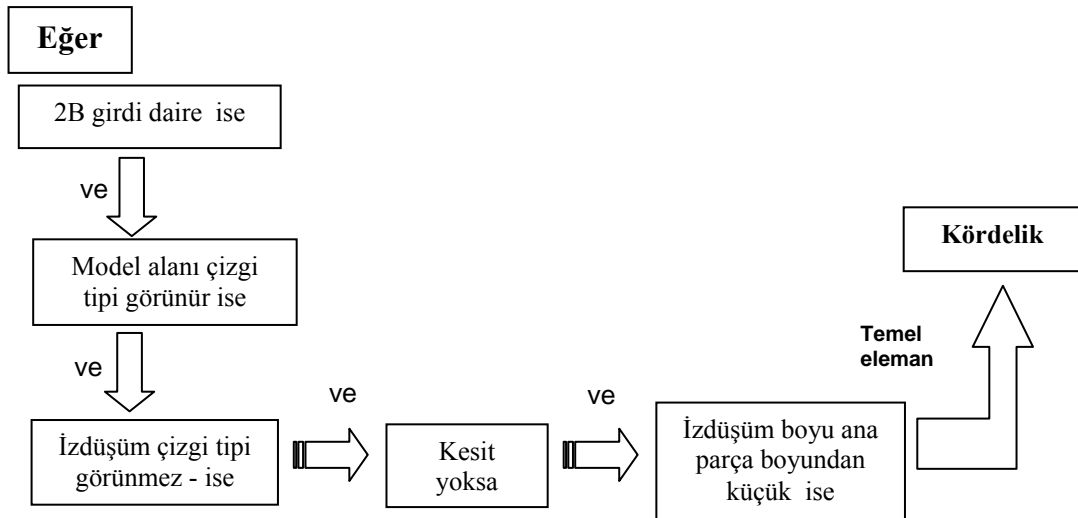
a) 2B görünüş ve izdüşümleri

b) Kurallar

9.11.3. Çıkarım mekanizması

US çıkarım mekanizması, veri tabanından aldığı veriyi kural tabanındaki kurallarla değerlendirerek sonuç üretir. Çıkarım mekanizması, model alanlarının izdüşüm ilişkileri ve fiziksel biçimlerinin yorumlanması ile oluşturulan verilerden yola çıkarak katı temel elemanın biçimine karar vermektedir. Bu karar, katı temel elemanın oluşturulmasında ve ana katı ile arasında yapılacak boolean operasyonunun belirlenmesinde kullanılmaktadır.

Çıkarım mekanizması genelde iki teknikle oluşturulur. Bunlar; ileri ve geriye zincirleme metotlarıdır. Tanıtılan çalışmada çıkarım mekanizması “*ileriye zincirleme*” yöntemi ile tasarlanmıştır. İleriye zincirleme metodu bilinen verilerden başlar ve uygun kural bulununca bu kuralın mevcut şartlarını tatmin etmeyi temel alır (Şekil 9.30).




















Şekil 9.30. İleriye zincirleme metoduyla katı temel eleman biçimi belirleme

Çıkarım mekanizması veri tabanındaki bilgi gruplarını alır, kural tabanındaki kurallarla bu bilgileri değerlendirir. Değerlendirme ilk kuraldan başlar sırayla tüm kurallar incelenerek sonuca gidilir. Veri tabanındaki model alanı bilgileriyle eşleşen kural çözüm olarak seçilir (Şekil 9.30).

Çizelge 9.1’de çıkarım mekanizmasının işleyiş yapısı örnek temel eleman türlerine göre gösterilmiştir. Burada çıkarım mekanizmasının bilgi tabanından aldığı veri ve kuralları değerlendirerek yaptığı çıkarımlar görülmektedir. Çıkarım mekanizması, veri tabanından aldığı veriyi kural tabanındaki kurallarla karşılaştırmaktadır. Bütün kurallar, çıkarım mekanizması tarafından sırayla değerlendirilerek uygun çözüm bulunmaktadır.

Çizelge 9.1. Uzman sistem bilgi tabanı ve örnek çıkarımlar

Bilgi Tabanı (Veri tabanı – Kural tabanı ilişkileri)																			Çıkarım
Görünüş			Girdi türü		Model alanı çizgi türü		İzdüşüm çizgi türü		Kesit durumu		İzdüşüm köşe sayısı		Model alanı İzdüşümü ana katı izdüşümüne göre			Temel katı boyu ana katıdan			Katı temel eleman
Ön	Yan	Üst											İçerde	Sağda	solda	>	<	=	
✓			✓		✓			✓		✓	✓		✓				✓		Prizmatik kör delik
	✓			✓	✓			✓		✓		✓	✓					✓	Silindirik havşalı delik
✓			✓		✓		✓		✓		✓		✓					✓	Prizmatik boydanboya delik
✓				✓		✓	✓			✓		✓			✓	✓			Silindirik havşalı çıkıntı
	✓			✓	✓			✓	✓		✓			✓				✓	Silindirik boydanboya delik
	✓		✓			✓	✓			✓	✓				✓		✓		Prizmatik çıkıntı
		✓		✓	✓		✓		✓		✓		✓				✓		Silindirik kör delik

 düzlemsel görünür kesit yok iki köşe silindirik görünmez kesit var tek köşe

Çıkarım mekanizması yardımıyla elde edilen sonuçlar ayrı bir veri tabanına aktarılır. Şekil 9.31’da, çıkarım mekanizması ile biçimine karar verilen 3B’lu temel elemanla ilgili veriler görülmektedir.

3bbilgi - Not Defteri				
Dosya	Düzen	Biçim	Görünüm	Yardım
kural_53 prizmatik_kor_delik açık_katı -9.51788 239.269 118.758 239.269 izdusum_noktaları 291.644 239.269 339.905 239.269 80.056 359 -80.056 "nil" kural_38 prizmatik_cikinti kapalı_katı 14.2358 216.415 97.3809 216.415 izdusum_noktaları 224.918 216.415 259.849 216.415 -34.931 359 34.931 "nil" kural_38 prizmatik_cikinti kapalı_katı 37.5643 202.674				

Şekil 9.31. US çıkarım mekanizması sonuç verileri

9.12. Katı Temel Elemanlar Oluşturma

Katı temel elemanların US yardımıyla elde edilmesinden sonra, model alanları doğrusal süpürme işlemi ile katı temel elemanlara dönüştürülürler. Bu işlem için gerekli veriler katı temel elemanın süpürme miktar, yön, açısı ve model alanı üzerinde bir seçim noktasıdır. Bu veriler US çıkarım mekanizması ile üretilen

sonuçların kaydedildiği veri tabanında bulunmaktadır. Buradaki veriler önce prizmatik, daha sonra silindirik veriler sırasıyla alınarak işlem gerçekleştirilir.

9.12.1. Prizmatik katı temel elemanların oluşturulması

Prizmatik katı temel elemanlar, *dma*'ların normal vektörleri veya normal vektörlerinin tersi yönünde süpürülmesiyle oluşturulur. İşlem, model alanı üzerinde belirlenen seçim noktası aracılığı ile model alanının seçimi ve algoritmanın daha önceki adımlarında elde edilen süpürme miktar, yön ve açısı ile model alanının katı temelemana dönüştürülmesidir. Model alanlarının katı temelemana dönüştürülmesinde AutoCAD “*extrude*” komutu kullanılmaktadır. Extrude komutu, bir seçim işlemi ve belirlenen uzunluk yardımıyla model alanlarını katı temel elemanlara dönüştürmektedir. Bu işlem, VisualLISP ortamında yazılan program parçası ile otomatik olarak gerçekleştirilmektedir.

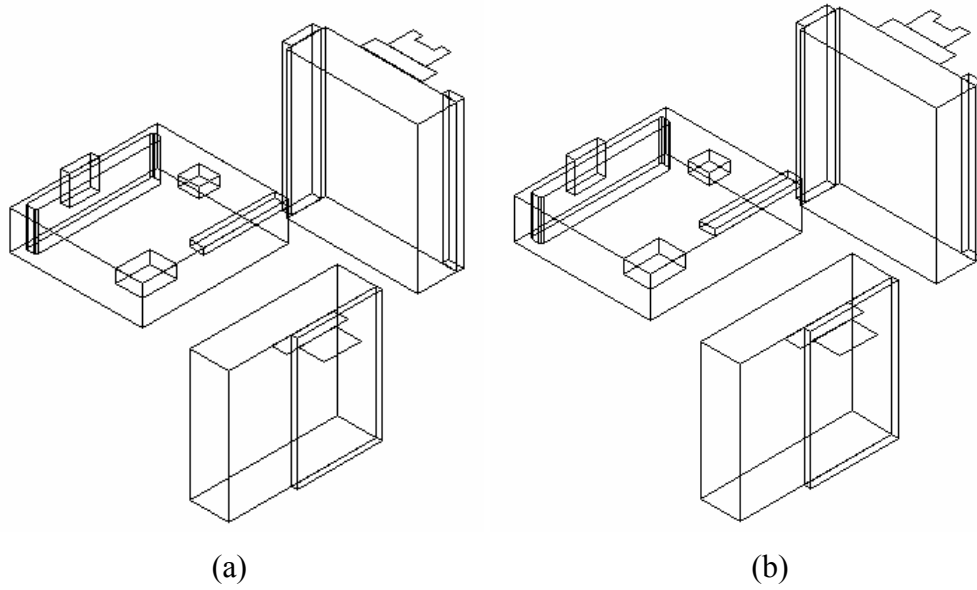
Ön görünüşteki açık ve kapalı kenar kümelerinden oluşturulan model alanlarının hepsinden katı temel eleman oluşturulur. Yan ve üst görünüşte ise durum farklıdır. Bu görünüşlerde görünmez kenarlarla ifade edilen açık kenar kümelerinden oluşturulan model alanlarından katı temel eleman oluşturulmaz. Delik unsurları kapalı kenar kümesi olarak tanımlandıkları görünüşlerde oluşturulmaktadır. Delikleri bütün görünüşlerde oluşturmak mümkündür fakat bu gereksiz işlemlere neden olacaktır.

Çıkıntı unsurları ise bütün görünüşlerde oluşturulmaktadır. Burada delik unsurları gibi bir yaklaşım benimsendiği takdirde hacimsel kesiştirme sonrası yanlış çıkıntıların oluşur. Hacimsel kesiştirme ortak hacmi katı model olarak verdiği için, çıkıntılar diğer görünüşlerde oluşturulmazsa ortak hacim içerisinde yer almayacaktır. Fakat deliklerde böyle bir durum söz konusu değildir.

Yan ve üst görünüşlerde açık kenar kümelerinden oluşturulan katı temel elemanların süpürme miktarları ana katı ile aynı alınmaktadır. Hacimsel kesiştirmede ortak alan

kapalı kenar kümesi ile oluşturulan katı temel eleman olduğu için, yan ve üst görünüşlerdeki çıkıntı unsurlarının boyu ana katı boyu ile aynı alınmıştır. Bu yöntem 3B koordinata taşıma işlem sayısı azaltılması yönüyle de avantajlıdır.

Şekil 9.32’de ön, yan ve üst görünüşte oluşturulan prizmatik katı temel elemanlar görülmektedir.



Şekil 9.32. Görünüş düzlemlerde prizmatik katı temel elemanlar:

- a) ana katı ile aynı düzlemde
- b) 3B konumuna getirilmiş halde

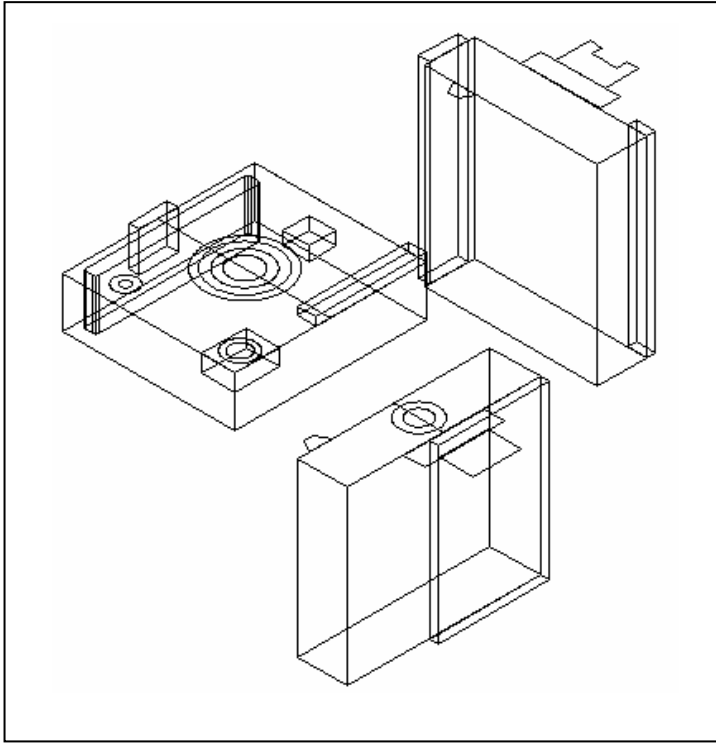
Katı temel elemanlar oluşturulduktan sonra 3B düzlemdeki konumlarına getirilmesi gerekmektedir. Bu işlem daha önce hesaplanan z koordinatları kullanılarak gerçekleştirilir. Prizmatik katı temel elemanlar, z koordinatları kadar taşınarak 3B’lu düzlemdeki konumlarına getirilirler.

9.12.2. Silindirik katı temel elemanların oluşturulması

Silindirik katı temel elemanlar prizmatik katı temel elemanlardan sonra oluşturulur. Silindirik katı temel eleman prizmatik katı temel elemanlara benzer şekilde silindirik model alanının normal vektörü doğrultusunda veya tersi yönde süpürülmesi sonucu oluşturulur.

Burada öncelikle silindirik model alanının tekrar çizimi yapılmaktadır. Silindirik elemanlar, 3B'lu koordinat düzlemlerine taşıma işlemini ortadan kaldırmak için doğrudan 3B'lu düzlemlerinde çizilirler. Bunun için *sma*'nı oluşturan daire merkez koordinatına z koordinatı eklenmektedir.

Daire merkez kordinat çiftine z koordinatı eklenmeden yapılan çizimde *sma*'ları ana katı yüzeyinde kalacaktır (Şekil 9.33). Bu durumda oluşturulan katı temel elemanlara boolean operasyonları uygulandığı takdirde yanlış model oluşturulur.

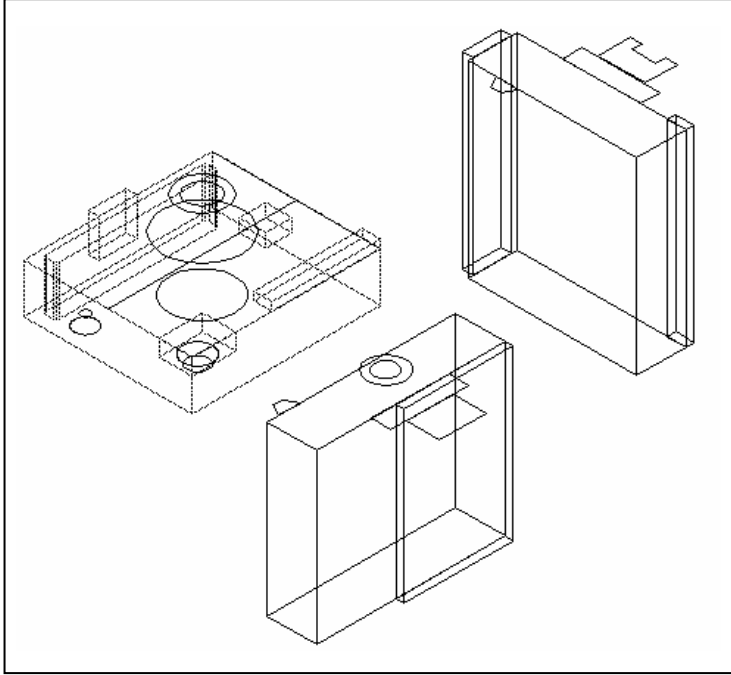


Şekil 9.33. *sma*'ların tekrar çizimi ($z = 0$ hatalı çizim)

Daire merkez kordinat çiftine z koordinatı eklendiği takdirde *sma* 3B'lu düzlemdeki gerçek konumunda olacaktır. Bu durumda oluşturulan katı temel elemanlara boolean operasyonları uygulandığı takdirde doğru model oluşturulur. Ayrıca, prizmatik katı temel elemanlarda yapılan taşıma işlemine gerek kalmaz (Şekil 9.34).

Sma'lardan oluşturulan silindirik delik unsurları sadece *sma*'nın bulunduğu görünüş

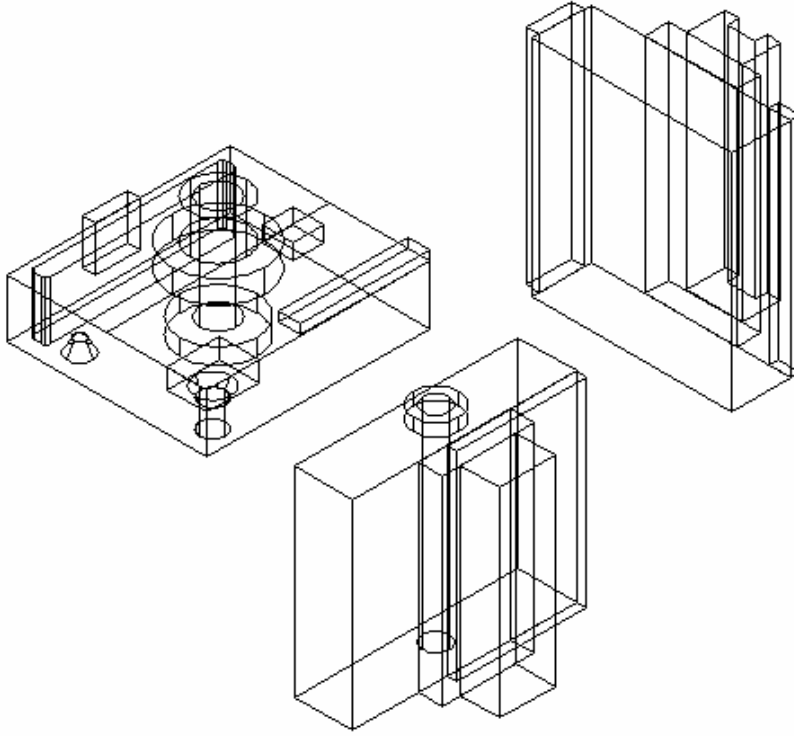
düzleminde oluşturulur. Delik elemanlarının izdüşümlerinden katı temel eleman oluşturulmaz.



Şekil 9.34. 3B konumlarında oluşturulmuş silindirik model alanları

Silindirik çıkıntıları oluşturan *sma*'ların durumu prizmatik katı temel elemanlara benzer bir yapı içerir. Bu elemanların izdüşümlerinin kapalı bölgelerinden oluşturulan *dma*'lar bulundukları görünüşteki ana katı boyunca süpürme işlemine tabi tutulurlar. *sma* izdüşümlerinden oluşturulan katı temel elemanın süpürme yönü üzerinde bulunduğu ana katıyla aynı yöndedir. Silindirik katı temel elemanlar, prizmatik katı temel elemanlara benzer bir şekilde AutoCAD “extrude” komutu ile otomatik olarak oluşturulmaktadır.

Şekil 9.35’de ön yan ve üst görünüşte oluşturulan temel ve ana katılar görülmektedir.

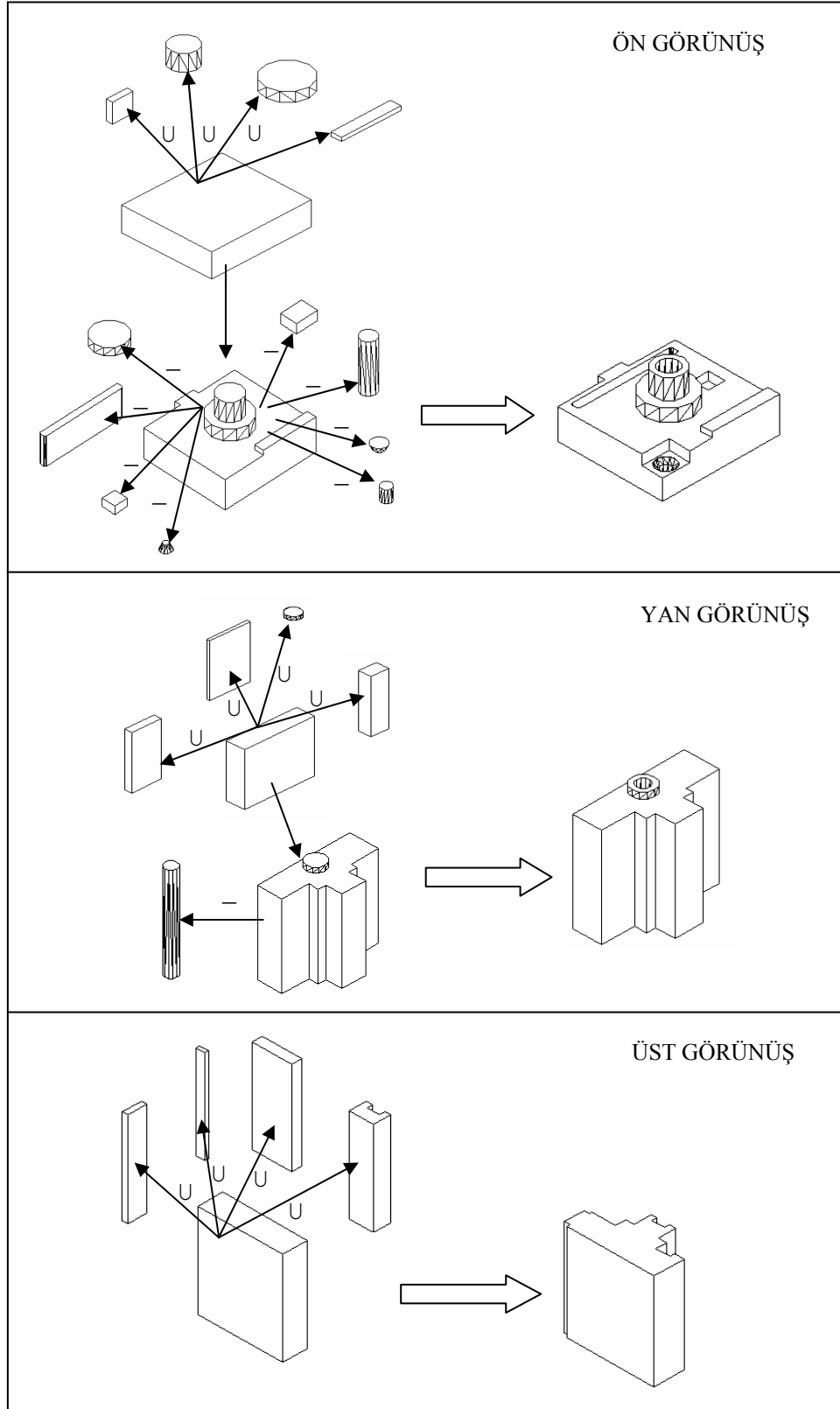


Şekil 9.35. Görünüş bölgelerinde oluşturulan katı temel elemanlar

9.13. Görünüş Bölgelerinde Ön Katılar Oluşturma

Ön katılar ana ve temel katılar arasında yapılan boolean operasyonları sonucu oluşturulur. Boolean operasyonları katı temel elemanın türüne göre değişir. Katı temel eleman bir çıkıntı unsuru ise bu eleman ana katı ile birleştirilir. Katı temel eleman delik unsuru ise bu eleman ana katıdan çıkartılır. Katı temel elemanların biçim bilgileri US çıkarım mekanizması ile elde edilip veri tabanına aktarılmıştır. Program, bu bilgilerden yararlanıp boolean operasyonlarını gerçekleştirmektedir. Boolean operasyonları AutoCAD “union” ve “subtract” komutları yardımıyla yapılmaktadır.

Boolean operasyonları sonrasında her bir görünüş bölgesinde birer tane ön katı oluşturulmaktadır (Şekil 9.36).



Şekil 9.36. Görünüş bölgelerinde ön katılar oluşturma

9.14. Hacimsel Kesiřtirme ve Katı Model

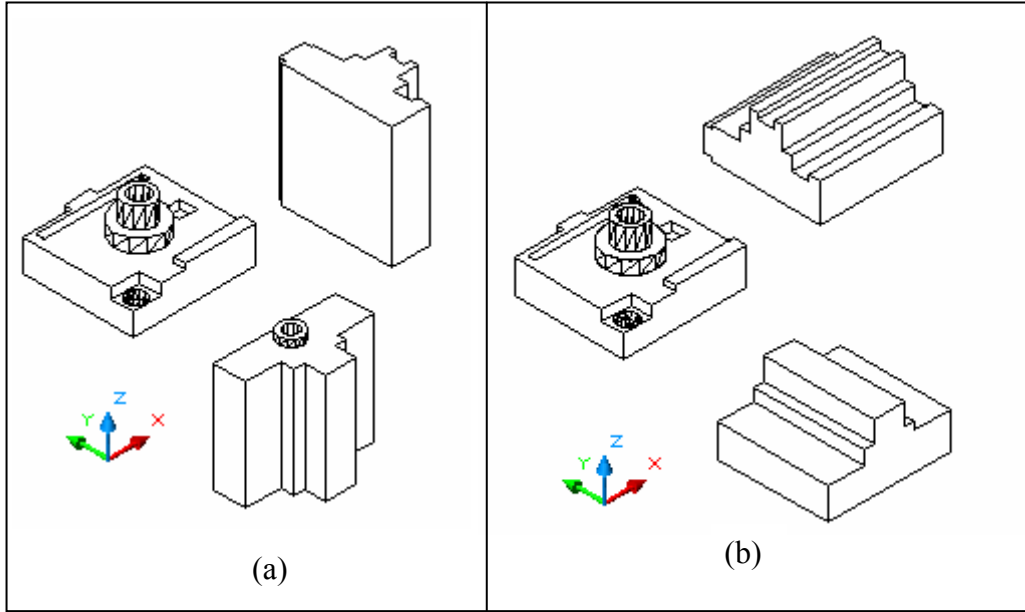
Görünüş düzlemlerinde ön katılar oluşturulduktan sonra hacimsel kesiřtirme işlemi yapılır. Hacimsel kesiřtirme sonucu ön katıların ortak hacmi belirlenir. Burada belirlenen ortak hacim katı modeli verir. Hacimsel kesiřtirme işlemi öncesinde ön katılara döndürme ve taşıma operasyonları uygulanır.

