

KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNA ANABİLİM DALI
DOKTORA TEZİ

127396

2B'LU MODELLERDEN 3B'LU MODEL ÇIKARIMI SIRASINDA UNSUR
TANIMAYA VE İŞLEM PLANLAMAYA YÖNELİK BİR TEKNİK
GELİŞTİRİLMESİ

TC. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

127386

İBRAHİM ÇAYIROĞLU

OCAK-2002

Fen Bilimleri Enstitü Müdürünün Onayı.


Doç. Dr. M. Yakup ARICA
Fen Bilimleri Enstitüsü
MÜDÜRÜ
Müdür

Bu tezin Doktora tezi olarak Makine Anabilim Dalı standartlarına uygun olduğunu onaylarım.


Prof. Dr. Veli ÇELİK
Anabilim Dalı Başkanı


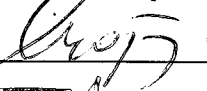


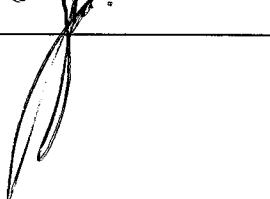
Bu tezi okuduğumuzu ve Doktora tezi olarak bütün gerekliliklerini yerine getirdiğini onaylarız.


Prof. Dr. Can ÇOĞUN
Ortak Danışman


Prof. Dr. Veli ÇELİK
Danışman

Jüri Üyeleri

Prof. Dr. Veli ÇELİK
Prof. Dr. Can ÇOĞUN
Doç. Dr. Ersan ASLAN
Doç. Dr. Ulvi ŞEKER
Yrd. Doç. Dr. M. Hüsnü DİRİKOLU

ÖZET

2B'LU MODELLERDEN 3B'LU MODEL ÇIKARIMI SIRASINDA UNSUR TANIMAYA VE İŞLEM PLANLAMAYA YÖNELİK BİR TEKNİK GELİŞTİRİLMESİ

ÇAYIROĞLU, İbrahim

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Makine Anabilim Dalı, Doktora Tezi

Danışman : Prof. Dr. Veli ÇELİK

Ocak 2002, 128 sayfa

Bu tez çalışmasında makine parçalarının 2B'lu teknik çizimlerinden, 3B'lu modelini oluşturmak ve üzerindeki işleme unsurlarını (negatif unsurlar) tespit etmek için yeni bir metot geliştirilmiştir. Algoritma programlanarak geçerliliği ispatlanmıştır. Metodun uygulandığı parçalar dik işleme merkezlerinde işlenebilen prizmatik parçalarla sınırlı tutulmuştur. Parçanın üç temel görünüşü (ön, üst, sol yan) programa girdi olarak verilmekte ve çıktı olarak ise parçanın 3B'lu modeli ile üzerindeki unsurların şekli ve özellikleri alınmaktadır.

Program, başlangıçta parçayı prizma şeklinde ham bir malzeme olarak almakta ve üzerindeki negatif unsurları yeni bir yöntem olan "İzdüşüm ve 3-alan metoduyla" oluşturmaktadır. Ardından unsurları ham malzemedен çıkararak geriye

kalan hacmi cismin 3B'lu modeli olarak göstermektedir. Karmaşık unsurları tespit edip, bilinen unsurlar cinsinden ifade edebilmektedir. Unsurların tipini, kör veya tam delik durumunu, kesicinin hangi kenarlardan girip giremeyeceğini, unsurların işlem sırasını tespit etmektedir. Cismi oluşturduktan sonra geri kontrolü yapabilmektedir.

Ayrıca 2B'lu çizimleri program içerisinde gerçekleştirmek için, genel bir 2B'lu çizim programı yazılmış ve programın başına eklenmiştir.

Anahtar Kelimeler: 2B'lu Çizim, 3B'lu Model, Unsur Tanıma, İşlem Planlama.



ABSTRACT

A STUDY OF A TECHNIC DEVELOPING FOR FEATURE RECOGNITION AND CAPP AS 3D MODEL EXTRACTION FROM 2D MODELS

ÇAYIROĞLU, İbrahim

Kırıkkale University

Institute of Science and Technology

Department of Mechanical Eng., Ph.D. Thesis

Supervisor: Prof. Dr. Veli ÇELİK

January 2002, 128 pages

In this study, a new method has been developed for extracting 3D models corresponding to 2D technical drawings and also for determining machining features (negative features) on parts. The Algorithm's validity is proved by programming. The parts on which the method was applied are limited to prismatic parts that are machineable on CNC vertical centers. As input to the program, the basic 3 views of parts (front, top, left side) are used. The followings are taken as output from the program; the 3D shape of part, the features on the part and the shape and characteristics of these features.

At the beginning; the program accepts the prismatic shaped part as raw material. Then, it forms the negative features by a new method called "Projection and 3-Area method". Later, it subtract the features from the raw material and shows the

remaining volume as a 3D model. It is able to define complex features in terms of simple standard features. It can identify feature's type, blind or through hole cases, direction of cutter feed and the manufacturing sequence. With an incorporated feedback algorithm, the programme can test the validity of the formed 3D model.

Also a 2D drawing program is written and then attached to the beginning of the main program to perform simple 2D drawings within the program.

Key Words: 2D Drawing, 3D Model, Feature Recognition, CAPP



TEŞEKKÜR

Yapmış olduğum bu çalışmamda bilgi, tecrübe, yönlendirme ve katkılarıyla yardımlarını esirgemeyen Danışman Hocalarım Prof. Dr. Veli ÇELİK ve Prof. Dr. Can ÇOĞUN Bey'e, bilgilerinden, kitaplarından ve kaynak eserlerinden yararlandığım Doç. Dr. Ersan ASLAN Bey'e ve Doç. Dr. Ulvi ŞEKER Bey'e, sağlamış oldukları bilgisayar donanımı ve techizat desteğiyle, dekanlığa, bütün öğretim üyesi hocalarıma ve araştırma görevlisi arkadaşlarıma, manevi desteğini esirgemeyen aileme, ayrı ayrı teşekkür eder saygılarımı sunarım.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ	xiv
1. GİRİŞ	1
1.1. Literatür İnceleme	6
1.1.1. 2B'lu Çizimlerden 3B'lu Modellerin Oluşturulması	6
1.1.2. 2B'lu Çizimlerden Unsur Çıkarımı	15
1.1.3. 3B'lu Modellerden Unsur Çıkarımı	21
1.1.4. Çalışmanın Amacı	23
2. MATERYAL VE YÖNTEM	26
2.1. 2B'lu Çizim Programı	26
2.1.1. Programın Yapısı	26
2.1.2. Programın Algoritması	28
2.1.3. Kullanılan Komutlar	31
2.1.3.1. Çizim Komutları (Draw)	31
2.1.3.1.1. Doğru Komutu (Line)	31
2.1.3.1.2. Daire Komutu (Circle)	32
2.1.3.1.3. Dikdörtgen Komutu (Rencangle)	32

2.1.3.1.4. Yay Komutu (Arc)	33
2.1.3.1.5. Çokgen Komutu (Polygon)	35
2.1.3.2. Düzeltme Komutları (Edit)	36
2.1.3.2.1. Taşıma Komutu (Move)	36
2.1.3.2.2. Kopyalama Komutu (Copy)	36
2.1.3.2.3. Döndürme Komutu (Rotate)	36
2.1.3.2.4. Aynalama Komutu (Mirror)	37
2.1.3.2.5. Silme Komutu (Erase)	40
2.1.3.3. Görüntü Komutları (View)	42
2.1.3.3.1. Büyütme Komutu (Zoom Positive)	42
2.1.3.3.2. Küçültme Komutu (Zoom Negative)	43
2.1.3.3.3. Hepsini Göster Komutu (Zoom All)	44
2.1.3.3.4. Pencereyi Göster Komutu (Zoom Window)	45
2.1.3.3.5. Önceki Ekran Komutu (Zoom Previous)	46
2.1.3.3.6. Ekran Kaydırma Komutu (Pan)	46
2.1.3.3.7. Yeniden Çiz Komutu (Redraw)	47
2.2. Çizimin Algoritmaya Hazırlanması	48
2.2.1. Temel Kavramlar	48
2.2.2. Çizgilerin Parçalara Ayrılması	51
2.2.2.1. Doğru-Doğru Kesişimi	52
2.2.2.1.1. Doğru1 ve Doğru2 Çapraz	54
2.2.2.1.2. Doğru1 Çapraz, Doğru2 Yatay	55
2.2.2.1.3. Doğru1 Çapraz Doğru2 Dikey	56
2.2.2.1.4. Doğru1 Dikey, Doğru2 Yatay	57
2.2.2.1.5. Doğru1 ve Doğru2'nin Paralel Olma Durumu	57