9.14.1. Ön katıları döndürme

Yan ve üst görünüşte oluşturulan ön katılar döndürülürler. 2B’lu izdüşümler 1. izdüşüm metoduna göre oluşturulmakta, AutoCAD ekranına $x-y$ bakış doğrultusuna çizilmektedir. Gerçekte, ön görünüş $x-z$, yan görünüş $y-z$, üst görünüş $x-y$ koordinat çiftleriyle tanımlıdır. AutoCAD’de bütün görünüşler $x-y$ koordinat çiftleri ile oluşturulduğu için girdi görünüşleri olması gereken koordinat düzleminde değildir. Bu durumda yapılacak taşıma ve hacimsel kesiřtirme işlemi yanlış katı model oluşumuna neden olacaktır.

Ön katıları kendi koordinat düzlemlerine yerleřtirme işlemi ilgili görünüşlerin döndürülmesi ile gerçekleştirilir. Döndürme işlemi önce yan ve üst görünüşlerde, hacimsel kesiřtirme sonrasında ön görünüşte yapılır.

Bu sebeple yan görünüşte oluşturulan ön katı, “y” eksenini etrafında ve saat akrebi tersi yönünde 90 derece; üst görünüşteki katı ise, “x” eksenini etrafında ve saat akrebi yönünde 90 derece döndürölmektedir (Şekil 9.37). Döndürme işlemi AutoCAD “Rotate3d” komutuyla yapılmaktadır. Rotate3d komutu, nesneyi bir döndürme noktası etrafında, x , y veya z ekseninde istenilen açıda döndürmektedir.



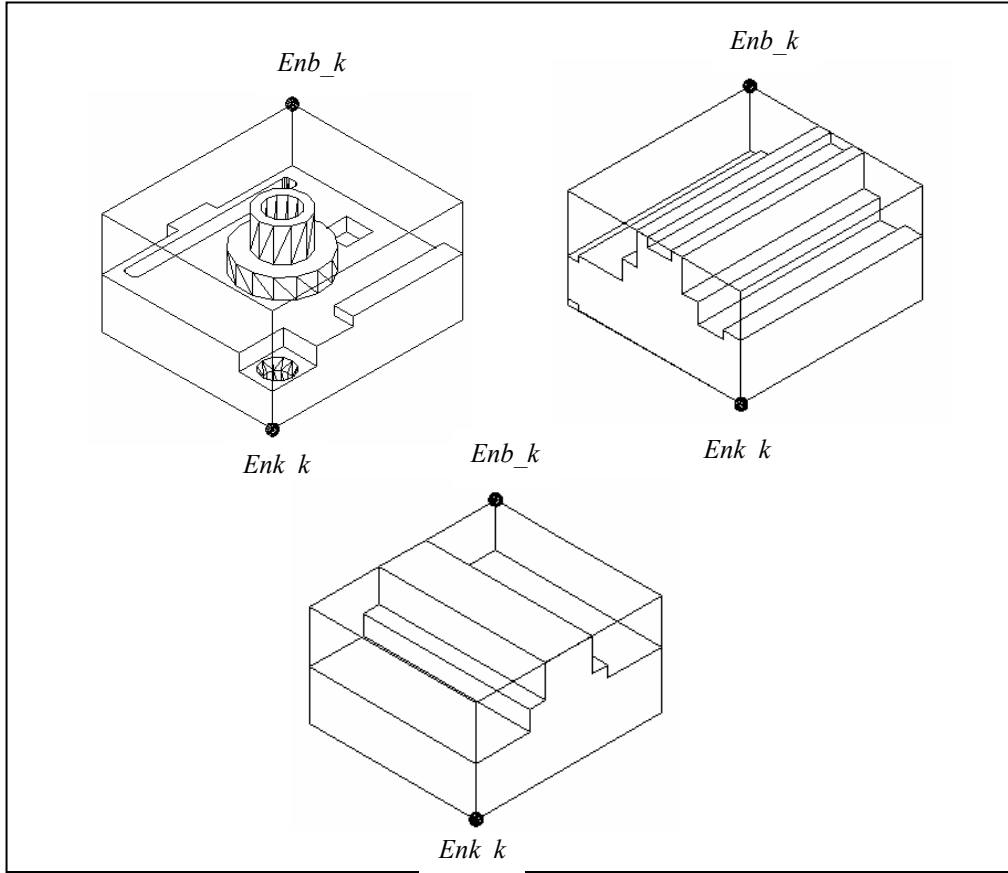
Şekil 9.37. Ön katılar (a) normal halde (b) döndürme sonrası

9.14.2. Ön katıların taşınması

Bu aşamada ön katılar birbirlerinin üzerine gelecek şekilde taşınmaktadır. İşlem, yan ve üst görünüş bölgelerinde oluşturulan ön katıların ön görünüşteki ön katı üzerine taşınmasıdır. Taşıma için, taşıma noktası önemlidir. Katılar yanlış nokta üzerine taşınırsa hacimsel kesiştirme hatalı sonuç verir.

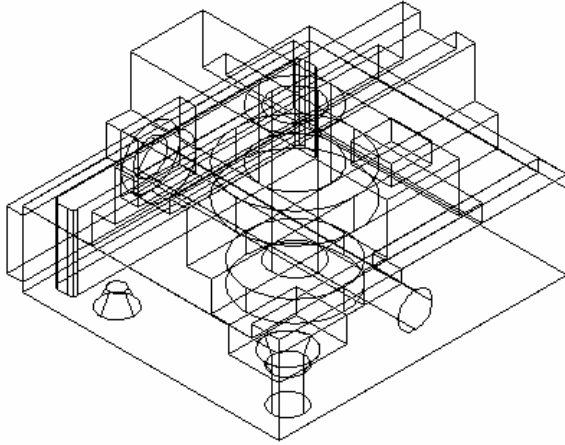
Taşıma ön katıların ortak bir referans noktası üzerine yapılır. Bu nokta, her üç katının en küçük x , en küçük y ve en küçük z koordinatına sahip köşe noktalarıdır. AutoCAD komutlarından *Massprop* (kütle özellikleri) komutu ile katıların bazı kütle özelliklerine ulaşılabilmektedir. Bunlar hacim, kütle, ağılık merkezi, atalet momentleri ve sınır noktaları şeklinde sayılabilir. Ön katıların taşınmasında kullanılan referans noktası sınır noktalarından elde edilir.

Sınır noktaları, katı modelin uç noktalarından geçen hayali bir 3B kutu düşünülerek oluşturulur. En küçük x , y ve z köşe noktaları (enk_k) ile en büyük x , y ve z köşe noktaları (enb_k) bu hayali kutunun sınır noktalarıdır (Şekil 9.38).



Şekil 9.38. Ön katıların en büyük ve en küçük sınır noktaları

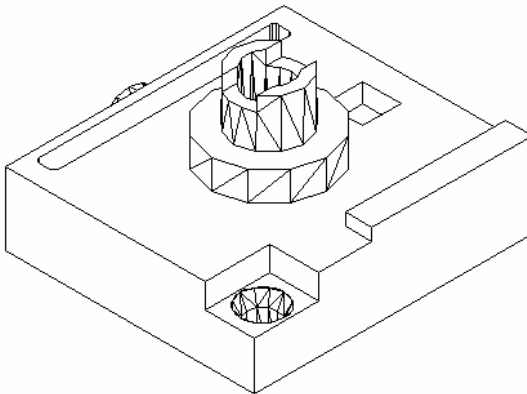
Ön katıların en küçük ve en büyük sınır noktaları belirlendikten sonra, yan ve üst görünüşte oluşturulan ön katılar, ön görünüşte oluşturulan ön katının üzerine en küçük sınır noktaları referans olacak şekilde taşınırlar (Şekil 9.39). Taşıma işlemi AutoCAD “Move” komutu ile gerçekleştirilmektedir.



Şekil 9.39. Ön katıların taşınması

9.14.3. Hacimsel kesiştirme

Bu aşamada ön görünüşte birleştirilen ön katılar kesiştirme işlemine tabi tutulur. Hacimsel kesiştirme, iki veya daha fazla mevcut bölgenin üst üste binen alanını ve iki veya daha fazla mevcut katının ortak hacmini hesaplar ve bulduğu alan ya da ortak hacmi tek bir nesne olarak oluşturur (Şekil 9.40). Kesiştirme işlemi AutoCAD “intersect” komutunun program içinde çalıştırılması ile gerçekleştirilir. Intersect komutu, üst üste gelmiş iki veya daha fazla katının ortak hacmini belirleyerek son katıyı oluşturmaktadır.



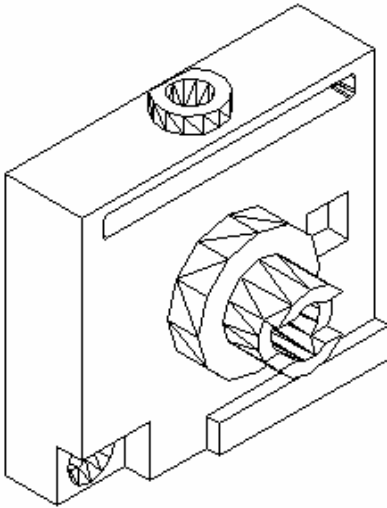
Şekil 9.40. Hacimsel kesiştirme sonrası katı model

9.15. Katı Model

Bu aşamada hacimsel kesiştirme işlemi ile oluşturulan katı model üzerinde son düzenlemeler yapılır. Bu düzenlemeler katı modeli gerçek bakış doğrultusuna getirmek için döndürme ve ekranda kalması muhtemel fazla çizgileri silmedir.

Katı model x ekseninde saat akrebi yönünde 90 derece döndürülerek 3B'lu bakış doğrultusuna getirilir. Bu işlem sonunda 3B'lu model, girdi izdüşüm yöntemi olarak benimsenen birinci izdüşüm yöntemine göre gerçek bakış doğrultusuna getirilmiş olur (Şekil 9.41).

Grafik ekranda kalan 2B'lu çizgi, yay ve daireleri silmek için katı model oluşturulan yeni bir katmana atılır. Daha sonra bu katman kapatılarak ekran temizlenir. Daha sonra katının bulunduğu katman açılır ve işlem tamamlanır. Bütün bu işlemler sonunda 3B katı model elde edilmiş olur.



Şekil 9.41. Döndürme sonrası katı model

10. AUTOMODEL PROGRAMININ KULLANIMI

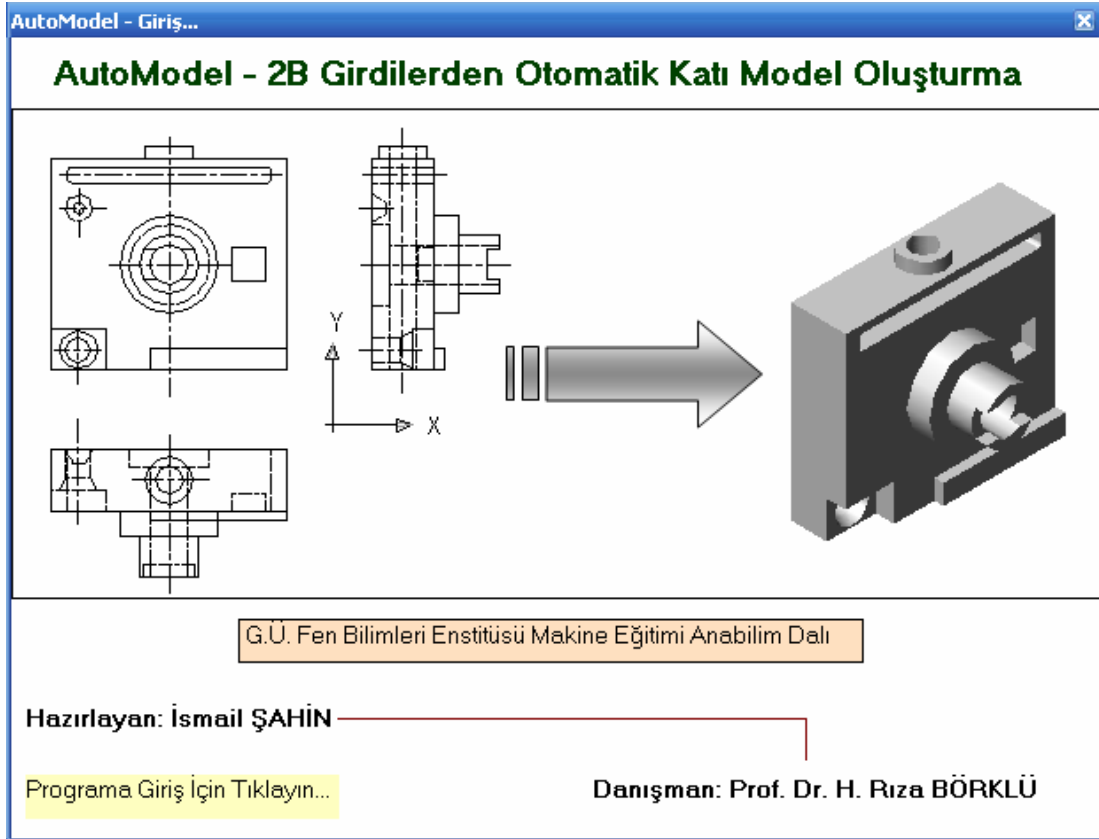
Çalışma kapsamında, araştırma sonuçlarının test dildiği bir bilgisayar programı hazırlanmıştır. Bilgisayar programı, Visual BASIC 6.0 ve VisualLISP programlama dilleri ile yazılmıştır. Visual BASIC, döngü yapısındaki anlaşılabilirlik, veri işlemede sunduğu kolaylıklar, nesneye yönelik programlar hazırlamaya uygun yapısı ve AutoCAD uygulamalarında kullanılabilmesi yönüyle tercih edilmiştir.

VisualLISP 2B'lu görünüşlerin tekrar çizimi, model alanı, katı temel elemanlar ve katı model oluşturma işlemlerinde kullanılmıştır. VisualLISP, AutoCAD içindeki çizim ve bazı basit analiz işlemlerini gerçekleştirmeye yönelik olarak geliştirilen bir programlama dilidir. VisualLISP AutoCAD komutlarını, AutoCAD çizim mantığında kullanabilmektedir. Bu durum VisualLISP'in AutoCAD fonksiyonlarını esnek bir tarzda gerçekleştirmesini sağlamaktadır. VisualLISP'in bu özelliği çalışmada tercih edilmesindeki en önemli nedendir.

Bu iki programlama dilinde hazırlanan ve AutoModel olarak adlandırılan bilgisayar programı, AutoCAD ortamında çizilen 2B'lu parça resimlerini yorumlayarak katı modellerinin oluşturulması aşamalarına klavuzluk etmektedir. Hazırlanan programla, ekrandaki çizimin DXF formatında kaydı, DXF verilerinin düzenlenerek program içerisinde kullanılabilir halde düzenlenmesi, düzenli verilerden model alanı bilgisi oluşturma, model alanlarından katı temel elemanlar oluşturma ve son olarak katı model oluşturma aşamaları kontrol edilmektedir.

10.1. Programa Giriş

Programın açılışında ekrana programa giriş penceresi gelmektedir. Giriş penceresi üzerinde bulunan *“programa giriş için tıklayınız”* seçeneği ile program ana menüsüne geçilir. Giriş penceresinde program hakkında tanıtıcı bilgiler bulunmaktadır (Şekil 10.1).

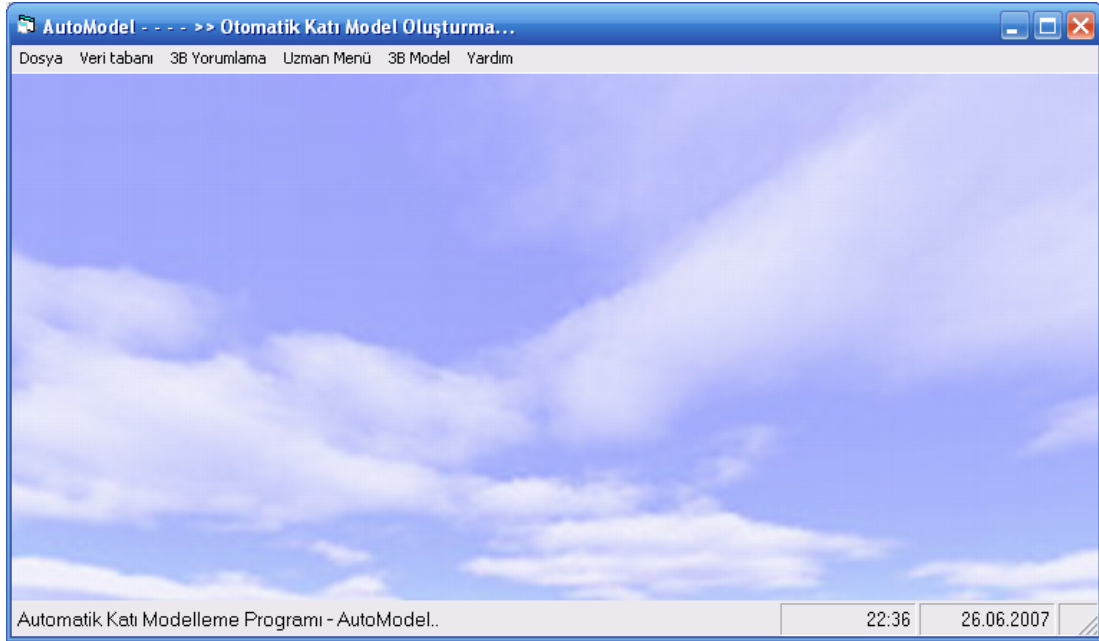


Şekil 10.1. Programa giriş ekranı

10.2. Ana Menü

Ana menü programın yönlendirildiği menüdür. 2B görünüş girdisi oluşturma sonrasında katı model oluşturma aşamasına kadar bütün süreç ana menü üzerindeki menü çubuğu aracılığı ile yönlendirilir (Şekil 10.2).

Menü çubuğu üzerinde bulunan dosya, veri tabanı, 3B yorumlama, Uzman menü, 3B model ve yardım çek menüleri aracılığı ile katı model oluşturma aşamaları kontrol edilmektedir.



Şekil 10.2. AutoModel programı ana menüsü

10.2.1. Dosya menüsü

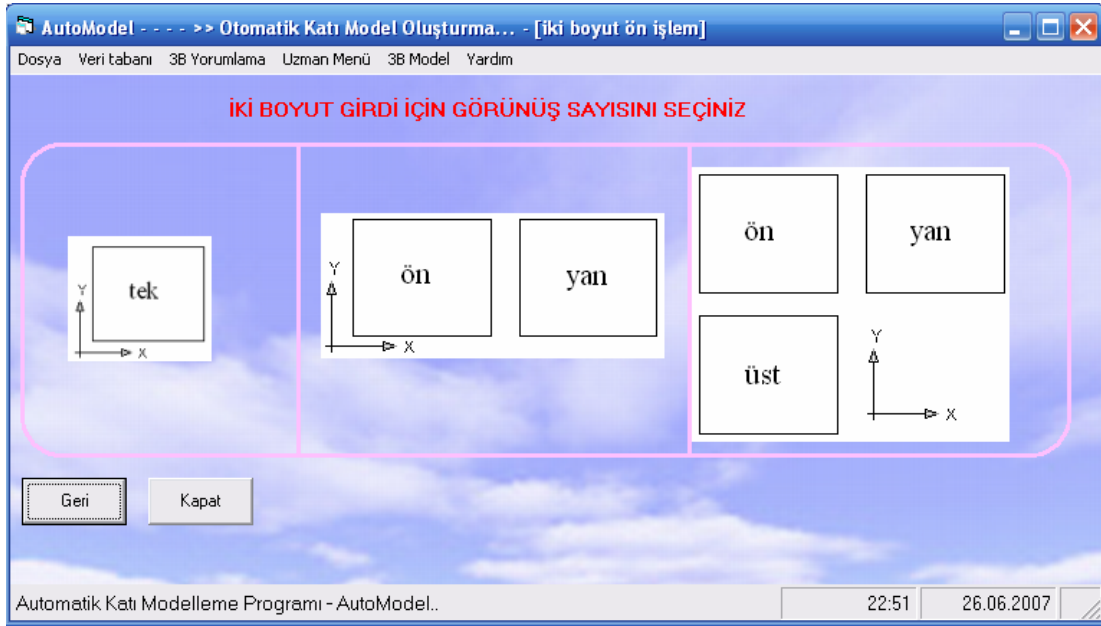
Dosya menüsü 2B çizimi oluşturmak için AutoCAD'a geçiş, AutoCAD'de boş bir çizim sayfası yaratma veya var olan bir çizimi açma, bu çizimi kaydetme gibi dosyalama işlerinin yapıldığı menüdür (Şekil 10.3).



Şekil 10.3. Dosya menüsü

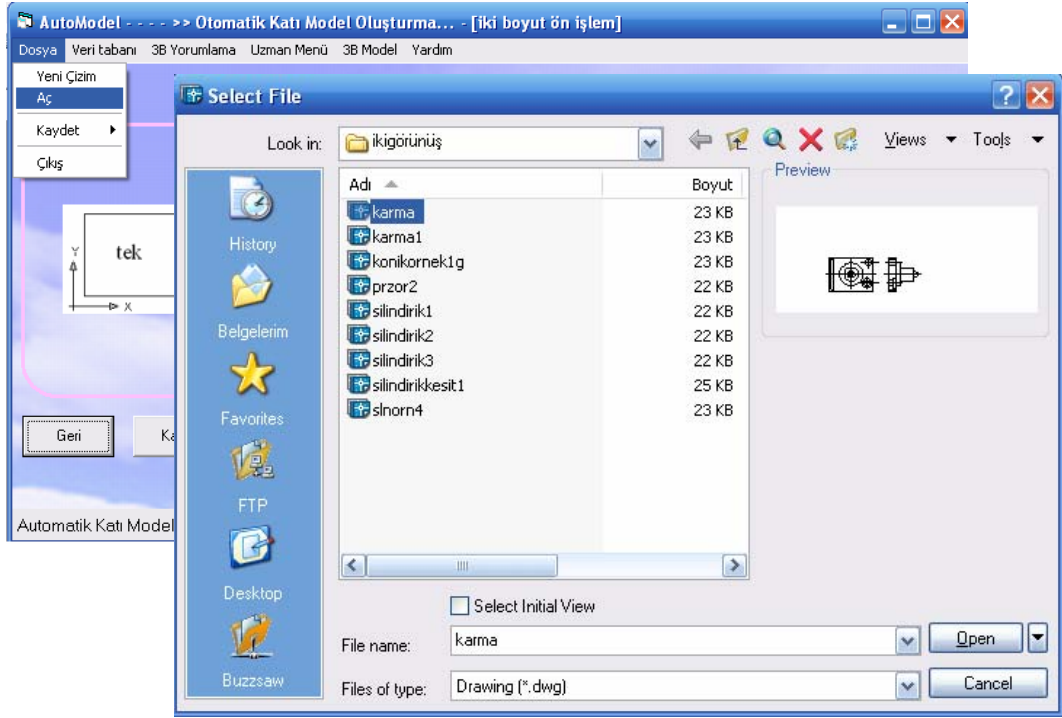
Yeni çizim: Yeni çizim seçimi tercih edildiği zaman 2B'lu çizim ile ilgili ön bilgilerin istendiği “iki boyut ön işlem” diyalog penceresi ekrana gelmektedir. Bu pencere ile

2B’lu girdinin görünüş sayısı tercih edilmektedir. Girdi görünüş sayısı tercih edildiği zaman, girdi sayısı veri tabanına kaydedilir ve otomatik olarak AutoCAD çizim sayfası açılır (Şekil 10.4). Bu aşamada, tercih edilen girdi türüne bağlı olarak katmanlar oluşturulur. Eğer kullanıcı “tek görünüş” tercih etmişse “tek”, “iki veya üç görünüş” tercih etmişse “ön, yan ve üst” isimli katmanlar oluşturulur.