2.2.2.2. Doğru-Daire Kesişimi	58
2.2.2.2.1. Doğru ile Dairenin İki Noktadan Kesişme Durumu ($\Delta>0$)	59
2.2.2.2.2. Doğru ile Dairenin Bir Noktadan Kesişme Durumu ($\Delta=0$)	60
2.2.2.3. Doğru-Yay Kesişimi	61
2.2.2.4. Daire-Daire Kesişimi	62
2.2.2.5. Daire-Yay Kesişimi	64
2.2.2.6. Yay-Yay Kesişimi	64
2.2.3. Görünüşlerin ve Sınır Çerçevenin Tespiti	65
2.2.3.1. Görünüşlerin Tespiti	65
2.2.3.2. Sınır Çerçevenin Tespiti	67
2.2.4. 2B'lu Çizim Elemanlarının Tespiti	68
2.2.4.1. Düğüm-Koordinat Dosyasının Oluşturulması	69
2.2.4.2. Kenar-Düğüm Dosyasının Oluşturulması	70
2.2.4.3. Düğüm-Kenar Dosyasının Oluşturulması	72
2.2.4.4. Köşe-Kenar-Koordinat Dosyasının Oluşturulması	72
2.2.4.5. Alanların Tespit Edilmesi	73
2.2.4.5.1. Alanları Tespit İçin Uygulanan Kriterler	75

3. ARAŞTIRMA BULGULARI	81
3.1. 3B'lu Modelin Oluşturulması ve Unsurların Tespiti	81
3.1.1. Temel Görünüşün Tespiti	81
3.1.1.1. Görünüşün Simetrikliğinin Bulunması	85
3.1.2. İzdüşüm Yöntemi	86
3.1.2.1. İzdüşüm Çizgilerinin Oluşturulması	86
3.1.2.2. Ortak Çözümlerin Bulunması	88
3.1.3. Taban ve Tavan Yüzeylerinin Oluşturulması	93
3.1.3.1. Algoritmada Kullanılan Renkler ve Anlamları	93
3.1.3.2. Taban Yüzeylerin Tespiti	94
3.1.3.3. Unsurların İşlem Sırasının Tespiti	95
3.1.3.4. Tavan Yüzeylerin Oluşturulması	96
3.1.3.5. Süpürme Taban ve Unsurların Oluşturulması	98
3.1.4. Unsur Tipinin Bulunması	100
3.1.5. Karmaşık Unsurların Tespiti	115
3.1.6. 3B'lu Cismin Oluşturulması	119
3.1.6.1. Sınır Kutunun Oluşturulması	119
3.1.6.2. 3B'lu Cismin Oluşturulması	120
3.1.7. Cismin Geri Kontrolü ve Eksik Unsurların Bulunması	121
4. TARTIŞMA VE SONUÇ	124
KAYNAKLAR	129
ÖZGEÇMİŞ	137
EK 1. GENEL UYGULAMA PARÇASI	138
EK 2. YATAKLAMA PARÇASI	151

ŞEKİLLER DİZİNİ

ŞEKİL

1.1. DSG yöntemiyle, ham malzemedен çıkarılan negatif unsurların gösterimi ..	5
2.1. 2B'lu çizim için hazırlanan programın ekran görüntüsü	26
2.2. 2B'lu çizim programının algoritması	30
2.1. Doğru komutunun uygulanışı	31
2.4. Daire komutunun uygulanışı	32
2.5. Üç noktadan geçen yay komutunun uygulanışı	33
2.6. Merkezi, başlangıç ve son açısı bilinen yay komutunun uygulanışı	35
2.7. Aynalama komutunun uygulama aşamaları	39
2.8. Bir doğrunun fare ile seçilmesi işlemi	40
2.9. Bıyıtme komutunun uygulanışı	42
2.10. Hepsini Göster komutunun uygulanışı	44
2.11. Pencereyi Göster komutunun uygulanışı	45
2.12. Ekran Kaydırma komutunun uygulanışı	47
2.13. Algoritmada kullanıldığı şekliyle köşe ve düğüm kavramlarının gösterimi	49
2.14. Algoritmada kullanıldığı şekliyle iç ve dış döngülerin gösterimi	50
2.15. Çizgilerin parçalara ayrıldığı noktaların gösterimi	52
2.16. Çizgileri parçalama algoritmasının genel yapısı	53
2.17. İki doğrunun kesişimi	54
2.18. Doğru1'in çapraz, doğru2'nin yatay bulunduğu konumda kesişimi	55
2.19. Doğru1'in çapraz, doğru2'nin dikey bulunduğu konumda kesişimi	56
2.20. Doğru1'in dikey, doğru2'nin yatay bulunduğu konumda kesişimi	57