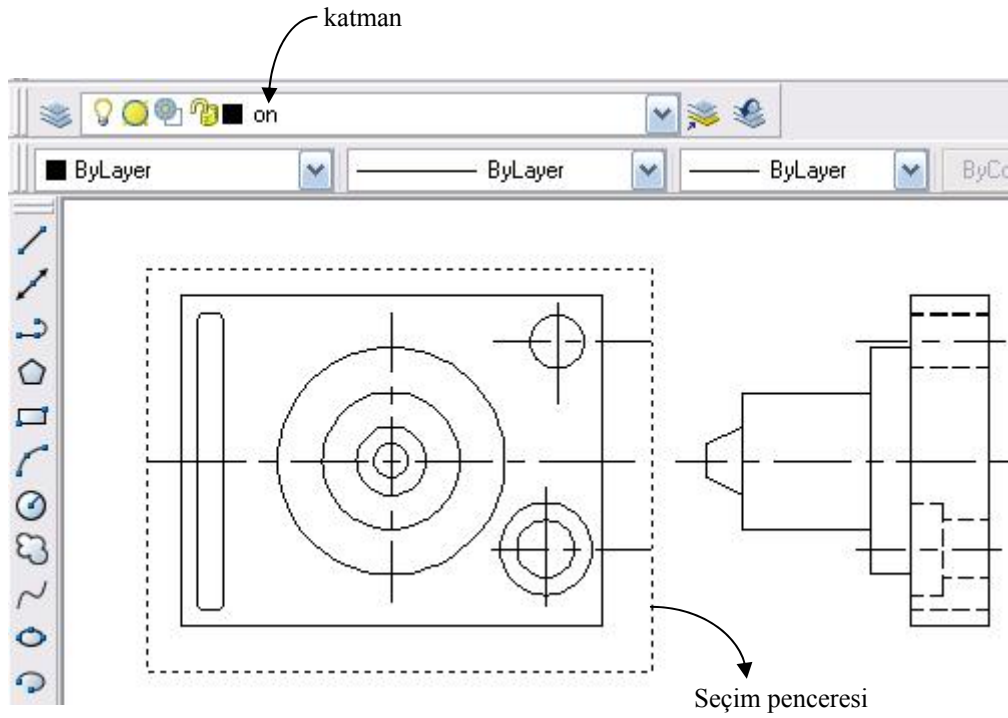
Şekil 10.4. İki boyut ön işlem penceresi

Aç: AutoCAD’la bağlantı kurarak daha önce AutoCAD ortamında çizilip kaydedilmiş resimlerin açılmasını sağlar (Şekil 10.5). Eğer program ilk defa açılıyor ise, *yeni çizim* menüsü açılarak görünüş sayısı tercih edilmelidir. Daha sonra, bilgisayarda kayıtlı resimleri açmak için *aç* menüsü seçilebilir.



Şekil 10.5. Aç menüsü ile AutoCAD’le bağlantı kurma

Eğer yeni bir çizim yapılacaksa, öncelikle çizim tamamlanmalı ve katmanlarına atılmalıdır. Çizim tek görünüş ise ekrandaki çizim otomatik olarak “tek” katmanına atılacaktır. İki veya üç görünüş ise, kullanıcının görünüşleri seçerek ilgili katmana atması gerekmektedir. Katmana atma işleminde kullanıcı görünüşleri sırasıyla bir seçim penceresi ile seçer ve ilgili katmana dahil eder (Şekil 10.6).

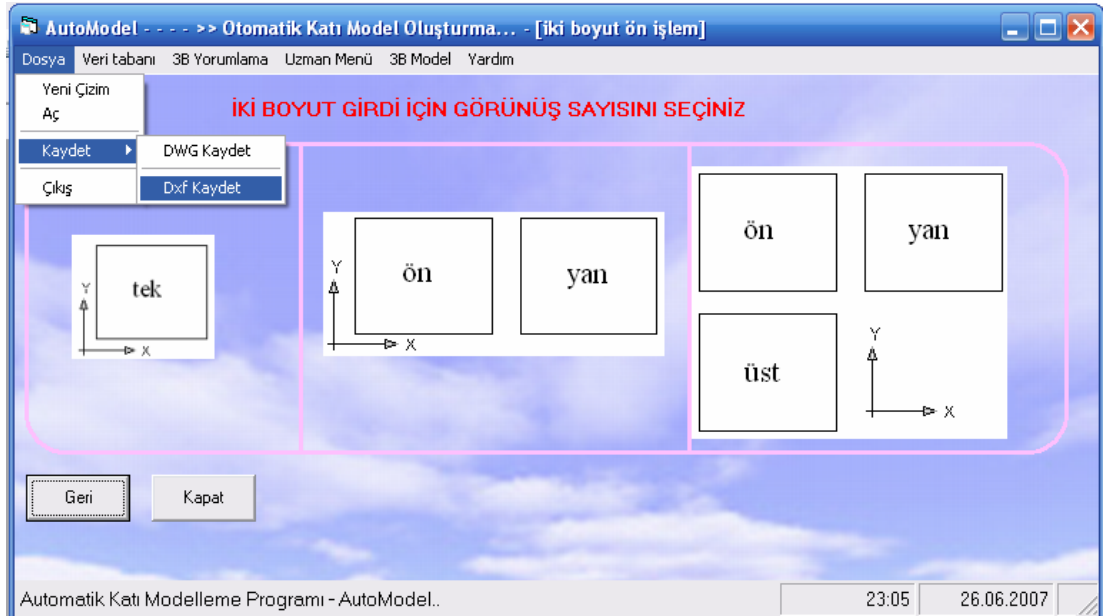


Şekil 10.6. 2B çizimin seçimi ve ilgili katmana dahil edilmesi

Çizim katmanlara atıldıktan sonra kaydedilmelidir. Program, *DXF* veri formatını kullandığı için çizim öncelikle *DXF* formatında kaydedilmelidir. Kullanıcı isterse resmi *DWG* formatında da kaydedebilir.

Kaydet: AutoCAD ortamında oluşturulan çizimleri AutoCAD çizim formatı olan *DWG* veya *DXF* veri formatında kaydetmek için kullanılır (Şekil 10.7).

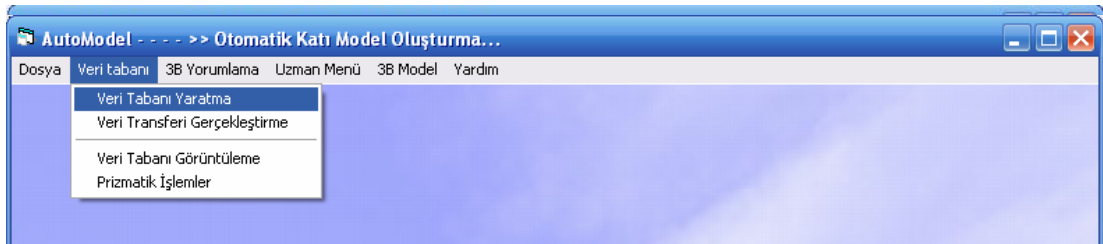
Çıkış: Çıkış tercih edildiği zaman programdan çıkılmaktadır. Çıkış komutu programı tamamen kapatmaktadır. Katı model oluşturma ile ilgili işlemler tamamlanmadan bu komut tercih edilmemelidir.



Şekil 10.7. *Kaydet* menüsü

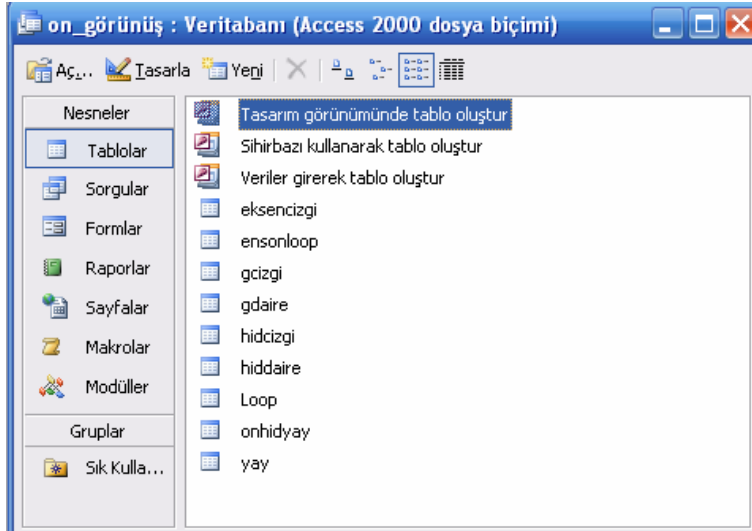
10.2.2. Veri tabanı menüsü

Veri tabanı menüsü, AutoCAD ortamında oluşturulup kaydedilen çizimle ilgili veri tabanı işlemlerinin yapılmasına aracılık eden menüdür. Bu menü, veri tabanı yaratma, veri transferi gerçekleştirme ve veri tabanı görüntüleme alt menülerinden oluşur (Şekil 10.8).



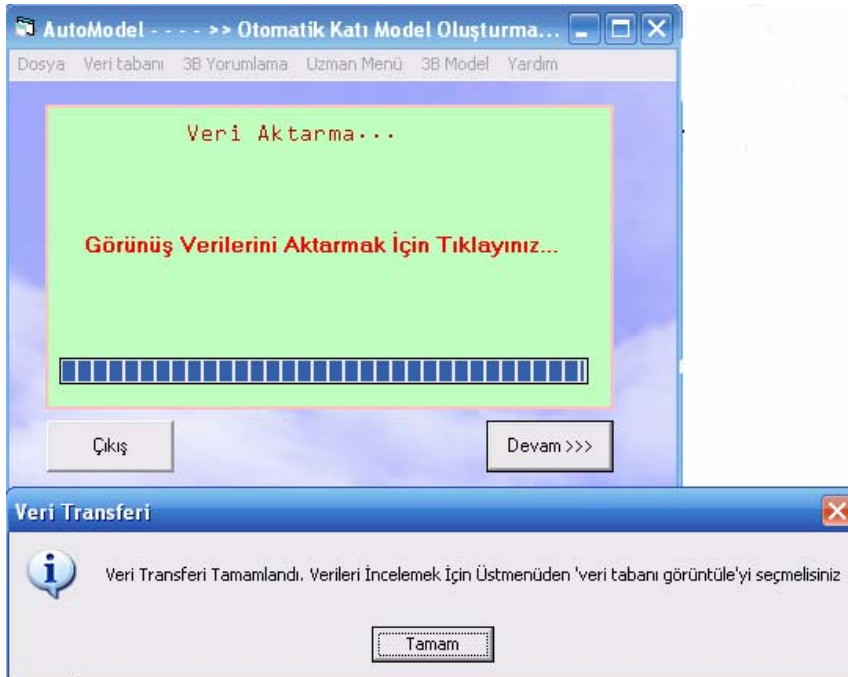
Şekil 10.8. Veri tabanı menüsü

Veri tabanı yaratma: Bu menünün tercih edilmesi ile Microsoft Access ortamındaki veri tabanı oluşturulur. Veri tabanı 7. bölümde anlatılan yöntemle, her bir görünüş bölgesi için oluşturulur (Şekil 10.9). Veri tabanı oluşturma işlemi program tarafından otomatik olarak gerçekleştirilmektedir.



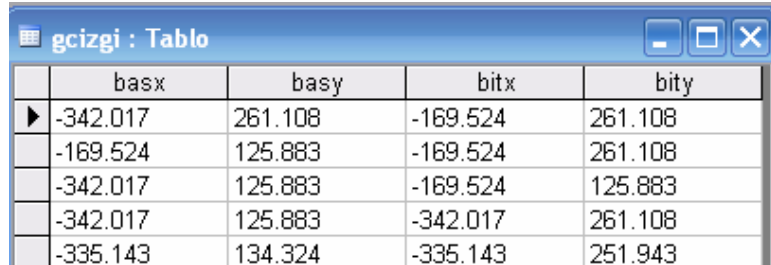
Şekil 10.9. Ön görünüş için Access ortamında oluşturulan veri tabanı

Veri transferi gerçekleştirme: Bu menünün tercih edilmesiyle ekrana veri aktarım penceresi gelir, bu pencere yardımıyla DXF formatındaki veriler düzenlenerek Access veri tabanına aktarılmaktadır (Şekil 10.10). Burada, bazı veri türlerinin text veri dosyaları da oluşturulmaktadır.



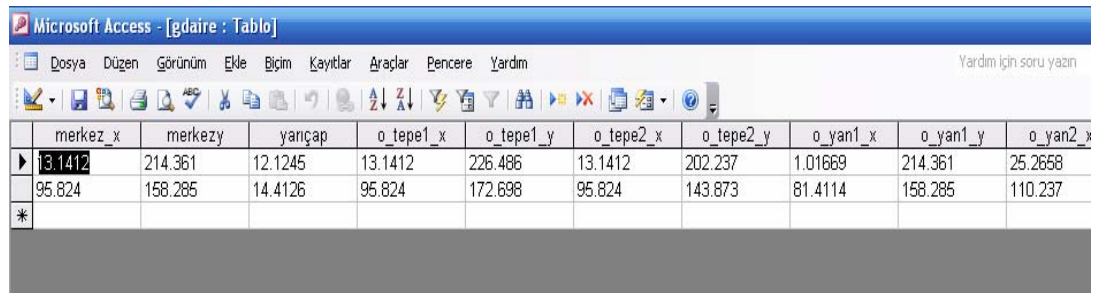
Şekil 10.10. Veri aktarma penceresi

DXF veri formatı, AutoCAD ekranındaki çizimle ilgili fiziksel verilerin yanında, çizgi türü, sistem değişkenleri ve renk gibi pek çok bilgiyi depolar. Çizim verileri, bu verilerin arasından süzgeçlenerek elde edilmekte ve daha sonra 7. bölümde verilen veri düzenleme algoritması çerçevesinde düzenlenmektedir (Şekil 10.11).



	basx	basy	bitx	bity
▶	-342.017	261.108	-169.524	261.108
	-169.524	125.883	-169.524	261.108
	-342.017	125.883	-169.524	125.883
	-342.017	125.883	-342.017	261.108
	-335.143	134.324	-335.143	251.943

(a)



	merkez_x	merkezy	yançap	o_tepe1_x	o_tepe1_y	o_tepe2_x	o_tepe2_y	o_yan1_x	o_yan1_y	o_yan2_x
▶	13.1412	214.361	12.1245	13.1412	226.486	13.1412	202.237	1.01669	214.361	25.2658
	95.824	158.285	14.4126	95.824	172.698	95.824	143.873	81.4114	158.285	110.237
*										

(b)



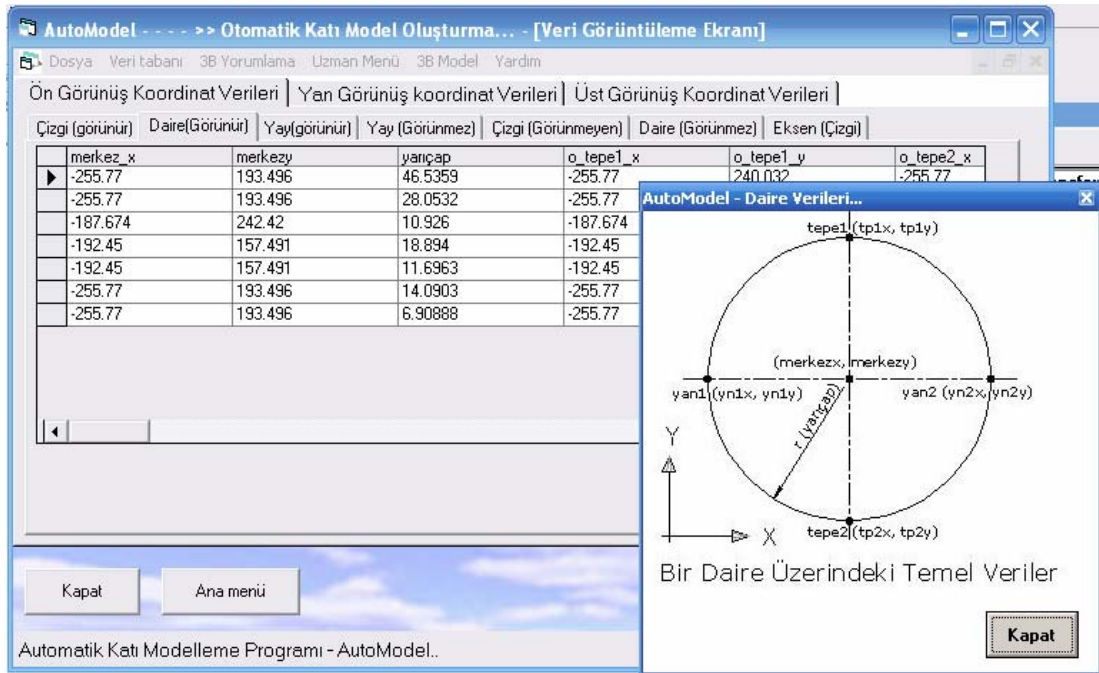
	merkez_x	merkezy	yançap	başaçı	bitaçı	basx	basy	bitx	bity
▶	9.50533	253.493	3.0	90.0	180.0	9.50533	256.493	6.50533	253.493
	9.50533	244.942	3.0	180.0	270.0	6.50533	244.942	9.50533	241.942
	141.911	253.493	3.0	0.0	90.0	144.911	253.493	141.911	256.493
	141.911	244.942	3.0	270.0	0.0	141.911	241.942	144.911	244.942
	137.655	204.885	4.44516	0.0	180.0	142.1	204.885	133.21	204.885
	137.655	151.424	4.44516	180.0	0.0	133.21	151.424	142.1	151.424
*									

(c)

Şekil 10.11. Access veri tabanında düzenlenmiş çizim verileri:

- a) doğru
- b) daire
- c) yay

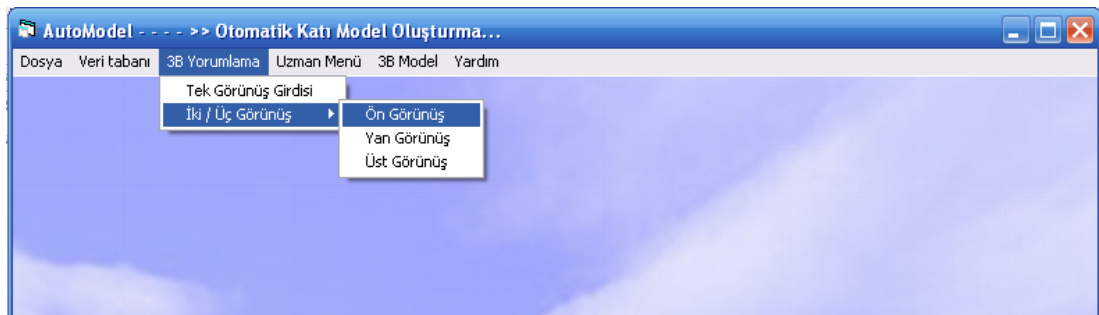
Veri tabanı görüntüleme: Bu menü yardımıyla veri tabanındaki veriler görüntülenmektedir (Şekil 10.12). Veri tabanı görüntüleme menüsünde geliştirilen anlık yardım pencereleri ile ilgili verilerin 2B temel elemanlarda ne anlama geldiği görülebilmektedir.



Şekil 10.12. Veri görüntüleme ekranı ve anlık yardım penceresi

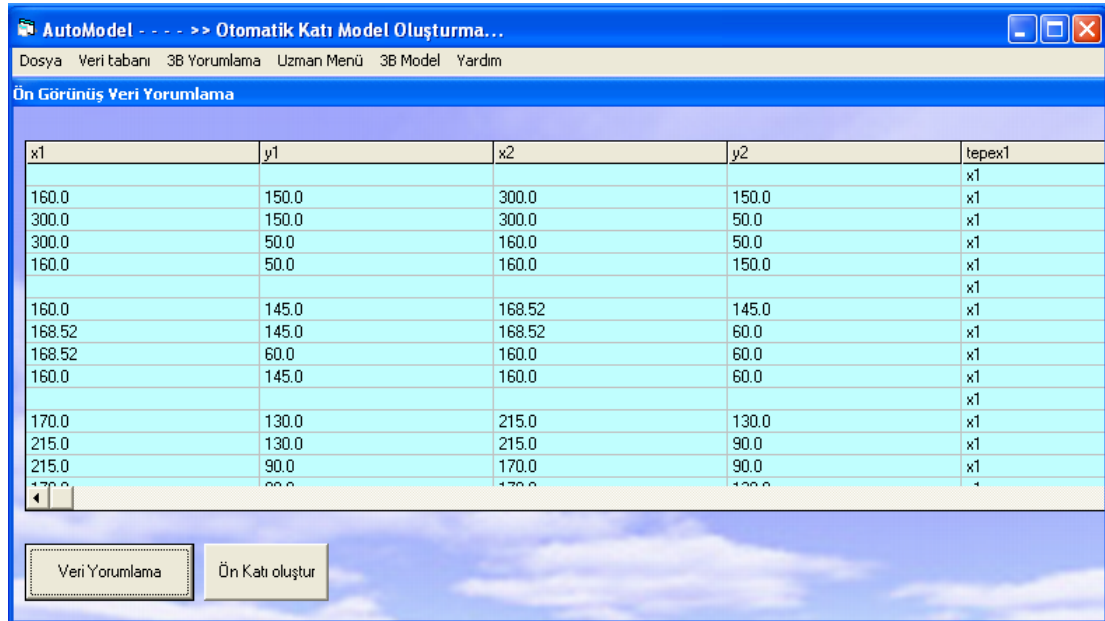
10.2.3. 3B yorumlama menüsü

Bu menü yardımıyla 3B yorumlama adımları kontrol edilir. Bu menüdeki, tek, iki ve üç görünüş alt menüleri ile 2B görünüşlerin yorumlanması için oluşturulan pencereler ekrana gelir (Şekil 10.13).



Şekil 10.13. 3B veri yorumlama menüsü

Bu alt menüler, girdi sayısına göre tercih edilerek yardımcı pencerelerin ekrana gelmesini sağlar. Yardımcı pencereler, 2B görünüşle ilgili düzenlenmiş verilerden model alanları oluşturma işlemini kontrol eder (Şekil 10.14).



Şekil 10.14. Veri yorumlama penceresi

Şekil 10.14'teki veri yorumlama penceresinde *veri yorumlama* butonuna basıldığında Access veri tabanından alınan 2B'lu temel eleman verileri yorumlanır. Burada önce ilişkili kenar kümeleri, daha sonra kapalı ve açık kenar kümeleri oluşturulur. Açık kenar kümelerinin kapalı kenar kümelerine dönüştürülmesi de bu aşamada gerçekleştirilir.

Kapalı ve açık kenar kümelerinden oluşturulan model alanları ile ilgili veriler veri yorumlama penceresindeki tablodan alınarak Access veri tabanına kaydedilir (Şekil 10.15). Veri tabanına model alanınının sınır değerleri, ait olduğu kenar kümesi (açık veya kapalı), 2B elemanların türü ve çizgi türleri kaydedilir.

ensonloop : Tablo									
	x_bas	y_bas	x_bit	y_bit	tepex	tepey	altx	alty	loop_türü
▶									kapalı loop
	-342.017	125.883	-342.017	261.108	x1	y1	x2	y2	line
	-342.017	261.108	-169.524	261.108	x1	y1	x2	y2	line
	-169.524	261.108	-169.524	125.883	x1	y1	x2	y2	line
	-169.524	125.883	-342.017	125.883	x1	y1	x2	y2	line
									kapalı loop
	-335.143	251.943	-333.143	253.943	x1	y1	x2	y2	line
	-333.143	253.943	-326.424	253.943	x1	y1	x2	y2	line
	-326.424	253.943	-324.424	251.943	x1	y1	x2	y2	line
	-324.424	251.943	-324.424	134.324	x1	y1	x2	y2	line
	-324.424	134.324	-326.424	132.324	x1	y1	x2	y2	line
	-326.424	132.324	-333.143	132.324	x1	y1	x2	y2	line
	-333.143	132.324	-335.143	134.324	x1	y1	x2	y2	line
	-335.143	134.324	-335.143	251.943	x1	y1	x2	y2	line
									kapalı loop
	-342.017	125.883	-342.017	261.108	x1	y1	x2	y2	line
	-342.017	261.108	-169.524	261.108	x1	y1	x2	y2	line
	-169.524	261.108	-169.524	125.883	x1	y1	x2	y2	line
	-169.524	125.883	-342.017	125.883	x1	y1	x2	y2	line
*									

Şekil 10.15. Model alanı veri tabanı

Model alanları ile ilgili bilgilerin veri tabanına aktarılmasından sonra, model alanlarının izdüşümleri bulunarak text veri tabanına aktarılır (Şekil 10.16).

```

duzlemsel - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
parca_prizmatik
kapalı loop
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunur
tepe noktası
-342.017
125.883
-342.017
261.108
izdusum_noktaları
-109.03
125.883
-76.4317
125.883
izdusum_cift_kose
parca_prizmatik
kapalı loop
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunmez
tepe noktası
-333.143
253.943
-326.424
253.943
izdusum_noktaları
-109.03
253.943
-76.4317
253.943
izdusum_cift_kose
parca_prizmatik
kapalı loop
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunur
tepe noktası
-342.017
125.883
-342.017
261.108
izdusum_noktaları
-109.03
125.883
-76.4317
125.883
izdusum_cift_kose
DOSYA SONU

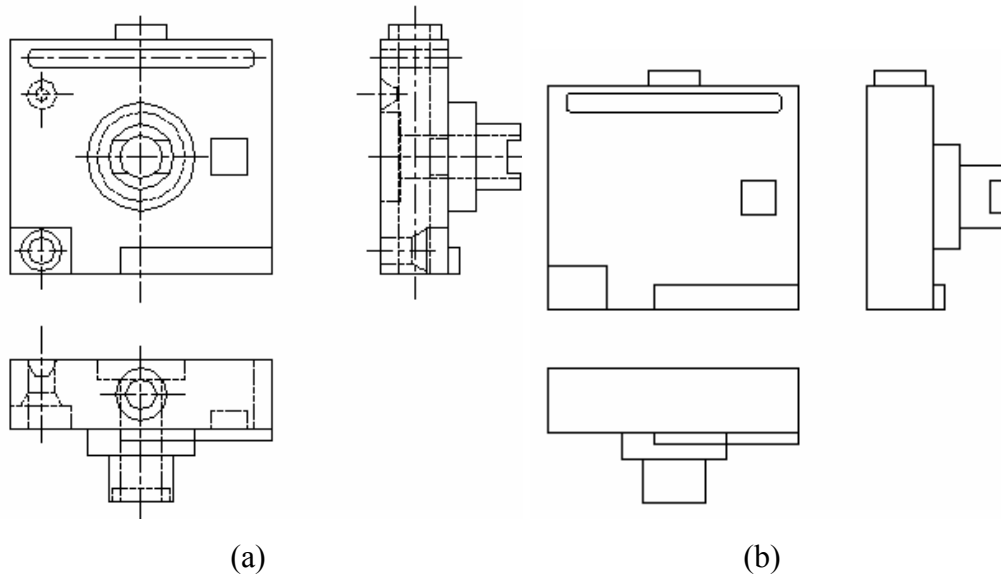
```

Şekil 10.16. Model alanı ve izdüşüm verileri

Veri tabanına, parça türü, model alanının ait olduğu kenar küme ismi, girdi ve izdüşüm çizgi türü, izdüşüm köşe nokta koordinatları ve izdüşüm köşe sayısı kaydedilmektedir. Burada oluşturulan veri tabanı uzman sistem veri tabanının oluşturulmasında kullanılmaktadır.

Bu aşamadan sonra model alanının AutoCAD ortamında tekrar çizimi ve US modülü yardımıyla katı temel elemanların oluşturulması işlemleri yapılmaktadır.

Tekrar Çizim: Tekrar çizim, VisualLISP ortamında hazırlanan bilgisayar programı aracılığı ile yapılır. Burada önce ekran temizlenmekte, daha sonra çizim yapılmaktadır. Çizim için hazırlanan program, Visual BASIC'ten AutoCAD komut satırına gönderilen komutla otomatik çalıştırılmaktadır (Şekil 10.17). Tekrar çizim işleminde ilişkili kenara sahip olmayan doğrular çizilmez. Ayrıca, çizgi türü görünmez olan açık kenar kümeleri de çizilmez. Bu tür kenar kümeleri delik unsurlarının (prizmatik veya silindirik) izdüşümleridir. Bu unsurların görünür çizgilerle ifade edildikleri görünüşlerde çizilmeleri yeterlidir.



Şekil 10.17. 2B görünüşlerin tekrar çizimi:

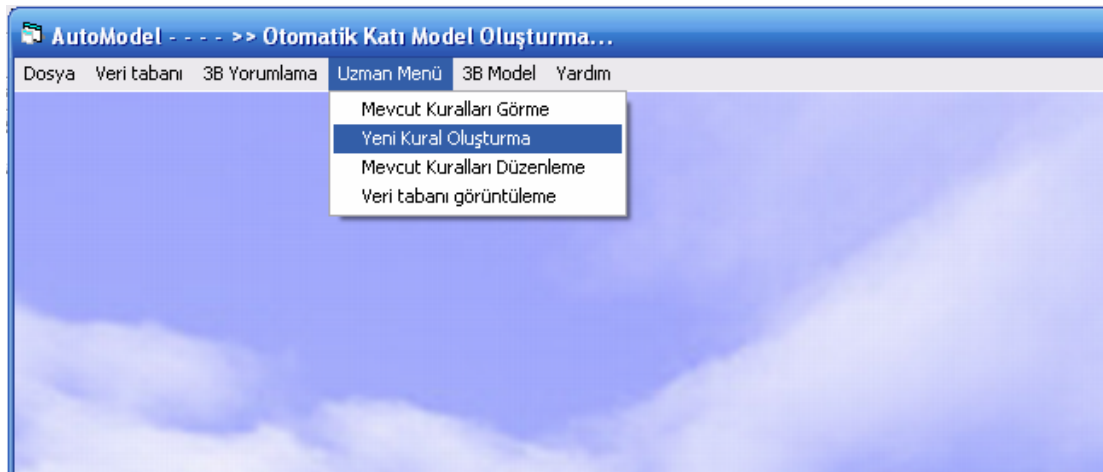
- a) Orijinal çizim
- b) Tekrar çizim sonrası

Tekrar çizim sonrası US devreye girmektedir. US veri tabanı oluşturma işlemi, tekrar çizim aşamasından sonra başlar. Veri tabanı, daha önce oluşturulan veri tabanlarındaki bilgilerin değerlendirilmesi ile otomatik olarak oluşturulmaktadır. Kullanıcı tarafından herhangi bir veri girişi yoktur. US veri tabanı, 8. bölümde anlatılan işlemler sonucu, model alanı ile ilgili temel verilerin bulunduğu veri tabanının değerlendirilmesi ile oluşturulmaktadır.

US çıkarım mekanizmasının US veri ve kural tabanını değerlendirerek temel katı çeşidine karar verme işlemi bu aşamada gerçekleşmektedir. US, programın bu aşamasında bilgi tabanını değerlendirerek ürettiği sonuçları ayrı bir veri tabanına kaydetmektedir.

10.2.4. Uzman menü

US bilgi tabanı ile ilgili işlemler *uzman menü* aracılığı ile yapılır (Şekil 10.18). US bilgi tabanı kurallar ve verilerden oluşmaktadır. Bu menü aracılığı ile bilgi tabanı ile ilgili inceleme, düzenleme ve ekleme işlemleri yapılmaktadır.



Şekil 10.18. Uzman menü

Menüye giriş için yönetici şifresi gerekmektedir (Şekil 10.19). Uzman menüye giren kullanıcı US bilgi tabanındaki verileri görme ve üzerinde düzenlemeler

yapabilmektedir. Bu bölüme *uzman kullanıcı* dışındaki kullanıcıların erişimi, US bilgi tabanına zarar verebilecektir.

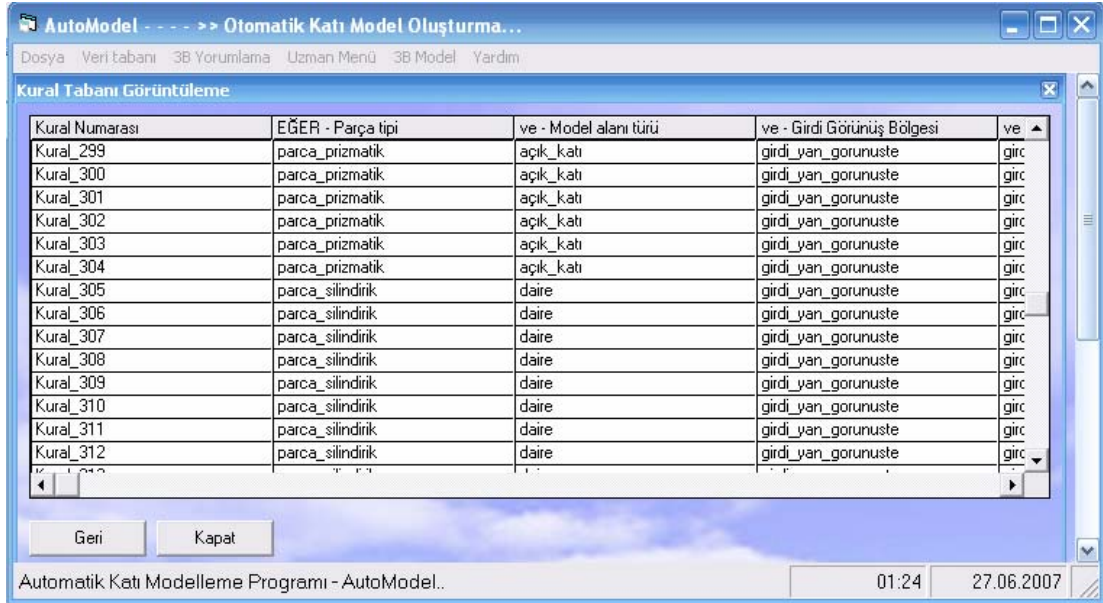


Şekil 10.19. Yönetici şifre giriş ekranı

Uzman menü aşağıdaki alt menülerden oluşmaktadır:

- Mevcut kuralları görme
- Yeni kural oluşturma
- Mevcut kuralları düzenleme
- Veri tabanı görüntüleme

Mevcut kuralları görme: Kural tabanında kayıtlı kuralların görüntülendiği alt menüdür (Şekil 10.20). Bu menü kullanılarak bilgi tabanı üzerinde herhangi bir değişiklik yapılmamaktadır.



Şekil 10.20. Mevcut kuralları görüntüleme birimi

Yeni kural oluşturma: Bilgi tabanına yeni kural eklemek için kullanılan alt menüdür. Bilgi tabanında yeni kural oluşturma kural ekleme menüsü ile yapılır. 8. bölümde belirtilen, kural oluşturma kriterlerine uygun şekilde kural eklemek için bir menü tasarlanmıştır (Şekil 10.21). Kural ekleme menüsünün dili, mevcut kural tabanının ve veri tabanının diline uyumlu olmalıdır. Aksi halde takdirde uygun çıkarımlar yapılamaz.

Uzman kullanıcı, kural tabanına ekleyeceği kuralı tanımladıktan sonra “kural ekle” butonuna basınca ekrana uyarı mesajı gelir. Bu mesaj onaylandığı takdirde kural, kural tabanına eklenir (Şekil 10.22).

Uzman Menü - Kural Ekleme

kural tabanında kayıtlı kural sayısı:573

Eklenecek Kuralın Numarası: kural_574

EĞER

2B'lu girdinin türü ----->> düzlemsel Ve

Girdi Çizgi tipi ----->> Görünmez Ve

Kenar Kümesinin Türü ---->> Açık Kenar Kümesi Ve

Girdi Görünüş Bölgesi---->> Yan Görünüş Ve

İzdüşüm Çizgi Tipi ----->> Görünür Ve

İzdüşüm Kesit Alınmış mı? >> Alınmış Ve

İzdüşüm köşe sayısı ----->> İki Ve

İzdüşüm Ana Parça
İzdüşümü Sınırları ----->> Soldan Çıkıntı **İSE**

Katı Temel Eleman: prizmatik_havsalı_boydanboya_de

Kural ekle **Ana Menü** **Kapat**

Şekil 10.21. Kural ekleme menüsü

transfer1

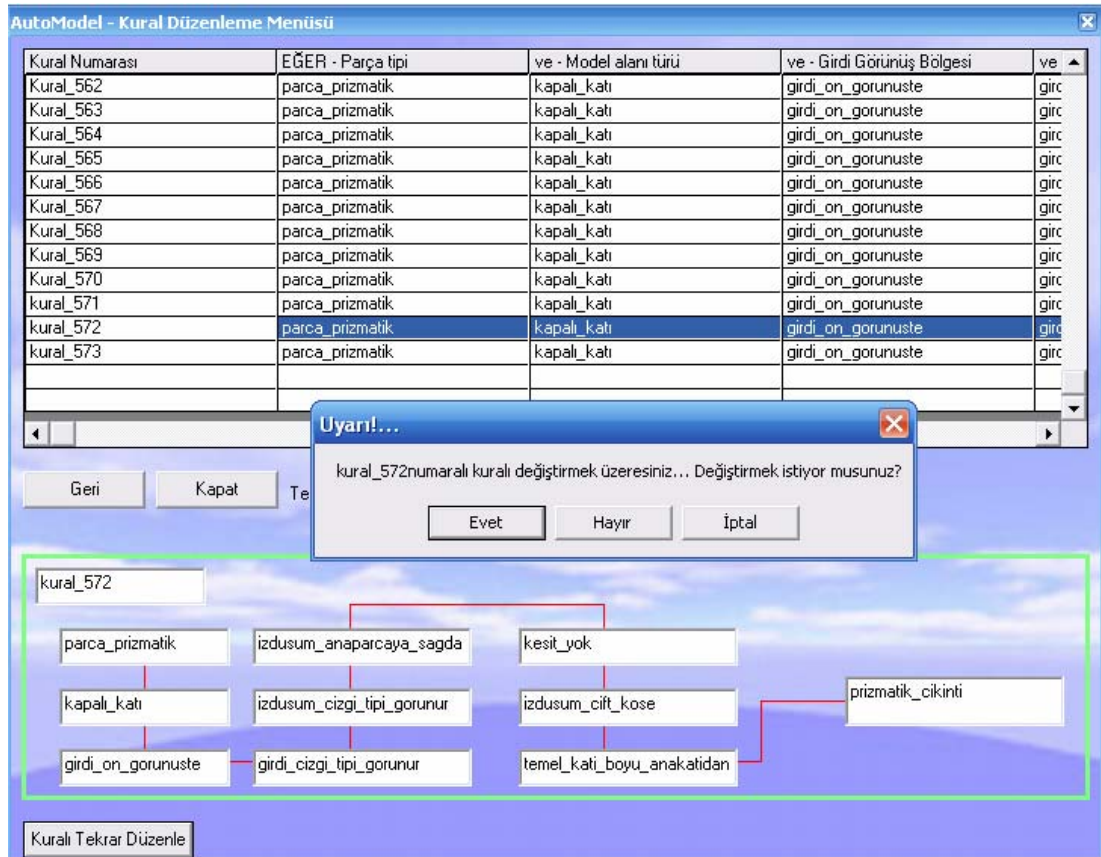
eklemeye çalıştığınız kural, kural tabanına574. kural olarak eklenecektir.Devam etmek istiyor musunuz?

Tamam **İptal**

Şekil 10.22. Kural ekleme onay uyarısı

Mevcut kuralları görme: Kural tabanında kayıtlı kurallar üzerinde değişiklik yapmak için kullanılır. Bu işlem için, kural düzenleme menüsünde görülen tablodan düzenlenmek istenen kural bulunur. İlgili kural fare ile seçildiği zaman, kuralla ilgili bilgiler menüde oluşturulan text kutularına taşınır. Uzman kullanıcı, kural üzerindeki değişikliği text kutuları yardımıyla yapmaktadır. Gerekli düzenlemeler yapıldıktan

sonra *kuralı tekrar düzenle* butonuna basılır. Uyarı mesaj kutusunda *hayır* veya *iptal* butonları seçilirse kural üzerinde yapılan düzenlemeler etkinleşmez. Evet butonu seçilirse kural düzenlenmiş haliyle kural tabanına kaydedilir.



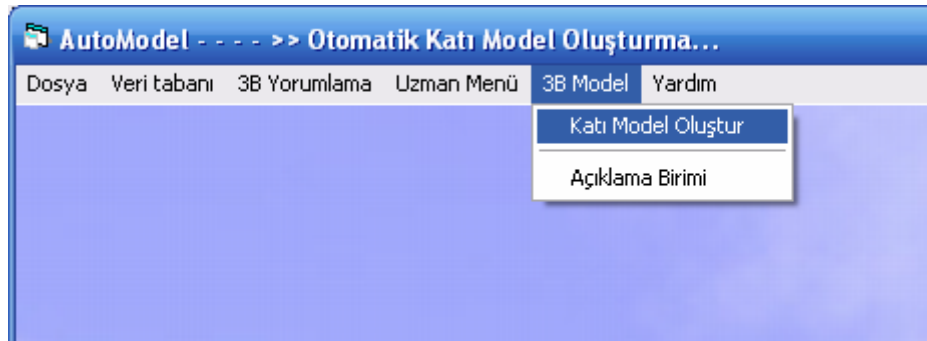
Şekil 10.23. Kural düzenleme işlemi

US yardımıyla tanımlanan katı temel elemanlar Şekil 10.14’te gösterilen veri yorumlama menüsündeki “ön katı oluştur” butonu tıklanarak oluşturulur. Ön katı oluştur butonu ile çalıştırılan program parçaları önce ilgili görünüşteki (ön, yan veya üst) katı temel elemanları daha sonra katı temel elemanlar arasında boolean operasyonları (ekleme ve çıkartma) uygulayarak ön katıları oluşturur. Katı temel eleman oluşturma aşamasında kullanılan veriler, US çıkarım mekanizması ile elde edilip veri tabanına aktarılmaktadır.

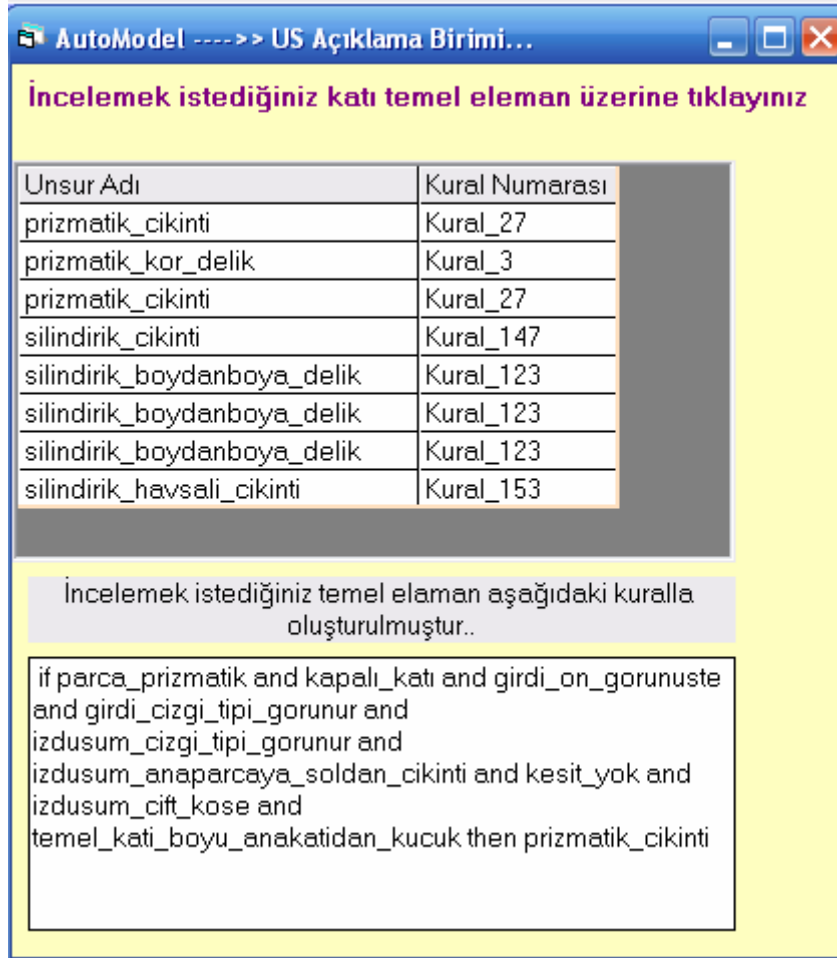
10.2.5. 3B model menüsü

3B model menüsü, ön katılardan katı model oluşturma işleminin gerçekleştirildiği menüdür (Şekil 10.24). 3B model menüsü kullanılarak Visual BASIC'te hazırlanan program parçası ile AutoCAD komut satırına komut gönderilerek nihai katı modeli oluşturmada kullanılan VisualLISP programı otomatik olarak çalıştırılır. Bu program, ön katıları kendi görünüş bölgelerinde döndürmekte, ön görünüş üzerine taşımakta ve kesiştirme ile son katı modeli oluşturmaktadır. Bu işlemlerde US çıkarım mekanizması ile elde edilen veriler kullanılmaktadır.

Ayrıca bu menü üzerinde US açıklama birimi de bulunmaktadır. Bu birim aracılığı ile, oluşturulan katı temel elemanların unsur türü, oluşturulduğu kural numara ve dizgisi görüntülenmektedir (Şekil 10.25).



Şekil 10.24. 3B model menüsü



Şekil 10.25. US açıklama birimi


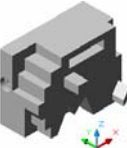
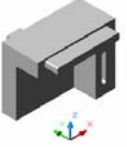



10.3. Program Performansının Test Edilmesi

Tez çalışması kapsamında hazırlanan bilgisayar programı Pentium 4 3.2 GHZ işlemcili, 1024 MB hıza sahip bir bilgisayarda test edilmiştir. Bu bilgisayarda 2B girdi oluşturulduktan sonraki katı model oluşturma aşamalarının hızı ölçülmüştür. Hız ölçümü sırasında, katı model oluşturma hızının ve programın çalışırken hafızada kapladığı alanın parça görünüş sayısı ve karmaşıklığa göre değiştiği gözlenmiştir (Çizelge 10.1 ve 10.2).

Çizelge 10.1. Bazı örnek parçalarda programın performansı

Örnek Parça	Görünüş Sayısı	Hafızada Kapladığı Alan (mb)	İşlem Süreleri (sn)		
			Model Oluşum Süresi		Toplam Süre
			Ön Katı	Son Katı	
	3	4.700	4.65	3.52	28.35
	3	5.200	5.83	3.86	30.16
	3	5.438	4.32	4.02	31.42
	3	6.860	4.22	4.16	32.08
	3	5.350	5.03	3.95	30.24
	1	2.556	-	1.31	15.05
	1	2.388	-	0.95	16.44

Çizelge 10.2. Bazı örnek parçalarda programın performansı

Örnek Parça	Görünüş Sayısı	Hafızada Kapladığı Alan (mb)	İşlem Süreleri (sn)		
			Modelin Gösterim Süresi		Toplam Süre
			Ön Katı	Son Katı	
	1	2.692	-	0.84	15.38
	2	3.700	4.02	2.02	21.84
	2	5.200	3.38	1.78	23.30
	2	3.250	3.62	1.83	22.40
	2	2.576	3.54	2.06	26.4
	2	4.163	3.47	1.95	24

Çizelgelerde verilen test sonuçlarında görüldüğü gibi katı model süresi oldukça kısadır. Çizelgede görülen süreler katı modeli oluşturulacak parçanın 2B görünüşlerinin oluşturulmasından sonraki aşamalarını kapsamaktadır. Bunlar, kayıt, veri tabanı oluşturma, tekrar çizim, ön ve son katı oluşturma aşamalarıdır.

Programda bir nesnenin katı modelini oluşturmak için geçen sürenin önemli kısmı veri tabanı oluşturma ve yorumlama aşamalarını kapsamaktadır. Veri tabanı işlemleri

sonrasında gerçekleştirilen tekrar çizim, ön katı ve son katı oluşturma aşamaları oldukça kısa bir sürede gerçekleştirilmektedir.

Modelin ekranda oluşturulma süresi parçaların karmaşıklık seviyesine göre değişmektedir. Çalışmada ele alınan örnek parçalarda programın modelleme zamanının, parça görünüş sayısı ve karmaşıklığına bağlı olarak 15 ila 35 saniye arasında değiştiği görülmüştür. Bu süreler kullanıcının programa hakimiyetine bağlı olarak uzayıp kısalabilir. Özellikle kayıt, veri tabanı oluşturma gibi işlemlerde kullanıcının menülere hakimiyeti işlem zamanının etkilemektedir.

Hazırlanan program bilgisayar belleğinde az yer kaplamaktadır. Çizelge 10.1 ve 10.2'den de görüldüğü gibi program çalışırken bilgisayar belleğinde 2.5 ila 5.2 mb kadar küçük bir yer kaplamaktadır. Programın bellekte kapladığı alan sadece tek işlem için ölçülmüştür. Programla üst üste işlem yapıldığı takdirde hafızada kapladığı yer artmaktadır. Bunu engellemek için her katı model oluşturma işlemi için program açılıp kapatılabilir. Yapılan ölçümlerde programın uzun süre çalışır halde bırakılması durumunda dahi bu oranın 40 mb'ın üzerine çıkmadığı gözlenmiştir.

Programın performansının test edilmesi sonucunda çalışmanın bilgisayar belleğinde az yer kapladığı ve hızlı sonuç ürettiği gözlenmiştir. Programın tüm işlem zamanı ile ilgili verilen süreler kesin süreler değil, yaklaşık süreler olarak kabul edilmelidir. Çünkü programın işlem zamanı programcının programı kullanma tarzına göre uzayıp kısalabilir. Burada görülmektedir ki, hazırlanan program 16 ila 40 saniye arasında değişen bir zaman diliminde katı model oluşturma işlemini tamamlamaktadır. Bu süreler CAD/CAM alanında kullanılan herhangi bir ticari paket programda bir katı modelin oluşturulma süresi ile karşılaştırıldığında çalışmanın modelleme zamanı açısından büyük avantajlar içerdiği görülmektedir. Çalışma anında bilgisayar belleğinde az yer kaplaması hem programın hem de bilgisayarın performansını olumlu etkilemesi yönünden önemlidir.

11. SONUÇ VE ÖNERİLER

Bu tezde tek görünüşle ifade edilen dönel simetrik, iki ve üç görünüşle ifade edilen silindirik ve prizmatik nesnelerin 3B katı modelini oluşturma amacı ile yürütülen bir çalışma tanıtılmıştır.

2B çizimlerle ifade edilen nesnelerin katı modellerini oluşturma amaçlı yürütülen çalışmada bu konudaki üç temel problemin çözümü amaçlanmıştır. Bunlar; düzenleme, yorumlama ve yeniden oluşturmadır.

Düzenleme, AutoCAD ekranında yapılan çizime ait verilerin düzenlenmesini kapsar. Bunlar, 2B’lu çizim elemanlarına (doğru ve yay) ait verilerin belirlenmiş bir yönde sıralanması, DXF formatındaki verilerin anlaşılabilir bir şekilde düzenlenmesi ve mevcut verilerden yararlanarak yeni verilerin oluşturulmasıdır.