2.21. Doğru ₁ ve doğru ₂ 'nin paralel olma durumu	57
2.22. Kesişim noktasının, yayın uçları arasında olup olmadığının tespit edilmesi	61
2.23. İki dairenin kesişme durumu	62
2.24. Daire ve yayların kesişme durumu	64
2.25. Görünüşleri ve sınır çerçeveyi tespit eden algoritmanın yapısı	66
2.26. 2B'lu çizimin görüntülere taksimi	67
2.27. Sınır çerçevenin tespiti ve çizgilerinin görüntülere eklenmesi	68
2.28. Düğüm tipleri ve gösterimi	69
2.29. Düğüm çeşitlerinin gösterimi	70
2.30. Algoritmada kullanılan köşe çeşitlerinin gösterimi	74
2.31. Algoritmada kullanılan çeşitli alan tipleri	75
2.32. Alanların tespitinde kullanılan algoritmanın yapısı	76
2.33. Sol taraftan en küçük açılı kenar yakalama metodu	77
2.34. İç alan, Dış alan ve En Dış Çerçevenin tespiti	79
3.1. Temel görünüşün tespiti için en uygun görünüşün bulunması	82
3.2. Doğru ve yayda simetriklik durumları	85
3.3. Yüzeyin eksenler üzerindeki izdüşümlerinin oluşturulması	87
3.4. Teğet yüzeylerin teğet çizgiden ikiye bölünmesi	89
3.5. İzdüşümlerden ortak çözümlerin bulunması	90
3.6. İç ve dış alanın ortak izdüşüm çözümleri	91
3.7. İzdüşüm yöntemiyle ilgili olarak geliştirilen algoritmanın yapısı	92
3.8. Taban çeşitlerinin gösterimi	96
3.9. Birinci yöntemle yeni tavanın oluşturulması	97
3.10. Düz altı boş tabanın eski tavandan çıkarılmasıyla oluşturulan yeni tavanlar	98

3.11. Taban düzlemsel iken süpürme tabanının ve unsurun oluşturulması	99
3.12. Taban kavisli eğik durumda iken süpürme taban ve unsurun oluşturulması	100
3.13. Taban, tavan, süpürme taban ve unsuru oluşturan algoritmanın yapısı	101
3.14. Kademe unsurunun gösterimi	102
3.15. Düz Kanal unsurunun gösterimi	103
3.16. Kapalı Kademe unsurunun gösterimi	103
3.17. Kapalı Kanal unsurunun gösterimi	104
3.18. Cep unsurunun gösterimi	104
3.19. Kör Delik unsurunun gösterimi	105
3.20. Pah (Açılı yüzey) unsurunun gösterimi	106
3.21. Dış Bükey Kavis unsurunun gösterimi	106
3.22. İç Bükey Kavis unsurunun gösterimi	107
3.23. Negatif-Ada unsurunun gösterimi	108
3.24. İç Bükey Yarım-Silindir unsurunun gösterimi	108
3.25. Kırılmaç-Kuyruğu-Kanal unsurunun gösterimi	109
3.26. V-Kanal unsurunun gösterimi	110
3.27. T-Kanal unsurunun gösterimi	110
3.28. Karmaşık bir süpürme tabanının gösterimi	115
3.29. Süpürme yayların oluşturduğu unsurların çıkarımı	116
3.30. Kırmızı ve mavi kenarların uzatılması ve alanların bölünmesi	117
3.31. Birleştirme ve çıkarma yoluyla yeni unsurların elde edilmesi	118
3.32. Unsur tipi ve işlem sırası belirlenmiş süpürme tabanlar	118
3.33. Sınır kutunun elde edilmesi	119
3.34. 3B'lu cismin oluşturulması	120
3.35. Orijinal çizimden elde edilen 3B'lu cisim	122