Yorumlama, düzenlenen verileri kullanarak görünüş bölgelerindeki girdiler arasında ilişkiler kurma ve 3B nesne yorumları oluşturmayı içerir. Yorumlama iki temel adımdan oluşur. Birinci adım, görünüş bölgelerindeki 2B’lu çizim elemanlarının (doğru, yay ve daire) birbirleriyle bağlantı durumlarının incelenerek çizim kümeleri oluşturmaktır. İkinci adım, görünüş bölgelerinde oluşturulan çizim kümelerinin izdüşümleri ile ilişkilerinin incelenerek 3B’lu nesne yorumlarının oluşturulmasıdır. Burada, 3B’lu nesne yorumları geliştirilen kurallar ve uzman sistem destekli bir karar mekanizması aracılığı ile oluşturulmuştur.

Yeniden oluşturma, yorumlama adımı sonuçlarının kullanılarak katı modelin oluşturulmasıdır. Bu aşamada birinci işlem, 2B’lu çizim kümelerinin tekrar çiziminin gerçekleştirilmesidir. İkinci işlem bu çizim kümelerinin, yorumlama adımı uzman sistem yardımıyla oluşturulan 3B’lu nesne yorum sonuçları kullanılarak katı temel elemanlara dönüştürülmesidir. Yeniden oluşturmada son aşama katı temel elemanlara boolean operasyonları uygulayarak katı model oluşturmaktır.

Araştırma kapsamında geliştirilen yaklaşımı test etmek için prototip bir bilgisayar programı geliştirilmiştir. Bu bilgisayar programı iki modülden oluşmaktadır. Bunlar, tek görünüşle ifade edilen dönel simetrik parçaların katı modellerini oluşturan program modülü, diğeri ise iki veya üç görünüş girdisi ile ifade edilen parçaların katı modellerini oluşturan program modülüdür.

Tek görünüşle ifade edilen dönel simetrik parçaların katı modellerini oluşturma işlemi ana adımları, girdiyi oluşturan 2B'lu temel elemanları düzenleme, düzenleme temel elemanlardan model alanı oluşturma ve model alanını kendi eksenine etrafında 360° döndürerek katı model oluşturmaktır. Geliştirilen yaklaşım ile dönel simetrik elemanların katı modelleri oluşturulabilmektedir. Burada tam ve yarım kesitle ifade edilmiş parçaların da katı modelleri oluşturulabilmektedir.

İki veya üç görünüşle ifade edilen parçaların katı modellerini oluşturmaya yönelik geliştirilen yaklaşımın ana işlem adımları; düzlemsel ve silindirik bölgelerden model alanları oluşturma, model alanlarından ana ve temel katılar oluşturma, görünüş bölgelerinde boolean operasyonları ile ön katılar ve ön katıların hacimsel kesiştirilmesiyle katı model oluşturmaktır.

Bu çalışmada hacimsel bir yaklaşım kullanılmaktadır. Hacimsel yaklaşım, 2B girdilere normal vektörleri doğrultusunda veya tersi yönünde kalınlık vererek katı temel elemanlar oluşturmaya esas alır. Katı model bu katı temel elemanlar arasında yapılan boolean operasyonları ile oluşturulur.

Katı model oluşturmada nesne temsil tekniği olarak Yapısal Katı Geometri (YKG) temsil tekniği kullanılmıştır. YKG temsilleri ile modelleme kolay girdi sağlaması, kontrolünün kolay olması sebebi ile tercih edilmektedir. Bunun yanında model oluşturma'nın kolay olması YKG'yi çalışmada tercih edilebilir bir temsil tekniği yapmıştır.

Yapılan çalışmada, Shum ve arkadaşlarının iki aşamalı hacimsel kesiştirme yaklaşımı temel alınmıştır [9, 65, 66]. Bununla birlikte burada işlem tek aşamaya düşürülmüştür. Shum'un iç ve dış katı olarak sınıflandırdığı katı temel elemanlar aynı aşamada oluşturulmaktadır. Ayrıca geliştirilen yöntem, prizmatik parçalarda çıkıntı unsurlarını oluşturabilmesi yönüyle de hacimsel kesiştirme yaklaşımını kullanan çalışmalardan ayrılmaktadır.

Çalışma üç farklı girdi sayısını bir arada kullanması ile benzer çalışmalardan ayrılmaktadır. Daha önce yapılan çalışmalarda girdi sayıları tek, iki, üç ve altı görünüş olarak tercih edilmiştir. Çalışmalarda girdi tercihleri bunlardan herhangi biri kullanılarak yapılmıştır. Çalışma, üç girdi türünü (tek, iki veya üç) bir arada kullanması yönüyle diğer çalışmalardan ayrılmaktadır.

2B izdüşümlerin yorumlanmasında DXF (Drawing Exchange Format) veri formatından faydalanılmıştır. DXF veri formatı, 2B'lu izdüşümlerin sınır değerlerine ulaşmadaki esnekliği ve kolay yorumlanabilir yapısı nedeniyle tercih edilmiştir.

Çalışmada 2B'lu girdi AutoCAD çizim elemanlarından doğru, yay ve çember kullanılarak oluşturulmuştur. Bu elemanlarla yapılacak çizimlere boyut, şekil, çizgi türü vb. sınırlamalar getirilmemiştir. Girdi görünüşleri birinci izdüşüm yöntemine göre oluşturulmuştur.

2B'lu girdileri yorumlayarak katı model oluşturmaya yönelik yürütülen tez çalışmasında aşağıdaki sonuçlar elde edilmiştir:

- Çalışmada silindirik elemanlardan boydan boya, kör, kademeli, havşa başlı delik ve silindirik çıkıntılar oluşturulabilmektedir. Prizmatik elemanlardan, prizmatik çıkıntı, boydan boya ve kör prizmatik delikler oluşturulabilmektedir. Bazı konik prizmatik elemanların da katı modelleri oluşturulmuştur.

- Hacme dayalı çalışmalarda en büyük problem kapalı bir alan oluşturmeyan düzlemsel bölgelerdir. Bunlar, kapalı kanallar, köşe cepler ve çıkıntılar vb. gibi unsurlardır. Geliştirilen AKKA (Açık Kenar Kapatma Algoritması) ile bu tür kenarlar kapalı kenar kümelerine dönüştürülmekte, daha sonra bunlardan katı temel elemanlar oluşturulmaktadır. AKKA, kenar sayı sınırlaması getirmemesi ve yayları işleme alması yönüyle daha önce geliştirilen çalışmalardan ayrılmaktadır.
- Çalışmada, 2B'lu görünüş bölgesinde görünmez çizgilerle ifade edilen (arka tarafında kalan) kapalı bölgeler de yorumlanabilmektedir. Çalışma, görünmez kenarlarla ifade edilen delik veya çıkıntı unsurlarını yorumlayabilmesi yönüyle önemli avantajlara sahiptir.
- İki ve üç görünüşle ifade edilen parçaların katı modellerini oluşturma sürecinde yapay zeka tekniklerinden uzman sistem kullanılmıştır. 3B'lu temel elemanın biçimine karar verme aşamasında uzman sistem destekli bir karar mekanizması geliştirilmiştir. Uzman sistem yardımıyla temel elemanın türüne (delik veya çıkıntı), biçimine (düz veya konik) ve yapılacak boolean operasyonuna (ekleme veya çıkartma) karar verilmektedir.
- Geliştirilen algoritma ile tek kalınlıklı parçaların tam ve yarım kesitleri yorumlanabilmektedir. Daha önce bu konuda yapılan bir çalışmada iki görünüş girdisi ile ifade edilen bazı parçaların kesit görünüşleri ele alınmıştır [64]. Bu çalışmada benzer şekilde kesitler ele alınmakta, kesit görünüş uygulamaları tek ve üç görünüş girdisini kapsayacak şekilde genişletilmektedir.
- Bütün bu aşamalarda çok yönlü bir veri tabanı yönetimi söz konusudur. Veri tabanı oluşturma ve düzenleme işlemleri 2B'lu girdi verilerinin DXF veri formatından dönüştürülmesi aşamasından katı model oluşturma aşamasına kadar devam eder. Tüm bu aşamalar sonunda, 2B'lu girdi, katı modelin oluşturulduğu katı temel elemanlar hakkındaki bilgiler bir veri tabanına aktarılmaktadır.

- Çalışmada 2B'lu girdinin düzenlenmesi, ilişkili kenar kümeleri oluşturma, köşe ve kenar ceplerin oluşturulması, katı temel elemanların türlerine karar verme vb. problemlerin çözümüne getirilen yeni yaklaşımlar, ileride yürütülecek çalışmalarda özellikle yorumlama adımlarına önemli bir teorik katkı sağlayacaktır.
- Geliştirilen bilgisayar programı, işlem anında bilgisayar hafızasında kapladığı yerin azlığı ve çalışma hızıyla etkili sonuçlar vermiştir. Ticari BDT programlarında dakikalarla ifade edilen katı modelleme işlemi, geliştirilen AutoModel programı ile birkaç saniyede gerçekleştirilmektedir.

Çalışma sayılan avantajlarının yanında bazı sınırlılıklara sahiptir. Bu sınırlılıklar ve nedenleri aşağıdaki şekilde belirtilebilir:

1. Prizmatik parçalarda üst üste gelen kenarlar yorumlanamamaktadır. Örnek olarak önden görünüşü üs üste gelmiş prizmatik çıkıntı veya delikler verilebilir. Bu tür parçalarda ön tarafta kalan (görünür) nesne oluşturulabilmekte arkada kalan nesne ise oluşturulamamaktadır. Silindirik parçalarda böyle bir sınırlama yoktur. Bu tip problemler yardımcı görünüşlerle halledilebilir. Çalışmada yardımcı görünüşler ele alınmadığı için üst üste gelen kenarların durumları, özellikle prizmatik parçalarda, çözülememiştir.
2. Prizmatik ve silindirik parçalarda görünmez kenarlarla ifade edilen iç çıkıntılar oluşturulamamaktadır. Program bu tür parçaları delik olarak tanımlayabilmektedir. Tez çalışmasında kullanılan nesne temsil tekniği olan YKG temsilleri bu tür karmaşık durumlarda yetersiz kalmaktadır. Bu tür karmaşık parçalar için sınır temsili tekniğine yönelmesi başarılı sonuçlar verebilir. Çünkü, sınır temsili modeller, karmaşık parçaların modellenmesinde YKG temsillerine göre önemli avantajlara sahiptir.

3. Bir çizginin iki farklı çizgi türüne sahip olması durumu yorumlanamamaktadır. Katı temel elemanın türüne karar vermede önemli kriterlerden birisi izdüşümün çizgi türüdür. Çizgi türündeki bu tür karmaşıklık temel eleman türünün (delik veya çıkıntı) belirlenmesine engel olmaktadır. AutoCAD ortamında yapılan çizimlerde bir çizgiye iki farklı çizgi tipi tanımlanamamaktadır. Elle yapılan çizimlerde tanımlanan bu durumlar AutoCAD ortamında ancak çizgi kırılarak gerçekleştirilebilir. Bu da program içinde izdüşüm boyunun yanlış hesaplanmasına yol açmaktadır. Bu tür parçalarda kısmi kesit uygulamaları ile başarılı sonuçlar alınabilir.
4. Çalışmada farklı kalınlıklara sahip (kademeli gibi) parçaların kesitleri yorumlanamamaktadır. Burada, kademe farklılıkları temel elemanların yanlış belirlenmesine sebep olabilmektedir.
5. Çok parçalı nesnelerin kesit görünüşleri yorumlanamamaktadır. Kesit alınan parçada ana ve temel katı izdüşümlerinin tespit edilmesi çok özel parçalar dışında oldukça güçtür. Bunun en önemli nedeni DXF veri yapısında kesit bölge sınırlarını tespit etmedeki güçlülüdür. Bu güçlükten dolayı, çalışmada kesit görünüşlerle ilgili alanda yeni bir takım sonuçlar elde edilse de istenen hedeflere ulaşılamamıştır. Kesitlerle ilgili bu sıkıntı, kesit bölge tanımlamasında daha etkili başka bir veri standardıyla aşılabılır.
6. Çalışmada AutoCAD grafik ekranının kullanılması ile AutoCAD'in grafik üstünlüklerinden faydalanılmaktadır. Fakat burada model kontrolünde güçlükler ortaya çıkmaktadır. Komutların bazı durumlarda çalışmaması, izdüşüm köşelerinin birbirini karşılayabilmemesi gibi problemlerle karşılaşılmaktadır.

Tez çalışması süresince ortaya konan yeni görüşler ve tespit edilen eksiklikler çerçevesinde, bu konuda gelecekte yürütülecek çalışmalarda aşağıdaki konular ele alınabilir:

1. Son yıllarda bazı çalışmalarda karma yöntemlerin uygulandığı görülmektedir [74-

76]. Karma (hybrid) yöntemler, YKG ve sınır temsili temsil tekniklerini bir arada kullanmaktadırlar. Bu çalışmalarda YKG'nin katı model oluşturmadaki esnekliği ile sınır temsili zor yüzeyleri oluşturma kabiliyeti bir araya getirildiği takdirde önemli sonuçlar elde edilebilecektir.

2. Kesitlerle ilgili uygulamalar çok parçalı nesnelere doğru genişletilebilir. Ayrıca, kademeli ve kısmi kesitler katı model oluşturma algoritmasına dahil edildiği takdirde katı model daha az görünüş girdisi ile oluşturulabilir.
3. Mevcut sistemde tek görünüşle ifade edilen dönel simetrik parçaların katı modelleri oluşturulabilmektedir. Prizmatik parçaların tek görünüş girdisi ile katı modellerini oluşturacak çalışmalar çözüm uzayını genişletecektir. Ekrandaki görüntüyü tanıyan, kullanıcıdan sadece unsur tipi ve kalınlık bilgisi isteyen bir sistemle bu tür parçalar, fazla girdi görünüşüne gerek kalmadan oluşturulabilir.
4. Yapılan çalışma AutoCAD ortamında yapılan çizimleri kabul etmektedir. Gelişen teknoloji ile birlikte alternatif girdi yöntemleri gelişmiştir. Bunlara internet aracılığı ile elde edilen *jpeg* formatındaki resimler, tarayıcılar aracılığı ile elde edilen görüntüler ve x-ray cihazları ile elde edilen görüntüler örnek verilebilir. Son yıllarda x-ray cihazı ile elde edilen görüntüleri inceleyen çalışmalar vardır [66]. Fakat *jpeg* formatındaki girdileri ele alan çalışmalara rastlanmamaktadır. İleriki çalışmalar taranmış veya internet aracılığı ile elde edilen resimleri ele alabilecek çalışmalara doğru yönlendirilebilir.
5. Görünmez kenarlarla ifade edilen iç çıkıntı gibi karmaşık durumlar karşısında yetersiz kalan üç görünüş girdisinin yanında yardımcı görünüşleri ele alan çalışmalar yürütülebilir.
6. Bu çalışmada hacimsel yaklaşımlarda kullanılmayan bir YZ uygulamasına başvurulmuştur. Tel kafes yaklaşımlarda görülen YZ uygulamaları hacimsel yöntemlerde görülmemektedir. Özellikle uzman sistemin kural eklenebilir yapısı

kullanılarak nesne yorumlamada ortaya çıkabilecek yetersizliklere müdahale edilebilir. İleride yürütülecek hacim tabanlı çalışmalarda, kural tabanlı diğer YZ sistemleri de katı model oluşturma sürecine dahil edilebilir.

KAYNAKLAR

1. Börklü, H.R., "Towards a language for mechanical design", Ph.D. Thesis, *University of Leeds*, Leeds, 1-3 (1993).
2. Pahl, G., Beits, W., "Engineering design: A systematic approach", *The Design Council Springer – Verlag*, London, 22-49 (1984).
3. Gujar, U.G., Nagendra, I.V., "Construction of 3D solid objects from orthographic views", *Computer & Graphics*, 13 (4): 505-521 (1989).
4. Wang, W., Grinstein, G.G., "A survey of 3D solid reconstruction from 2D projection line drawings", *Computer Graphics Forum*, 12 (2): 137-158 (1993).
5. You, C.F., Yang, S.S., "Reconstruction of curvilinear manifold objects from engineering drawings", *Proceedings of the 12th National Conference of the Chinese Society of Mechanical Engineers*, Chia-Yi, 637-646 (1995).
6. Şahin İsmail, Börklü, H.R., "2B görünüş verilerinden bilgisayar destekli 3B modeller oluşturulması: Kaynak araştırması", *Mühendis ve Makine*, 507: 20-31 (2002).
7. Huffman, D.A., "Impossible objects as nonsense sentences", *Machine Intelligence*, 6: 295-323 (1971).
8. Clowes, M.B., "On seeing things", *Artificial Intelligence*, 2: 79-116 (1971).
9. Shum, S.S.P., Lau, W.S., Yuen, M.M.F., YU, K.M., "Solid reconstruction from orthographic opaque views using incremental extrusion", *Computer & Graphics*, 21 (6): 787-800 (1997).
10. Falk, G., "Interpretation of imperfect line data as three dimensional scene", *Artificial Intelligence*, 3 (2): 101-144 (1972)
11. Turner, K.J., "Computer perception of curved object using a television camera", Ph.D Thesis, *Edinburg University School of Artificial Intelligence*, Edinburg, 32-38 (1974).
12. Shpitalni, M., Lipson, H., "Identification of faces in a 2D line drawing projection of a wireframe object", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 (10): 1000 - 1012 (1996).

13. Lee, S.H., Haralick, R.M., Zhang, M.C., “Understanding objects with curved surfaces from a single perspective view of boundaries, *Artificial Intelligence*, 26: 409-460 (1985).
14. Malik, J., “Interpreting line drawing of curved objects, *International Journal of Computer Vision*, 1: 73-103 (1987).
15. Şahin, İ., “Temel izdüşümlerden bilgisayar destekli 3B’lu katı modeller oluşturma, Yüksek Lisans Tezi, *G.Ü Fen Bilimleri Enstitüsü*, Ankara, (2001).
16. Idesawa, M., “A system to generate a solid figure from a three view”, *Bull. JSME*, 16: 216-225 (1973).
17. Idesawa, M., Soma, T., Goto, E., Shibata, S., “Automatic input of line drawing and generation of solid figure from three view data”, *Proceedings of the International Joint Computer Symposium*, 304-311 (1975).
18. Sakurai, H., and Grossard, D.C., “Solid model input through orthographic views, *Computer Graphics*, 17 (3): 243-251 (1983).
19. Shin, B.S., Shin, Y.G., “Fast 3D solid model reconstruction from orthographic views, *Computer Aided Design*, 30 (1): 63-76 (1998).
20. Lafue, G., “Recognition of three dimensional objects from orthographic views”, *Computer Graphics*, 10 (2): 103 - 108 (1976).
21. Sutherland, I.E., “Three dimensional data input by tablet, *Proc. IEEE Special Issue on Computer Graphics*, 62 (4): 453-461 (1974).
22. Woo, T.C., Hammer, J.M., “Reconstruction of three dimensional designs from orthographic projections”, *Proc. Ninth CIRP Conference*, Cranfield, 247-255 (1977).
23. Markowsky, G. and Wesley, M.A., “Fleshing out wire frames, *IBM J. Res. Develop.*, 24: 582-587 (1980).
24. Markowsky, G., Wesley, M.A., “Fleshing out projections, *IBM J. Res.& Develop.*, 25: 934-954 (1981).
25. Preiss, K., “Constructing the 3D representation of a plane-faced object from a digitized engineering drawings”, *Fifth International Conference and Exhibition on Computer in Engineering and Building Design*, 142-154 (1980).

26. Preiss, K., "Algorithms for automatic conversion of a 3-view drawing of a plane faced part to the 3-D representation", *Computers in Industry*, 2: 133-139 (1981).
27. Preiss, K., "constructing the solid representation from engineering projections", *Computers and Graphics*, 8: 381-389 (1984).
28. Haralick, R.M., Queeney, D., "Understanding engineering drawings", *Computer Graphics and Images Processing*, 20: 244-258 (1982).
29. Gu, K., Tang, Z., Sun, J., "Reconstruction of 3D objects from orthographic projections", *CG Forum*, 5: 807-811 (1985).
30. Gujar, U.G., Nagendra, I.V., "Construction of 3D solid objects from orthographic views", *Computer&Graphics*, 13 (4): 505-521 (1989).
31. You, C.F., Yang, S.S., "Reconstruction of curvilinear manifold objects from engineering drawings, *Proceedings of the 12th National Conference of the Chinese Society of Mechanical Engineers*, Chia-Yi, 637-646 (1995).
32. Yan, Qing-wen., Chen, C.L.P., Tang, Z., "Efficient algorithm for the reconstruction of 3D objects from orthographic projections", *Computer Aided Design*, 26 (9): 699-717 (1994).
33. Watanabe, T., "Revision of inconsistent orthographic views, *Journal For Geometry and Graphics*, 2 (1): 45-53 (1998).
34. Tanaka, M., Iwama K., Hosoda, A., Watanabe, T., "Decomposition of a 2D assembly drawing into 3D part drawings", *Computer Aided Design*, 30 (1): 37-46 (1998).
35. Tanaka, M., Anthony, L., Kaneeda, T., Hitooka, J., "A single solution method for converting 2D assembly drawings to 3D part drawing, *CAD*, 36: 723-734 (2004).
36. Kuo, M.H., "Reconstruction of quadric surface solids from three-View engineering drawings, *Computer Aided Design* , 30 (7): 517-527 (1998).
37. Kuo, M. H., "Automatic extraction of quadric surfaces from wire-frame Models", *Computer & Graphics*, 25: 109 -119 (2001).
38. Dori, D., Weiss, M., A., "Scheme for 3d object reconstruction from dimensioned orthographic views, *Engineering Applications of Artificial Intelligence*, 9 (1): 53-64 (1996).

39. Wei, X., "Computer vision method for 3D quantitative reconstruction from a single line drawing", Phd Thesis, *Beijing University Department of Mathematics*, China, (1987).
40. Sugihara, K., "Machine interpretation of line drawings", *MIT Press.*, London, 125-138 (1986).
41. Lamb, D., Bandopadhyay, A., "Interpreting a 3D object from a rough 2D line drawing", *Proceeding of Visulation '90*, 59-66 (1990).
42. Gong, J. H., Zhang, G. F., Zhang, H., Sun, J. G., "Reconstruction of 3D curvilinear wire-frame from three orthographic views", *Computers & Graphics*, 30: 213-224 (2006).
43. Gong, J. H., Zhang, G. F., Zhang, H., Sun, J. G., "Solid reconstruction using recognition of quadric surfaces from orthographic views", *Computers Aided Design*, 38: 821 – 835 (2006).
44. Oh, B. S., Kim, C.H., "Systematic reconstruction of 2D curvilinear object from two view drawing", *Computer & Graphics*, 23: 343-352 (1999).
45. Zhang A. Z. C., "Reconstructing wireframe model of curvilinear objects from three orthographic views", *Journal of Southwest Jiaotong University*, 9 (2): 122-129 (2001).
46. Şahin, İ., Börklü, H.R., "2B izdüşüm görünüşlerinden otomatik 3B sınır temsili modeller oluşturma", *G.Ü Fen Bilimleri Enstitüsü Dergisi*, 14 (4): 1455-1465 (2001).
47. Liu, SX, Hu, SM, Sun JG, "Two accelerating techniques for 3D reconstruction", *Journal Of Computer Science And Technology*, 17 (3): 362 – 368 (2002).
48. Aldefeld B., "On automatic recognition of 3D structures from 2D representation", *Computer Aided Design*, 15 (2): 59-64 (1983).
49. Aldefeld, B., Richter, H., "Semi automatic three dimensional interpretation of line drawing", *Computer&Graphics*, 8 (4): 371-380 (1984).
50. Mekhabucakij, K., Tamamoto T., and Aoki I., "A CSG-Based interpretation of three-view drawings", *Trans. IECE Japan*, 12: 1354-1364 (1986).
51. Ho, B., "Inputting constructive solid geometry representations directly from 2D orthographic engineering drawings", *CAD*, 18 (3): 147-155 (1986).

52. Wang W., Grinstein, G., "A polyhedral object's CSG-Rep construction from a 2D single drawing", *Proc. Of 1989 SPIE Intelligent Robots and Computer Vision III: Algorithms and Techniques*, Boston, 1192: 230-238 (1989).
53. Wang, W.D., "Regular curved object's CSG-Rep reconstruction from a single 2D line drawing", *Proc. Of 1991 SPIE Intelligent Robots and Computer Vision X: Neural Biological and 3D Methods*, Boston, 1608 (1991).
54. Wang, W., "On the automatic reconstruction of a 3D object's constructive solid geometry representation from its 2D line drawing", D.Sc Thesis, *University of Massachusetts*, Lowell, 124-146 (1992).
55. Chen, Z., Perng, D.B., "Automatic reconstruction of 3D solid objects from orthographic views", *Pattern Recognition*, 21: 439-449 (1988).
56. Nagasami, V., Langrana N. A., "Reconstruction of three-dimensional objects using a knowledge-based environment, *Engineering with Computers*, 7 (1): 254 – 271 (1991).
57. Hosomura, T., "Reconstruction of 3D solid objects from 2D orthographic views", *Proc. SPIE: Videometrics IV*, 2598: 106-113 (1996).
58. Dutta, D., Srinivas, Y.L., "Reconstruction of curved from the polygonal orthographic views", *Computer Aided Design*, 24 (3): 149-159 (1992).
59. Wu, M.C., Lin, M.S., "Automatic construction of CSG solids from a single isometric drawing", *International Journal of Computer Integrated Manufacturing*, 9 (21): 1 - 21 (1996).
60. Tam, K. S., Atkinson, J., "An approach for creating solid models from orthogonal views by identification of boolean operations", *Journal of Materials Processing Technology*, 138: 163-169 (2003).
61. Suh, T.J., Kim W.S., Kim, C.H., "Two phase 3D object reconstruction from two-view drawings", *IEICE Transactions On Information and Systems*, 7: 1093-1100 (1999).
62. Zhao, H., Kim, Y.S., "Geometric operations for visual reasoning of a solid from orthographic projections", *Advanced in Engineering Software*, 30: 445-450 (1999).

63. Hubbard, C., Kim, Y. S., “Geometric assistance for visual reasoning and construction of solids with curved surfaces from 2D orthographic views”, *Advanced in Engineering Software*, 32: 21-35 (2001).
64. Dimri, J., Gurumoorthy, B., “Handling sectional views in volume based approach to automatically construct 3D solid from 2D views”, *CAD*, 37: 485-495 (2005).
65. Shum, S.S.P., Lau, W.S., Yuen, M.M.F., Yu, K.M., “Solid reconstruction from orthographic views using 2-Stage extrusion”, *Computer Aided Design*, 33: 91-102 (2001).
66. Shum, S.S.P., Lau, W.S., Yuen, M.M.F., Yu, K., “Cost effective solid reconstruction from an x-ray image”, *Journl of Materials Processing Technology*, 121: 207-216 (2002).
67. Şahin, İ, Börklü H. R., “İzdüşümlerden hacme dayalı 3B’lu nesneler elde etmede dahili elemanların kullanımı, *G.Ü. Fen Bilimleri Dergisi*, 15 (2): 351-361 (2002).
68. Şahin, İ, Börklü H. R., “Teknik resimleri verilen parçaların bilgisayar destekli 3B modellerinin elde edilmesi: Hacimsel bir yaklaşım”, *Mühendis ve Makina* , 510: 37-45 (2002).
69. Çiçek A, Gülesin M., “2 boyutlu çizimlerden 3 boyutlu katı modellerin otomatik elde edilmesi”, *Gazi Üniv. Müh. Mim. Fak. Der.*, 20 (3): 387-394 (2005).
70. Liu, C.H., Perng, D.B., Chen, Z., “Automatic form feature recognition and 3d part reconstruction from 2D Cad data”, *Computer&Industrial Engineering*, 26 (4): 689-707 (1994).
71. Lee, H., Han, S., “Reconstruction of 3D solids revolution from 2D orthographic views”, *CAD*, 37: 1388 – 1398 (2005).
72. Sahin, İ., Borklu H. R., “Üç temel görünüşü verilen nesnelerin AutoCAD ortamında otomatik 3B modellerini oluşturma”, *Makine Tasarımı ve İmalat Dergisi*, 5 (1): 9 – 22 (2003).
73. Sahin İ., Börklü, H. R., “Hacimsel kesiştirme yaklaşımı kullanarak silindirik parçaların katı modellerini oluşturma”, *Politeknik Dergisi*, 10 (1): 71-77 (2007).

74. Masuda, H., Masayuki, N., “A cell-based approach for generating solid objects from orthographic projections”, *Computer Aided Design*, 29 (3): 177-187 (1997).
75. Wang, Z., Latif, M., “Reconstruction of a 3D solid model from orthographic projections”, *International Conference on Geometric Modeling and Graphics (GMAG'03)*, 75 – 87 (2003).
76. Geng, W., Wang, J., Yiang, Z., “Embedding visual cognition in 3D reconstruction from multi-view engineering drawings”, *CAD*, 34: 321-336 (2002).
77. Allahverdi, N., “Uzman sistemler – Bir Yapay Zeka Uygulaması”, *Atlas Yayın Dağıtım*, İstanbul, 1-13, 15-23, 71-93 (2002).
78. Shank, R.C., “What is anyway?”, *AI Magasine*, 8 (4): 12-18 (1987).
79. Başak, H., “Prizmatik parçalar için unsur tabanlı parametrik tasarım”, Doktora Tezi, *G.Ü Fen Bilimleri Enstitüsü*, Ankara, 70-80 (2000).
80. Medsker, L., Leibowits, J., “Design and development of expert systems and neural networks, *Macmillan*, Newyork, 48-60 (1994).
81. Aslan, E., “Uzman sistem yaklaşımıyla son işlemci tasarımı”, Doktora Tezi, *G.Ü Fen Bilimleri Enstitüsü*, Ankara, 20-35 (1995).
82. Benzer, R., “Zeki karar sistemleri ve bazı askeri uygulamalar”, Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, 14-22(1998).
83. Dologite, D.G., “Developing knowledge based systems using VP-Expert”, *Prentice Hall*, Newyork, 108-118 (1993).
84. Martin, J., Oxman, S., “Building expert systems”, *Prentice Hall*, New Jersey, (1988).
85. Davies, P.B., “Expert database system a gentle introduction”, *McGraw-Hill*, New York, 34-51 (1991).
86. Kumar, B., “Knowledge processing for structural design topics in engineering”, *WIT Press*, 25: 5-19 (1995).
87. Paszek, R.K., “The method of the knowledge representation in an expert system for metal cutting engineering”, *Journal of Material Processing Technology*, 64: 319-326 (1997).

88. Turban, E., "Decision support and expert systems", *Macmillan Pub. Comp.*, New York, 1-80 (1990).
89. Rolston W.D., "Principles of artificial intelligence and expert system development", *McGraw-Hill*, New York, 156-219 (1985).
90. Bowerman, R.G., Glover, D.E., "Putting expert systems into practice", *Van Nostrand Donihold Company*, 135-142 (1988).
91. Edmunda, C., Rober, C., "Developing expert systems", *John Willy Inc. Engineers*, Newyork, 44: 67-73 (1992).
92. Pham, D.T., Pham, P.T.N., "Expert system in mechanical and manufacturing engineering", *International Journal Advanced Manufacturing Technology*, 3 (3): 3-21 (1988)
93. Burgut, H., "Uzman sistem ve otomotiv sektöründeki bir uygulaması", Yüksek Lisans Tezi, *İTÜ Fen Bilimleri Enstitüsü*, İstanbul, 9-26 (1999).
94. Poyrazoğlu, O., Gülesin, M., Kayır, Y., "Uzman sistem tekniklerine dayalı alışılmamış imalat yöntemlerinin tespiti", *Makine Tasarım ve İmalat Yöntemleri Konferansı*, Ankara, 80-89(1997).
95. Sümen, H., "Otomasyon yazılım yönü: uzman sistemler", *Otomasyon*, 28: 108-116 (1994).
96. Harmon, P., King, D., "Expert systems", *Wiley*, Newyork, 65-82(1985).
97. Özdemir, A., Aslan, E., Şeker, U., "Yapay zeka ve uzman sistemler", *G.Ü Tekink Eğitim Fakültesi Dergisi*, 3: 91-97 (1993).
98. Alto, A., Dassisti, M., Galantucci, D., "An expert system for reliable tool-replacement policies in metal cutting", *Journal of Engineering for Endustry*, 116: 405-407 (1994).
99. Jang, H.S., Baghci, A., "Tool selection in machining by integration of a data base and rule based system", *Journal of Mechanical Working Technology*, 20: 25-34 (1989).
100. Gül Gökay, E., Çağatan, T., "Genetik algoritmalar ve uygulama alanları", *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fak. Dergisi*, 21(1): 129-152 (2002).

101. Toktaş, İ., “Mekanik sistemler için bir kavramsal tasarım modelinin geliştirilmesinde yapay sinir ağlarının kullanılması”, Doktora Tezi, **G.Ü Fen Bilimleri Enst.**, Ankara, 15-25 (2003).
102. Lemley, B., “Machines that think”, **Discover**, 22: 74-79 (2001).
103. Goldberg, D.E., “Genetic algorithms in search, optimisation and machine learning”, **Addison-Wesley Publishing Comp.**, 327-349 (1989).
104. Chambers, L., “Practical handbook of genetic algorithms”, **CRC Press.**, New York, 122-140 (2001).
105. Elmas, Ç., “YSA kuram, mimari, eğitim, uygulama”, **Seçkin Yayınları**, Ankara, 1-15, 42-66 (2002).
106. Engin Ö., Fırlı, A., “Akış tipi çizgeleme problemlerinin genetik algoritma yardımı ile çözümünde uygun çaprazlama operatörünün belirlenmesi”, **Doğuş Üniversitesi Dergisi**, 6: 27-35 (2002).
107. Michalewicz, Z., “Genetic algorithms + data Structures = evolution programs”, **Springer – Verlag**, New York, 123 – 132 (1992).
108. Forest, S., “Genetic algorithms: principle of natural selection applied to computation”, **Science**, 261: 872-878 (1993).
109. Öztemel, E., “Yapay zeka teknolojisi”, **Otomasyon**, 46: 130-132 (1996).
110. Jang, J.S.R., “Derivative free optimisation”, **Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence**, **Prentice-Hall**, 114-167 (1997)
111. Dalgıç, T., “Yapay zeka, bilinç ve tasarım”, Yüksek Lisans Tezi, **Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü**, İstanbul, 18-22 (2001).
112. Chan, F.T.S., Mak, K.L., Luong, L.H.S., Ming, X.G.I., “Machine component grouping using genetic algorithms”, **Robotic and Computer-Integrated Manufacturing**, 14: 339-346 (1998).
113. Brokhus, C.G., “Shortest path optimisations using a genetic search technique and simulation”, **Pros. Of the 14.th Annual Pittsburgh Conference**, Pittsburgh, 230-241 (1983).

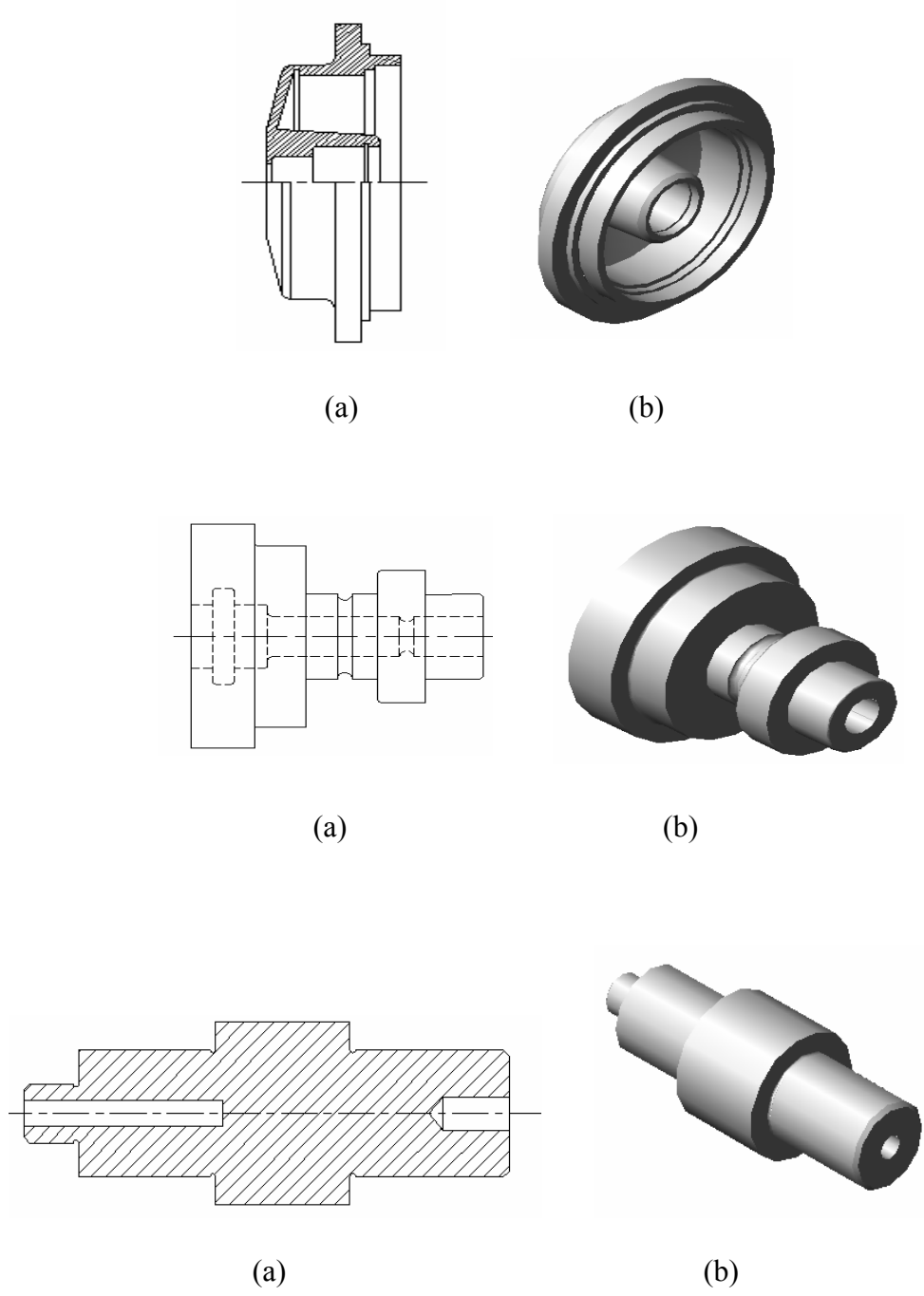
114. Gonzales, E., Fernandez, M.A.R., “Genetic optimisation of a fuzzy distribution model”, *International Journal of Physical Distribution & Logistics Menagement*, 30 (7-8): 681-696 (2000).
115. Öztemel, E., “Bir toplumda yönetim bilişim sistemlerinin gelişimi ve yapay zeka”, *Otomasyon*, mart: 72-74 (1998).
116. Schutte, J.F., Pevear, D.R., “Matchmod: a genetic algorithm to interpret x-ray diffraction patterns”, *In Artificial Intelligence in The Petroleum Industry Symbolic and Computational Applications*, 14: 397-415 (1995).
117. Karanfil, S., “Fuzzy lojik problemlerinde üyelik fonksiyonunun belirlenmesinde deneysel verilere dayanarak bir yöntem geliştirilmesi”, Doktora Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 3-21 (1997).
118. Crick, F., “Şaşırtan varsayım”, *TÜBİTAK*, Ankara, 13-24 (1997).
119. Rumelbelhart, E., Hinton, G.E., Williams, R.J., “Learning representations by back-propagation errors”, *Nature*, 313: 533-536 (1986).
120. Öztemel, E., “Integrating expert systems and neural networks for intelligent on-line statistical process control”, PhD Thesis, *School of Electrical, Electronic and Systems Engineering*, Cardiff, 67-98 (1992).
121. Ross, T.J., “Fuzzy logic with engineering applications”, *McGraw-Hill*, New York, 371-402 (1995).
122. Zadeh, L.A., “Fuzzy sets”, *Informal Control*, 8:338-353 (1965).
123. Aydemir, A., “Sonlu eleman gerilme analizinde fuzzy mantığı”, Doktora Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, 8-16 (2001).
124. Atacak, İ., Bay, Ö. F. “Bulanık mantık denetimli seri aktif güç filtresi kullanarak harmonik gerilimlerin bastırılması”, *Gazi Univ. Müh. Mim. Fak. Dergisi*, 19 (2): 205-215 (2004).
125. Nalbant, M., “Bilgisayarla bütünleşik imalat”, *Beta Yayınları*, 120-125(1997).
126. Suh, Y.S., “A feature-conversion CAD system for the concurrent engineering environment”, P.hD Thesis, *Rwnsselear Polytechnic Instute Troy*, Newyork, 20-48 (1995).
127. Chuang, S.H., Hendersan, M.R., “Three dimensional shape pattern recognition using vertex classification and vertex edge graphs”, *Computer Aided Design*, 22

- (6): 377-385 (1990).
128. Kertlioğlu, “Üretim için tasarım”, Yüksek Lisans Tezi, **G.Ü. Fen Bilimleri Enstitüsü**, Ankara, 62-64 (1998).
 129. Başak, H., “Bilgisayar destekli tasarım (BDT) ve bilgisayar destekli üretimde (BDÜ) unsur kullanımı”, **Fen Bilimleri Enstitüsü Dergisi**, 12 (2): 255-270 (1999).
 130. Henderson M. R., Probhakar S., “Automatic form feature recognition using neural network based techniques on boundary representation of solid model, **Computer Aided Design**, 24 (7): 391-393 (1994).
 131. Zeid, I., “Mastering CAD/CAM”, **Mac Graw Hill**, New York, 43-54 (2004).
 132. Çelik, S., Börklü, H.R., “Dönel makine parçalarının eksenel simetrik unsurlarla modellenmesi”, **Makine Tasarım ve İmalat Dergisi**, 3, (4): 174-183 (1998).
 133. Shah, J.J., Mantyla, M., “Parametric and feature based CAD/CAM”, **John Wiley & Sons**, 140 – 145 (1995).
 134. Eldem, C., “Dönel geometrik unsurlarla parça modelleme ve CNC kodu türetme”, Doktora Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 24-41 (2001).
 135. Van Emmerik G. M., Jonsen, F. W., “User interface for feature modeling computer application in production and engineering”, **Elsevier, IFIP**, 84-92 (1989).
 136. Patel R. M., Mclead A. J., “Engineering feature description in mechanical engineering design”, **Computer Aided Engineering Journal**, 5 (5): 180-183 (1988).
 137. Yıldız, Y., “Dönel parçalar için unsur tabanlı bir CAM programı hazırlanması ve diğer programlama yöntemleri ile karşılaştırılması”, Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 27-34 (2004).
 138. Rinderle J. R., Suh N. P., “Measurer of functional coupling in design, **J. of Engineering For Industry**, 104: 383-388 (1986).
 139. Çölkesen, R., “Veri yapıları ve algoritmalar”, **Papatya yayıncılık Eğitim**, İstanbul, 22-29 (2006).

140. Güngören, B., “UML ile nesne tabanlı çözümleme ve tasarım”, *Seçkin Yayıncılık*, Ankara, 32-39 (2005).
141. Al-Ahmad, W., “A framework for conceptual modeling in OOP”, *Journal of the Franklin Institute*, 343: 532–544 (2006).
142. Nowack, P., “Conceptual modeling for ubiquitous systems”, *in: Proceedings of ECOOP*, 125-147 (2004).
143. De Lillo, N. J. “Object oriented design”, *Thomson Brooks / Cole*, Belmont, 2-13, 56-79 (2004)
144. Wille, A.P., B, Shom., P, Hammes., Sur, S.S., “On the performance of pure and impure paralel functional programs”, *Paralel Computing*, 25: 1723-1740 (1999).
145. S, Syed., M, Deepak, S, Shukla, “A functional programming framework for latency insensitive protocol validation”, *Electronic Notes in Theoretical Computer Science*, 146: 169-188 (2006).
146. Stark, R.F. “Axiom for strict and lazy functional programs”, *Annals of Pure and Applied Logic*, 133: 293-318 (2005).
147. Baransel, C., Mumcuoğlu, A., “Web tabanlı, üç katmanlı yazılım mimarileri: UML, EJB ve ORACLE ile sistem modelleme tasarım ve gerçekleştirim”, *SAS Yayınları*, Ankara, (2003).
148. TSE 88, Teknik Resim Genel kurallar.

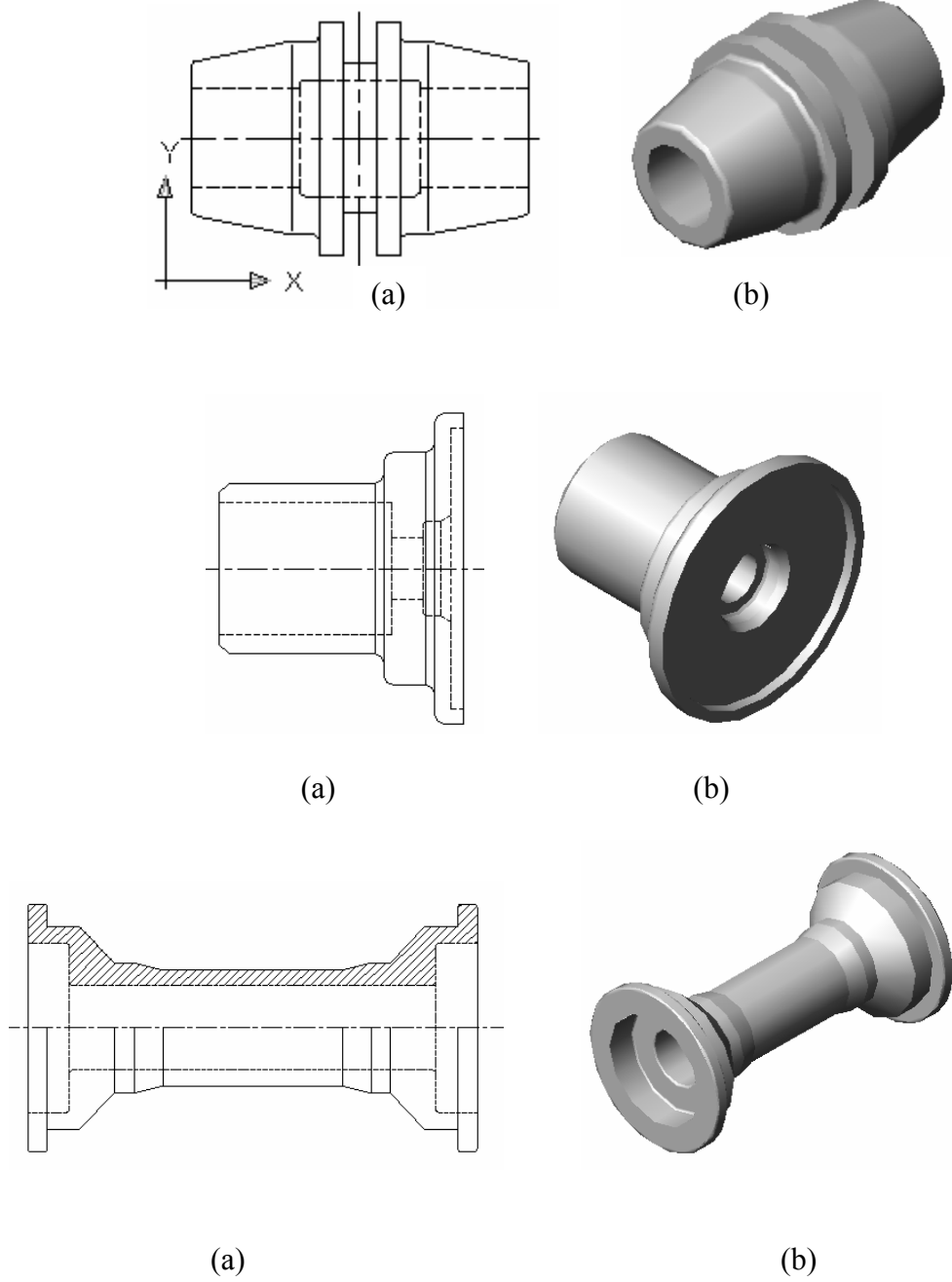
EKLER

EK - 1 Örnek Parçalar



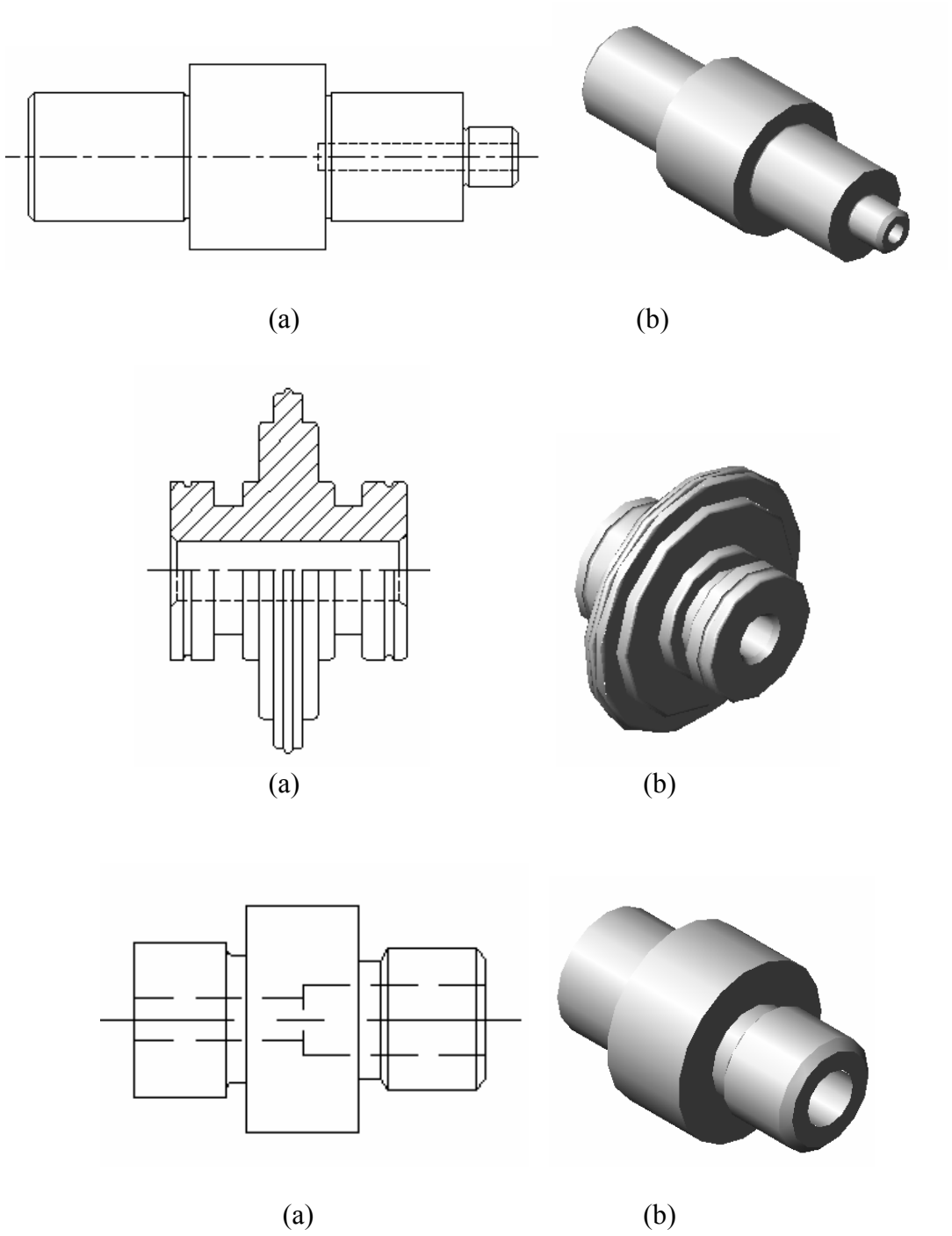
Şekil 1.1. Tek görüşlü parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar



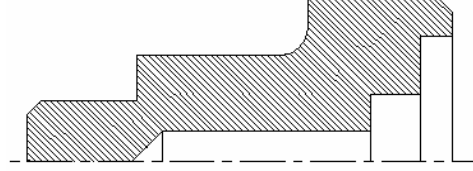
Şekil 1.2. Tek görünüşlü parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar

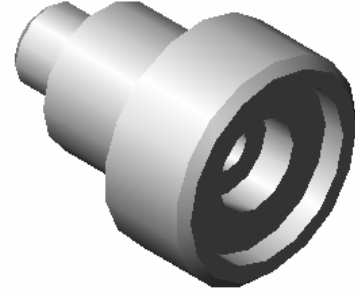


Şekil 1.3. Tek görüşlü parça örnekleri (a) 2B görünüş (b) katı model

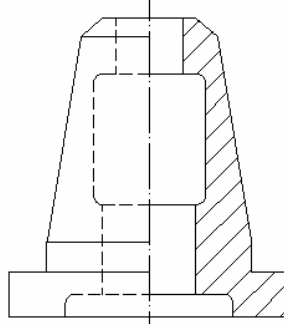
EK - 1 (Devam) Örnek Parçalar



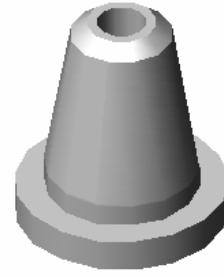
(a)



(b)



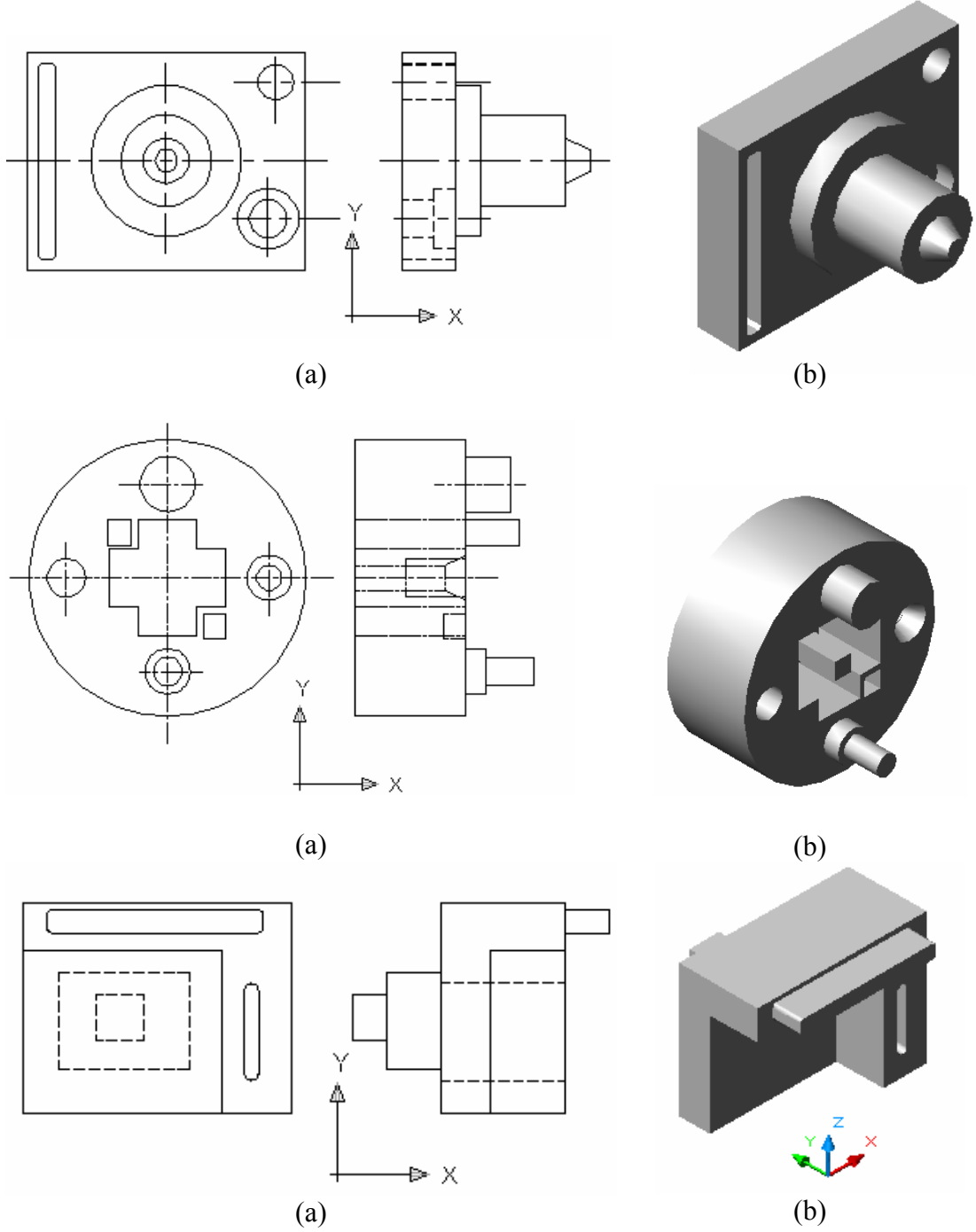
(a)



(b)

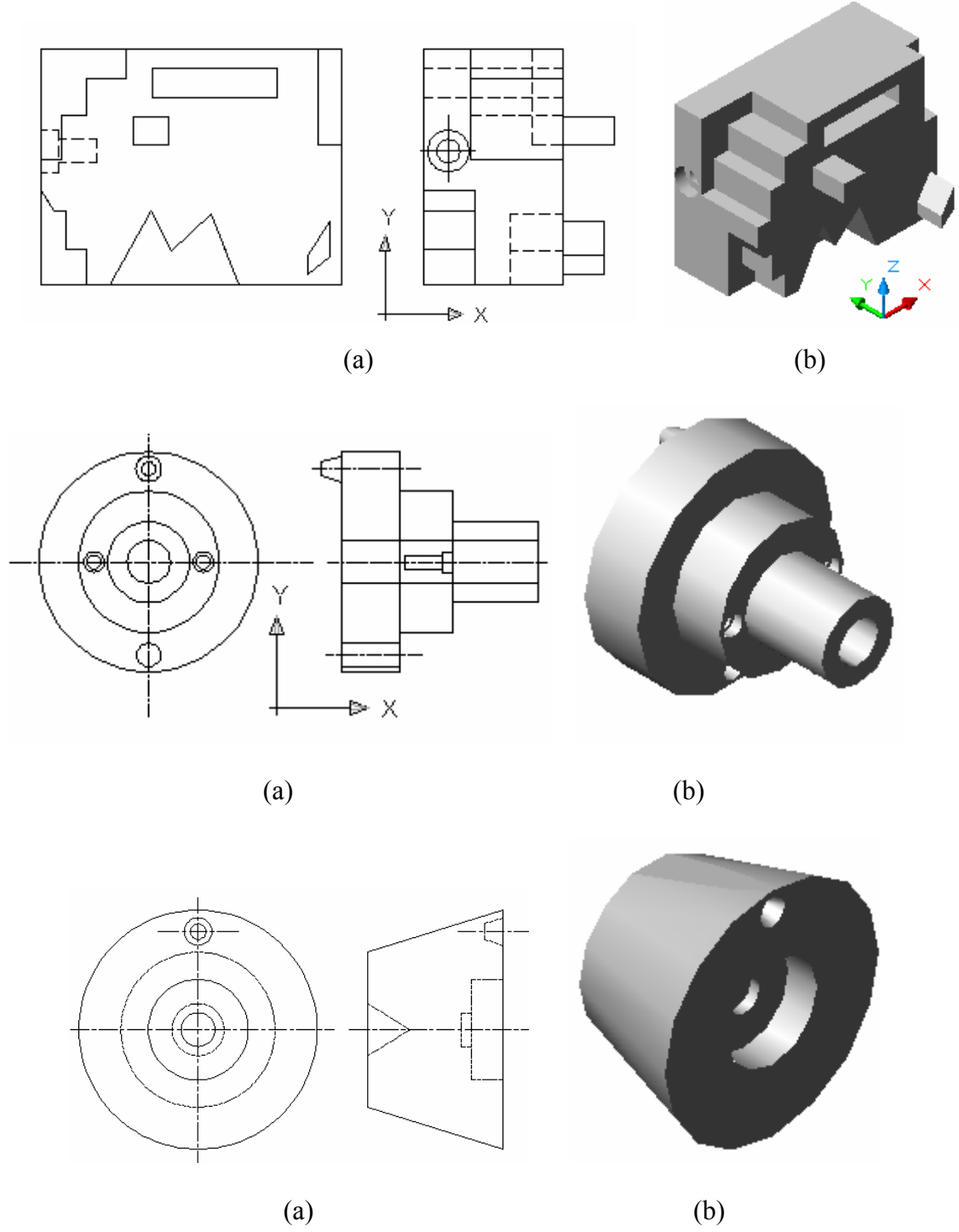
Şekil 1.4. Tek görünüşlü parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar



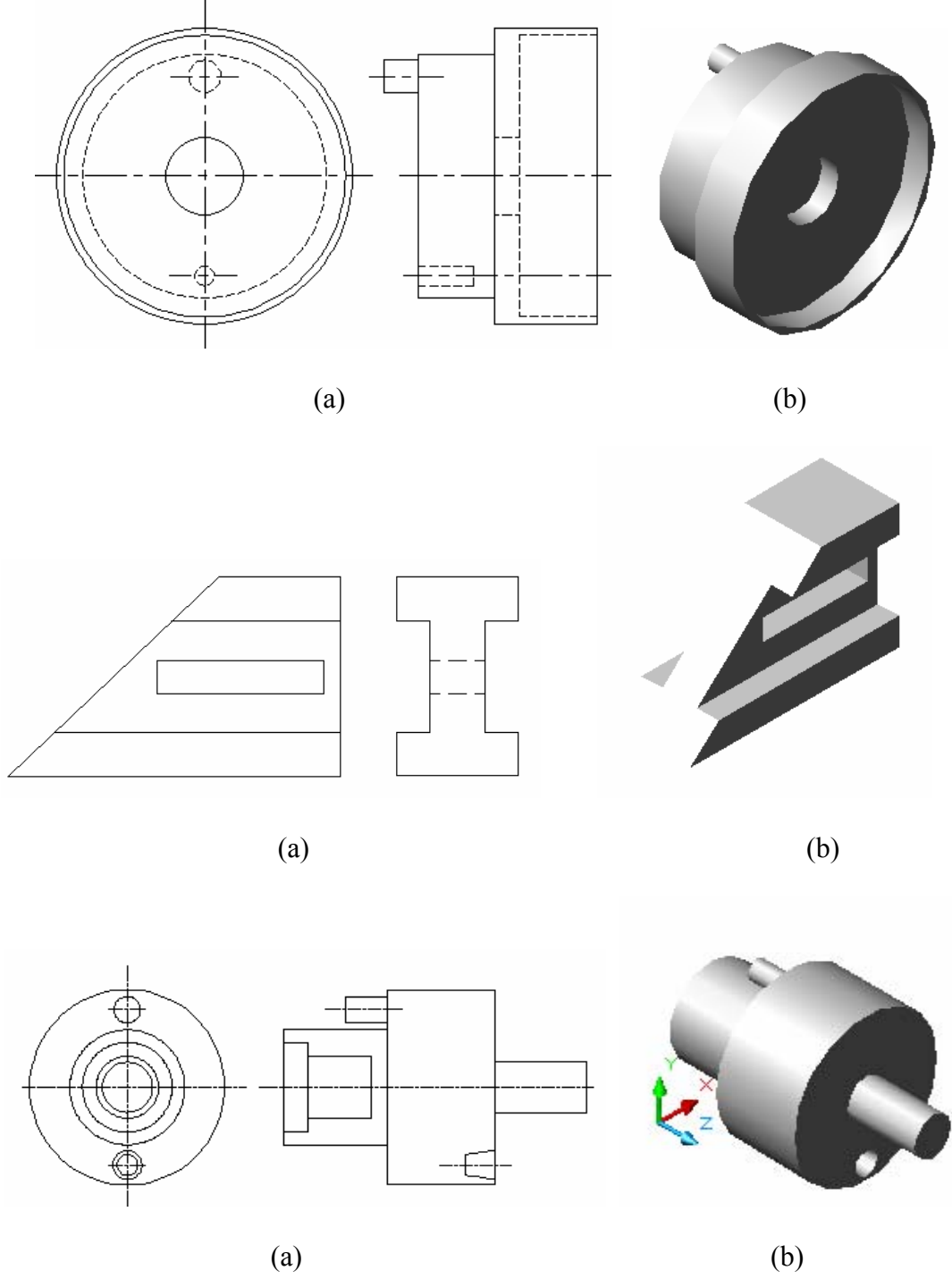
Şekil 1.5. İki görünüş parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar



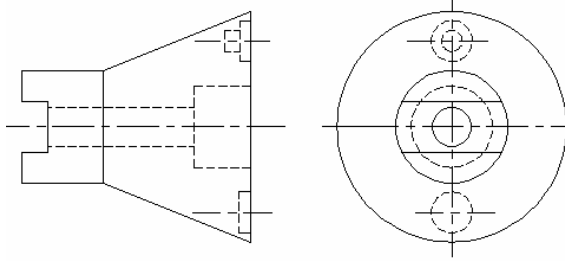
Şekil 1.6. İki görünüş parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar

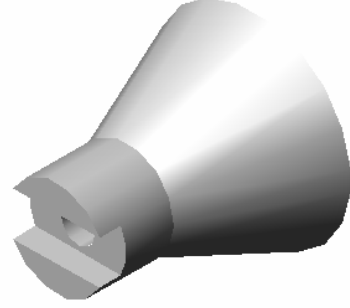


Şekil 1.7. İki görünüş parça örnekleri (a) 2B görünüş (b) katı model

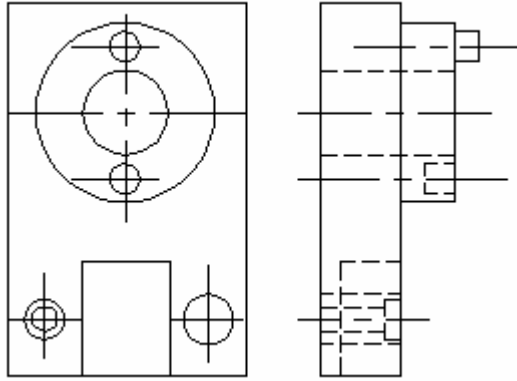
EK - 1 (Devam) Örnek Parçalar



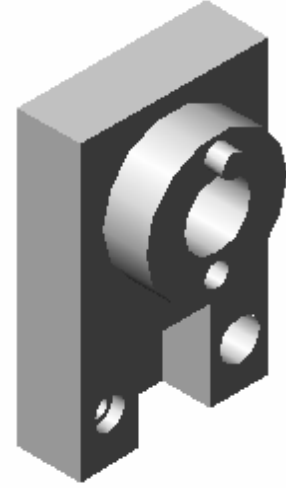
(a)



(b)



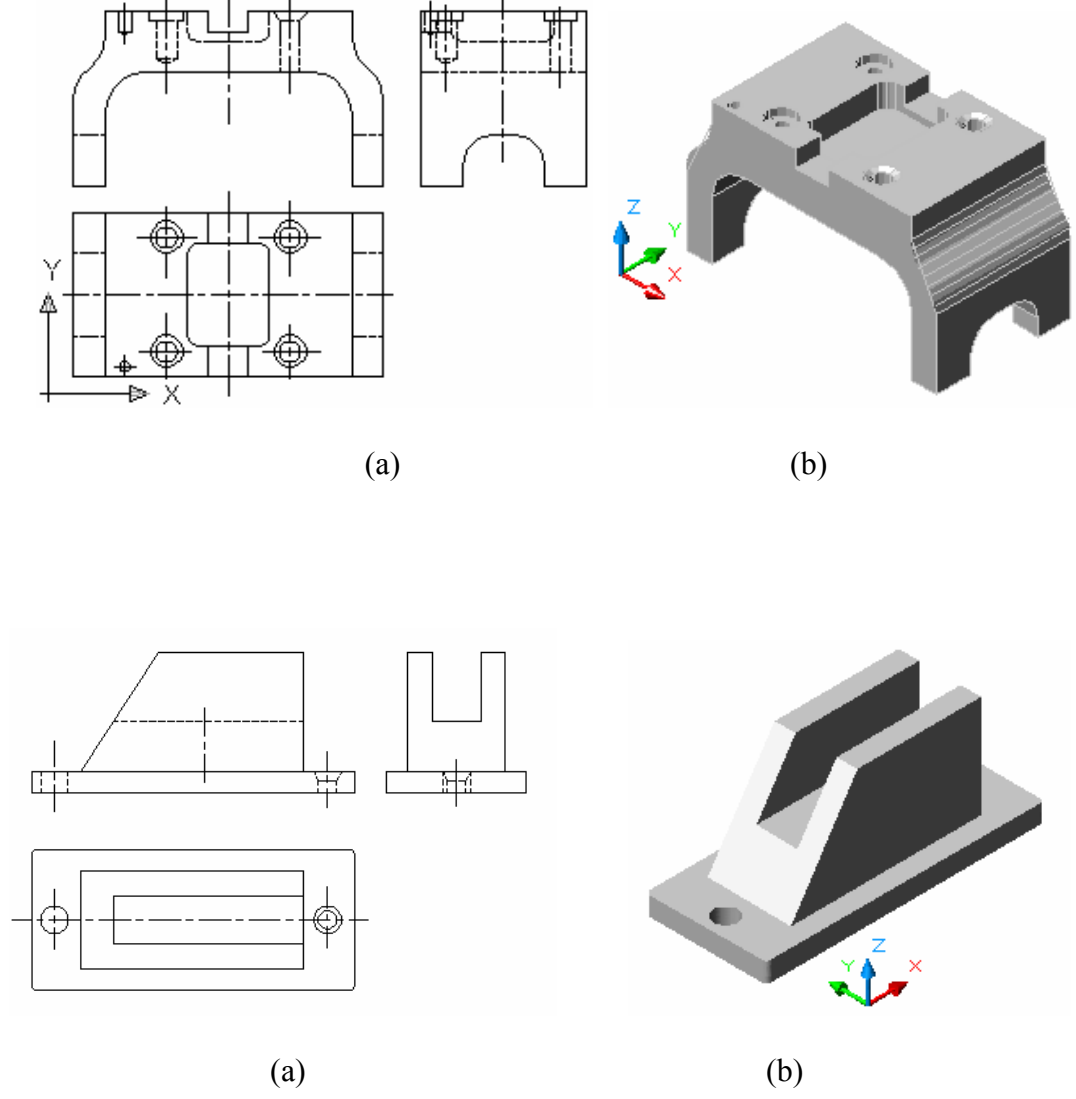
(a)



(b)

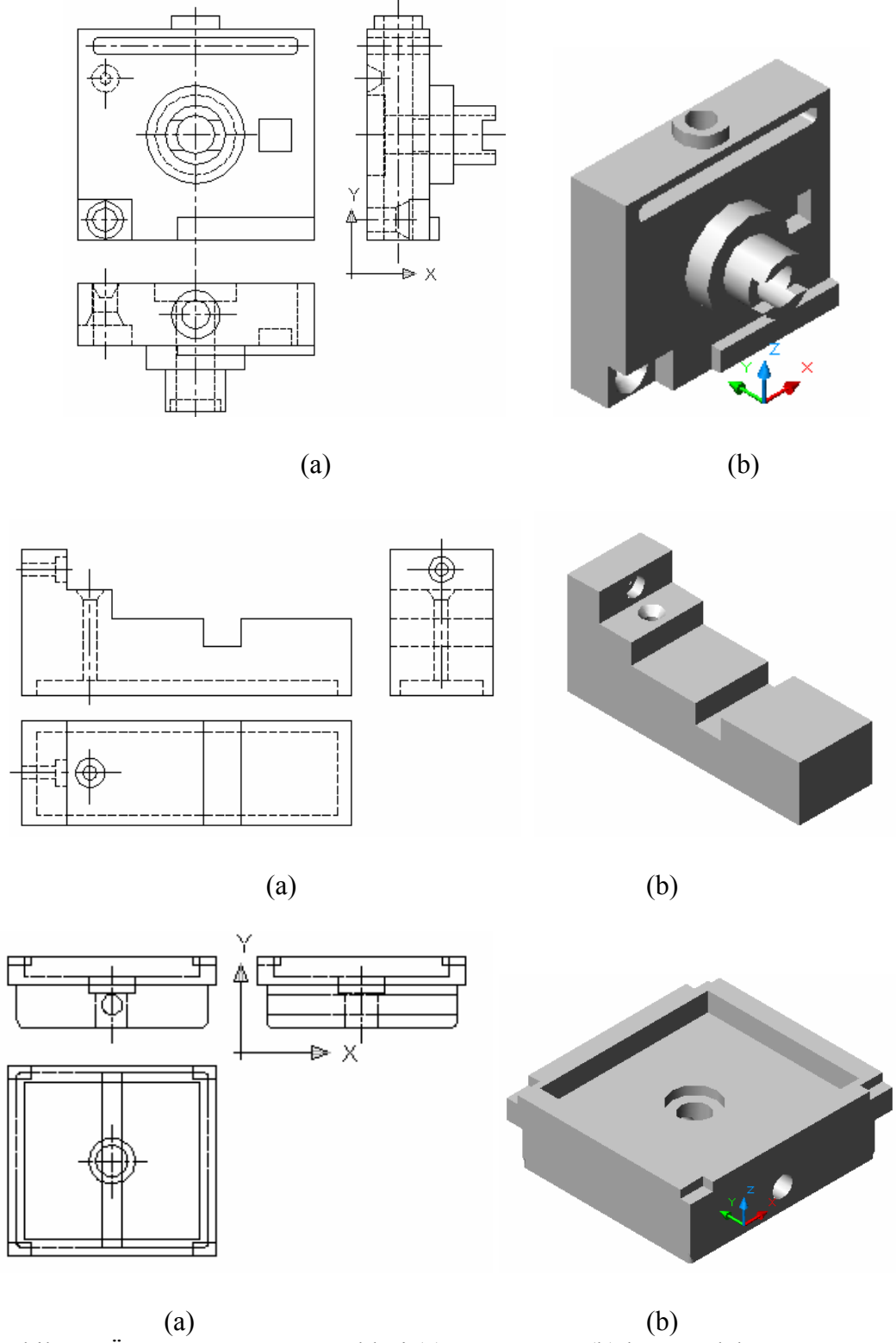
Şekil 1.8. İki görünüş parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar



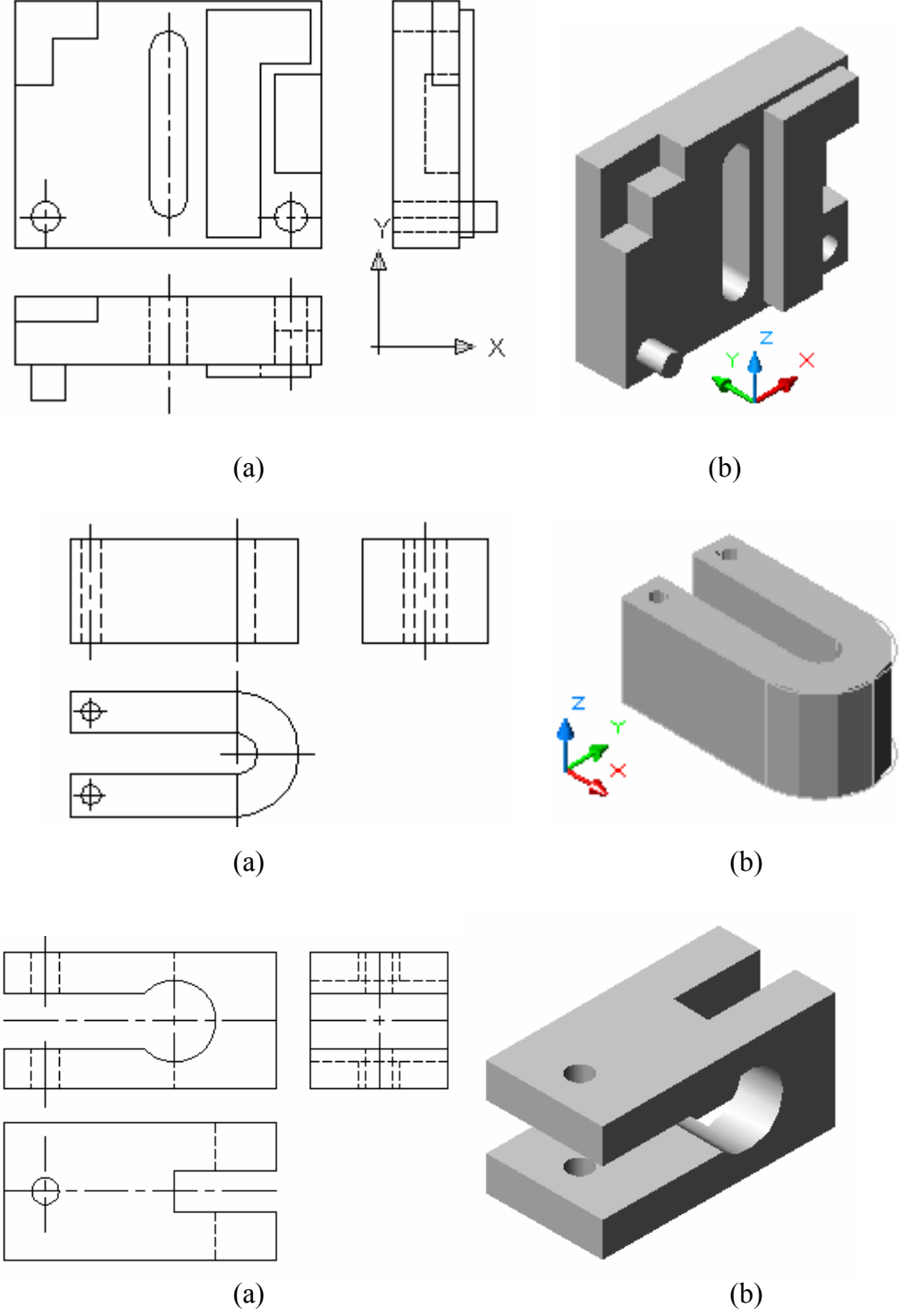
Şekil 1.8. Üç görünüş parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar



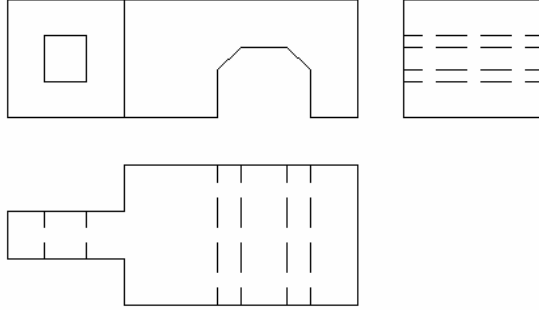
Şekil 1.9. Üç görünüş parça örnekleri (a) 2B görünüş (b) katı model

EK - 1 (Devam) Örnek Parçalar

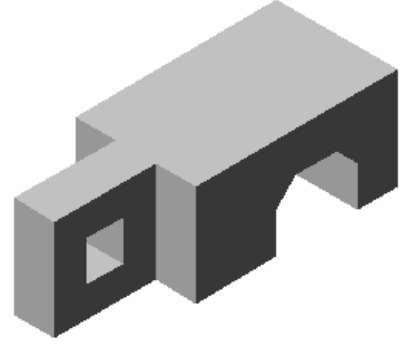


Şekil 1.10. Üç görünüş parça örnekleri (a) 2B görünüş (b) katı model

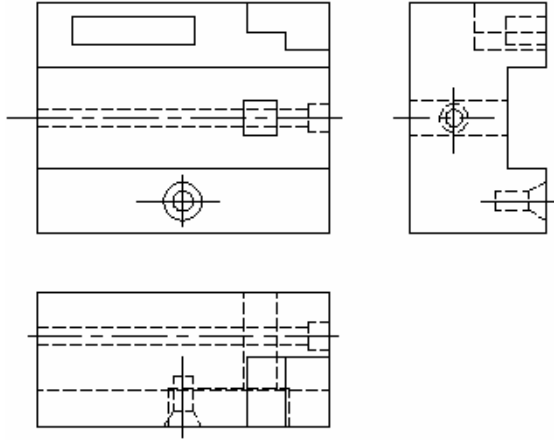
EK - 1 (Devam) Örnek Parçalar



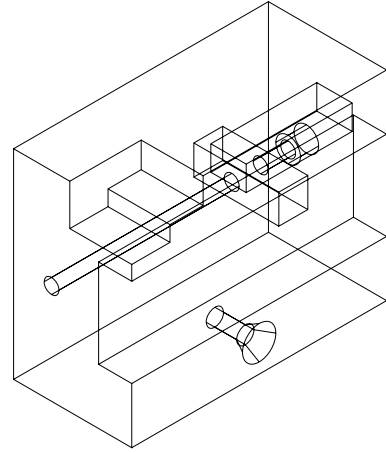
(a)



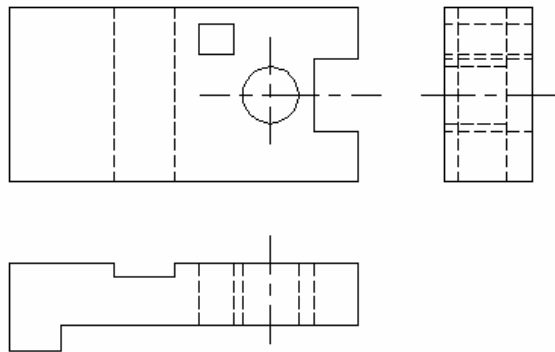
(b)



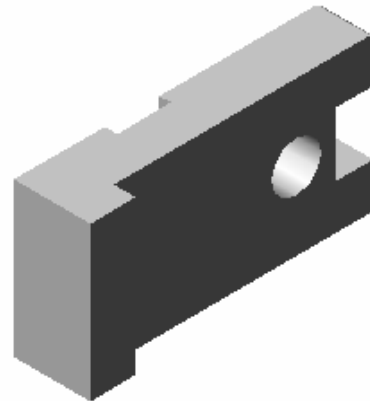
(a)



(b)



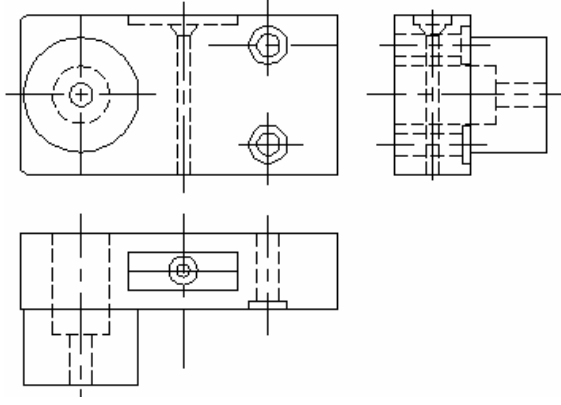
(a)



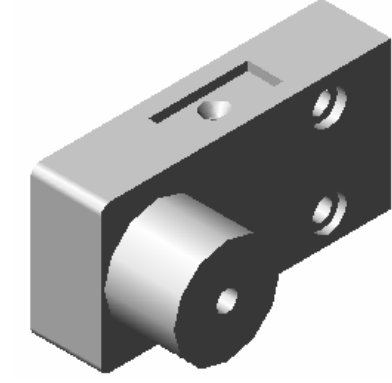
(b)

Şekil 1.11. Üç görüşlü parça örnekleri (a) 2B görünüş (b) katı model

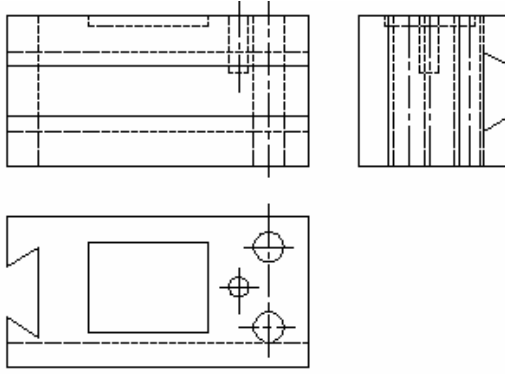
EK - 1 (Devam) Örnek Parçalar



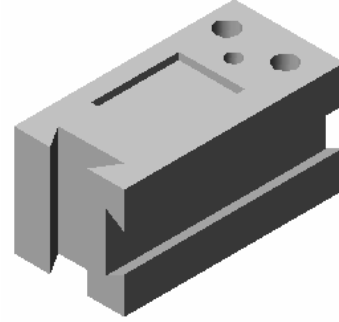
(a)



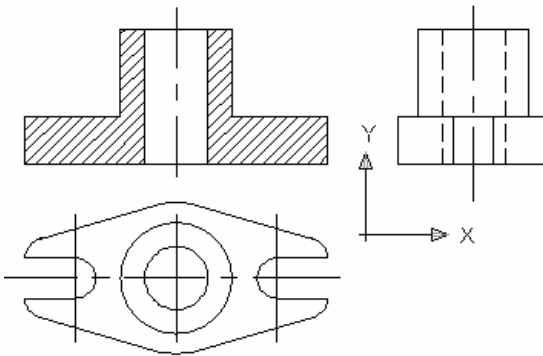
(b)



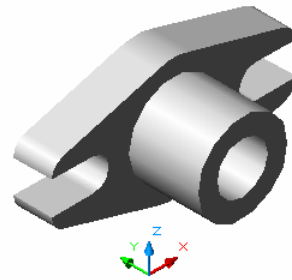
(a)



(b)



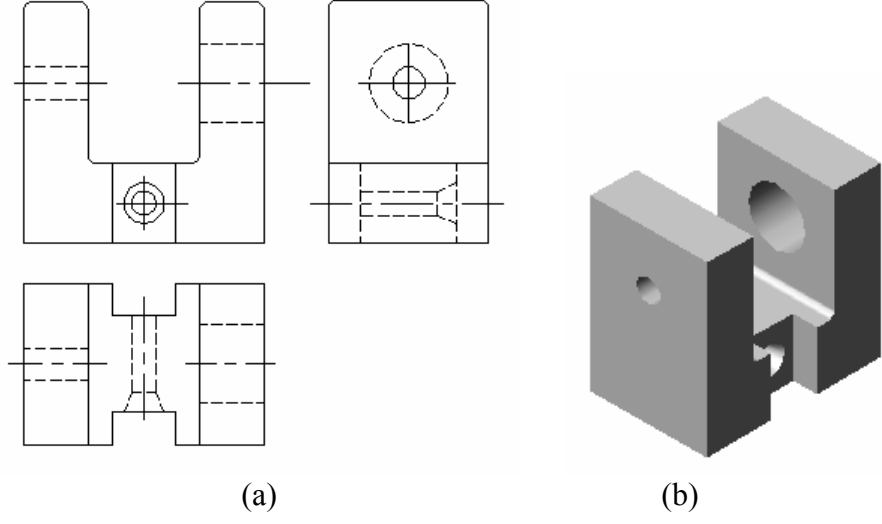
(a)



(b)

Şekil 1.12. Üç görüşlü parça örnekleri (a) 2B görüş (b) katı model

EK - 1 (Devam) Örnek Parçalar



Şekil 1.13. Üç görüşlü parça örnekleri (a) 2B görüş (b) katı model

EK – 2 Örnek Kurallar

Kural_1

parca_prizmatik
 kapalı_katı
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sınırları_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 boydanboya_prizmatik_delik

Kural_5

parca_prizmatik
 kapalı_katı
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sınırları_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_buyuk
 boydanboya_prizmatik_delik

Kural_6

parca_prizmatik
 kapalı_katı
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sınırları_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_kor_delik

Kural_7

parca_prizmatik
 kapalı_katı
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur

izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_havsali_boydanboya_delik

Kural_14
 parca_prizmatik
 kapali_kati
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_buyuk
 prizmatik_delik

Kural_17
 parca_prizmatik
 kapali_kati
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sinirlari_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_buyuk
 prizmatik_boydanboya_delik

Kural_25
 parca_prizmatik
 kapali_kati
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_cikinti

Kural_49
 parca_prizmatik
 açık_katı
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sınırları_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_kor_delik

Kural_74
 parca_prizmatik
 açık_katı
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_buyuk
 prizmatik_cikinti

Kural_79
 parca_prizmatik
 açık_katı
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_havsali_cikinti

Kural_134
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunmez

izdusum_anaparca_sınırları_icinde
kesit_yok
izdusum_cift_kose
temel_kati_boyu_anakatidan_kucuk
silindirik_kor_delik

Kural_136
parca_silindirik
daire
girdi_on_gorunuste
girdi_cizgi_tipi_gorunmez
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sınırları_icinde
kesit_var
izdusum_cift_kose
temel_kati_boyu_anakatiyla_esit
silindirik_boydanboya_delik

Kural_137
parca_silindirik
daire
girdi_on_gorunuste
girdi_cizgi_tipi_gorunmez
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sınırları_icinde
kesit_var
izdusum_cift_kose
temel_kati_boyu_anakatidan_kucuk
silindirik_kor_delik

Kural_139
parca_silindirik
daire
girdi_on_gorunuste
girdi_cizgi_tipi_gorunmez
izdusum_cizgi_tipi_gorunmez
izdusum_anaparca_sınırları_icinde
kesit_yok
izdusum_tek_kose
temel_kati_boyu_anakatiyla_esit
silindirik_boydanboya_havsali_delik

Kural_142
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcasınırıları_ıinde
 kesit_var
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_boydanboya_havsali_delik

Kural_145
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_cikinti

Kural_151
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_havsali_cikinti

Kural_154
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti

kesit_var
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_havsali_cikinti

Kural_158
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_cikinti

Kural_161
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_cikinti

Kural_164
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_havsali_cikinti

Kural_174
 parca_silindirik
 daire
 girdi_on_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_havsali_cikinti

Kural_187
 parca_prizmatik
 kapalı_katı
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sınırları_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 boydanboya_prizmatik_delik
 Kural_189

parca_prizmatik
 kapalı_katı
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sınırları_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_kor_delik

Kural_191
 parca_prizmatik
 kapalı_katı
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sınırları_icinde
 kesit_var
 izdusum_cift_kose

temel_kati_boyu_anakatidan_buyuk
boydanboya_prizmatik_delik

Kural_193
parca_prizmatik
kapalı_katı
girdi_yan_gorunuste
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunmez
izdusum_anaparca_sınırları_icinde
kesit_yok
izdusum_tek_kose
temel_kati_boyu_anakatidan_buyuk
prizmatik_havsali_boydanboya_delik

Kural_197
parca_prizmatik
kapalı_katı
girdi_yan_gorunuste
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sınırları_icinde
kesit_yok
izdusum_tek_kose
temel_kati_boyu_anakatidan_kucuk
prizmatik_havsali_kor_delik

Kural_201
parca_prizmatik
kapalı_katı
girdi_yan_gorunuste
girdi_cizgi_tipi_gorunmez
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sınırları_icinde
kesit_var
izdusum_cift_kose
temel_kati_boyu_anakatiyla_esit
prizmatik_delik

Kural_203
parca_prizmatik
kapalı_katı
girdi_yan_gorunuste
girdi_cizgi_tipi_gorunmez

izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sinirlari_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_delik

Kural_210
 parca_prizmatik
 kapali_kati
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_cikinti

Kural_371
 parca_prizmatik
 kapali_kati
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_cikinti

Kural_213
 parca_prizmatik
 kapali_kati
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_buyuk
 prizmatik_cikinti

Kural_221

parca_prizmatik
 kapalı_katı
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_cikinti

Kural_224
 parca_prizmatik
 kapalı_katı
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_cikinti

Kural_228
 parca_prizmatik
 kapalı_katı
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatidan_buyuk
 prizmatik_havsali_cikinti
 Kural_305
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sınırları_icide
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit

silindirik_boydanboya_delik

Kural_306

parca_silindirik

daire

girdi_yan_gorunuste

girdi_cizgi_tipi_gorunur

izdusum_cizgi_tipi_gorunmez

izdusum_anaparca_sinirlari_icinde

kesit_yok

izdusum_cift_kose

temel_kati_boyu_anakatidan_kucuk

silindirik_kor_delik

Kural_309

parca_silindirik

daire

girdi_yan_gorunuste

girdi_cizgi_tipi_gorunur

izdusum_cizgi_tipi_gorunur

izdusum_anaparca_sinirlari_icinde

kesit_var

izdusum_cift_kose

temel_kati_boyu_anakatidan_kucuk

silindirik_kor_delik

Kural_311

parca_silindirik

daire

girdi_yan_gorunuste

girdi_cizgi_tipi_gorunur

izdusum_cizgi_tipi_gorunmez

izdusum_anaparca_sinirlari_icinde

kesit_yok

izdusum_tek_kose

temel_kati_boyu_anakatiyla_esit

silindirik_boydanboya_havsali_delik

Kural_313

parca_silindirik

daire

girdi_yan_gorunuste

girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatidan_buyuk
 silindirik_boydanboya_havsali_delik

Kural_315
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sinirlari_icinde
 kesit_var
 izdusum_tek_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_kor_havsali_delik

Kural_329
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_cikinti

Kural_330
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_cikinti

Kural_332
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_cikinti

Kural_335
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_sagdan_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_havsali_cikinti

Kural_342
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_cikinti

Kural_344
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunmez

izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_cikinti

Kural_347
 parca_silindirik
 daire
 girdi_yan_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_soldan_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_havsali_cikinti

Kural_373
 parca_prizmatik
 kapali_kati
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 boydanboya_prizmatik_delik

Kural_375
 parca_prizmatik
 kapali_kati
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_kor_delik

Kural_377

parca_prizmatik
 kapalı_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sınırları_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_buyuk
 boydanboya_prizmatik_delik

Kural_379

parca_prizmatik
 kapalı_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sınırları_icinde
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_havsali_boydanboya_delik

Kural_382

parca_prizmatik
 kapalı_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sınırları_icinde
 kesit_var
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_havsali_boydanboya_delik

Kural_384

parca_prizmatik
 kapalı_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sınırları_icinde

kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_havsali_kor_delik

Kural_388
 parca_prizmatik
 kapali_kati
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sinirlari_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_delik

Kural_390
 parca_prizmatik
 kapali_kati
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sinirlari_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_delik

Kural_393
 parca_prizmatik
 kapali_kati
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_havsali_delik

Kural_400
 parca_prizmatik
 kapalı_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_ustten_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_cikinti

Kural_402
 parca_prizmatik
 kapalı_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_ustten_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_cikinti

Kural_404
 parca_prizmatik
 kapalı_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_ustten_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatidan_buyuk
 prizmatik_havsali_cikinti

Kural_468
 parca_prizmatik
 açık_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparcaya_sınırları_icin

kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 prizmatik_kor_delik

Kural_481
 parca_prizmatik
 açık_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_alttan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_cikinti

Kural_487
 parca_prizmatik
 açık_katı
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_ustten_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 prizmatik_havsali_cikinti

Kural_493
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sınırları_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_boydanboya_delik

Kural_494
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_kor_delik
 Kural_497
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparca_sinirlari_icinde
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatidan_kucuk
 silindirik_kor_delik

Kural_499
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_boydanboya_havsali_delik

Kural_500
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunmez
 izdusum_anaparca_sinirlari_icinde
 kesit_yok
 izdusum_tek_kose

temel_kati_boyu_anakatidan_kucuk
silindirik_kor_havsali_delik

Kural_503
parca_silindirik
daire
girdi_ust_gorunuste
girdi_cizgi_tipi_gorunur
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sinirlari_icinde
kesit_var
izdusum_tek_kose
temel_kati_boyu_anakatidan_kucuk
silindirik_kor_havsali_delik

Kural_505
parca_silindirik
daire
girdi_ust_gorunuste
girdi_cizgi_tipi_gorunmez
izdusum_cizgi_tipi_gorunmez
izdusum_anaparca_sinirlari_icinde
kesit_yok
izdusum_cift_kose
temel_kati_boyu_anakatiyla_esit
silindirik_boydanboya_delik

Kural_510
parca_silindirik
daire
girdi_ust_gorunuste
girdi_cizgi_tipi_gorunmez
izdusum_cizgi_tipi_gorunur
izdusum_anaparca_sinirlari_icinde
kesit_var
izdusum_cift_kose
temel_kati_boyu_anakatidan_buyuk
silindirik_boydanboya_delik

Kural_520
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_ustten_cikinti
 kesit_var
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_cikinti

Kural_523
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunur
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_ustten_cikinti
 kesit_yok
 izdusum_tek_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_havsali_cikinti

Kural_529
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_alttan_cikinti
 kesit_yok
 izdusum_cift_kose
 temel_kati_boyu_anakatiyla_esit
 silindirik_cikinti

Kural_536
 parca_silindirik
 daire
 girdi_ust_gorunuste
 girdi_cizgi_tipi_gorunmez
 izdusum_cizgi_tipi_gorunur
 izdusum_anaparcaya_alttan_cikinti

kesit_yok
izdusum_tek_kose
temel_kati_boyu_anakatidan_kucuk
silindirik_havsali_cikinti

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ŞAHİN, İsmail
 Uyruğu : T.C.
 Doğum tarihi ve yeri : 07.05.1971 Trabzon
 Medeni hali : Evli
 Telefon : 0 (505) 256 13 43
 Faks : 0 (312) 212 00 59
 e-mail : isahin@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi / Makine Eğitimi Bölümü	2001
Lisans	Gazi Üniversitesi/ Makine Resim ve Konst. Eğt. Anabilim Dalı	1997
Lise	Kırıkkale Endüstri Meslek Lisesi	1988

İş Deneyimi

Yıl	Yer	Görev
1998-1999	Milli Eğitim Bakanlığı	Öğretmen
1999-	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar

1. **Şahin, İ.**, Börklü, H.R., 2007. Hacimsel Kesiştirme Yaklaşımı Kullanarak Silindirik Parçaların Katı Modellerinin Oluşturulması, Politeknik Dergisi, 10 (1): 71-77.
2. **Şahin, İ.**, Börklü, H.R., 2007. Reconstruction of Solid Models of Prismatic Parts from their Two Views With Use Of Expert System, Journal of The Faculty of Engineering and Architecture of Gazi University, 22 (4): 819-827.
3. **Şahin, İ.**, Okay, Ş., Özdemir, S., 2007. “Türkiye’de Mesleki Teknik Eğitimin Durumu ve Karşılaşılan Sorunlar”, Ulusal Teknik Eğitim, Mühendislik ve Eğitim Bilimleri Genç Araştırmacılar Sempozyumu (UMES), Kocaeli Üniv, Kocaeli.
4. Başak, H., **Şahin, İ.**, Gülen, M., 2007. İnsansız Hava Aracı Kazalarının Önlenmesi İçin Uzman Sisteme Dayalı Risk Yönetim Modeli, Teknoloji Dergisi. (Yayınlanmak üzere kabul edildi)
5. **Şahin, İ.**, Börklü, H.R., 2007. 2B İzdüşümlerden Otomatik Katı Modeller Oluşturmada Uzman Bir Yaklaşım, Mühendislik Bilimleri Dergisi. (Hakem incelemesinde).
6. **Şahin, İ.**, Börklü, H.R., 2005. Automatic Reconstruction of 3D Objects In AutoCAD Environment From Their Projections, 1st International Vocational and technical Education Technologies Congress, İstanbul, Turkey.
7. **Şahin İ.**, Börklü, H.R., 2003. Üç Temel Görünüşü Verilen Nesnelerin AutoCAD 2000 Ortamında Otomatik 3B Modellerinin Oluşturulması, Makine Tasarım ve İmalat Dergisi, 5 (1): 9-22.

8. **Şahin, İ.**, Börklü H. R., 2002 . “İzdüşümlerden hacme Dayalı 3B’lu Nesneler Elde Etmede Dahili Elemanların Kullanımı”, G.Ü. Fen Bilimleri Dergisi, 15, 2, 351-361,.
9. **Şahin, İ.**, Börklü H. R., 2002 . “Teknik Resimleri Verilen Parçaların Bilgisayar Destekli 3B Modellerinin Elde Edilmesi: Hacimsel Bir Yaklaşım”, Mühendis ve Makina , 510, 37-45.
10. **Şahin İ.**, Börklü, H.R., 2002. 2B Görünüş Verilerinden Bilgisayar Destekli 3B Modeller Oluşturulması: Kaynak Araştırması, Mühendis ve Makine , 507: 20-31, 2002.
11. **Şahin, İ.**, Börklü, H.R., 2001. Temel İzdüşüm Görünüşlerinden Bilgisayar Destekli Üç Boyutlu Nesnelerin Elde Edilmesi, Politeknik Dergisi, 4 (3): 1-10.
12. **Şahin, İ.**, Börklü, H.R., 2001. 2B İzdüşüm Görünüşlerinden Otomatik 3B Sınır Temsili Modeller oluşturma, G.Ü Fen Bilimleri Enstitüsü Dergisi, 14, 4, 1455-1465.
13. **Şahin, İ.**, Börklü, H.R., 2001. İzdüşümlerden 3B’lu Geometrik Nesneler Elde Edilmesi, 20. Tarımsal Mekanizasyon Kongresi, Harran Üniversitesi, Şanlıurfa.

Projeler

Takım Tezgahlarında Talaş Kaldırma İlkelerinin Web Tabanlı Eğitimi, G.Ü B.A.P, Proje Kod: O7/2004-34. (Araştırmacı).

Yapay Zeka Teknikleri Kullanarak İki Boyutlu İzdüşümlerden Katı Model oluşturma, G.Ü. B.A.P., Proje Kod: O7/2004-27. (Araştırmacı)

Editörlük

Başak, H., VBA ve AutoCAD Uygulamaları, **Ed. İsmail Şahin**, Anahtar Kitapları, 2005, Ankara

Hobiler

Kitap, Makine teknolojileri, Bilgisayar teknolojileri, Futbol, Web tasarımı.