3.36. Elde edilen 3B'lu cisimden tekrar kontrol görünüşlerinin çıkarımı	123
3.37. Temel görünüşün sol görünüş alınmasıyla bulunan eksik unsur	123
Ek 1.1. Örnek parçanın üç görünüşü ve çizgilerin parçalara ayrıldığı noktalar ..	138
Ek 1.2. C-yüzeyinin diğer görünüşlerdeki projeksiyon karşılığının bulunması ..	139
Ek 1.3. Temel görünüşteki alanların, diğer görünüşlerdeki ortak çözümleri	141
Ek 1.4. Tabanların gösterimi	141
Ek 1.5. A-tabanına ait tavan, S-taban ve oluşturulan Y-tavanın gösterimi	142
Ek 1.6. G-tabana ait tavan, S-taban ve oluşturulan Y-tavanların gösterimi	143
Ek 1.7. F-tabana ait tavan, S-taban, unsur ve oluşturulan yeni tavanlar	144
Ek 1.8. B-tabanına ait tavan, S-taban ve unsurun gösterimi	145
Ek 1.9. C-tabanına ait tavan, S-taban, unsur ve oluşturulan tavanların gösterimi	146
Ek 1.10. D-tabanına ait tavan, S-taban ve unsurun gösterimi	147
Ek 1.11. E-tabanına ait tavan, S-taban, ve unsurun gösterimi	147
Ek 1.12. Taban kalmadığında, mevcut tavadan oluşturulan unsurlar	148
Ek 1.13. Sınır kutudan unsurların çıkarılması ve 3B'lu cismin oluşturulması ...	149
Ek 2.1. Örnek parçanın üç görünüşü ve alanların ortak çözümleri	151
Ek 2.2. Örnek modeldeki taban yüzeylerin gösterimi	152
Ek 2.3. Taban-A, İlk Tavan ve oluşan Yeni Tavanlar	153
Ek 2.4. Algoritmanın Taban-B'üzerinde uygulanan aşamaları	153
Ek 2.5. Algoritmanın Taban-C için uygulanan aşamaları	155
Ek 2.6. D,E,F tabanlarından oluşturulan unsurların gösterimi	156
Ek 2.7. Geriye kalan tavanlardan oluşturulan unsurların gösterimi	157

ÇİZELGELER DİZİNİ

ÇİZELGE

1.1. 2B'tan 3B'ta dönüřüm yapan bazı algoritmaların kısa açıklaması	14
1.2. 2B'lu resimlerden unsurları çıkaran algoritmalar arasındaki farklar	20
3.1. Temel görünüşün tespitindeki kriterlerin örnek parçaya uygulanması	84
3.2. Unsur kütüphanesinde bulunan unsurların özellikleri	111



1. GİRİŞ

Bir makine parçasının gösteriminin en yaygın biçimi 2B'lu teknik resimlerdir. Bu seviyedeki bir tanımlama bir parçanın şeklini anlatmak için ne yeterince bir bilgi sağlayabilir ne de bu yöndeki Bilgisayar Destekli İşlem Planlaması (BDİP)/Bilgisayar Destekli İmalat (BDİ) gibi tasarım sonrası aktiviteleri destekleyebilir. Tasarımcılar, tasarımın içerdiği bilgileri elde etmek için bu resimleri yorumlamak zorundadırlar.

Mühendislik çizimleri şu anda tasarım, üretim ve saklama için en önemli teknolojik dökümanlardır. Aynı zamanda farklı ülkelerdeki teknik personel arasında, teknoloji ve bilgi aktarımını sağlayan uluslararası dillerin bir türüdür. Bugün endüstride ve bürolarda milyonlarca çizim daha önceden çizilmiş olup saklanmaktadır. Çoğu yeni tasarım, bu eski tasarımların değiştirilmiş/geliştirilmiş şeklidir. Bu yüzden dik izdüşüm görünüşlerden cisim oluşturan ve unsurları çıkaran bir algoritma sınırlı bile olsa Bilgisayar Destekli Tasarım ve İmalat (BDT)/(BDİ) sistemlerinde önemli uygulamalara sahip olabilir. Bu uygulamalar;

1. Katı modeller eski çizim kağıtlarındaki görünüşlerden doğrudan elde edilebilir. Kağıt üzerindeki çizimlerin bilgisayar ortamına aktarılması ise iki şekilde yapılabilir:
 - a) Resimler tarayıcıda taranıp, elde edilen bilgiler resim işleme (image processing) metotları ile işlenip, resim üzerindeki elemanlar sayısal değerlere dönüştürülür. 2B'tan 3B'ta dönüşüm programı da bu değerleri alıp parçanın katı modelini oluşturur. Burada hassas bir taramaya ve özel

- bir resim işleme programına ihtiyaç vardır. Bu yöntemle elde edilecek koordinatlar tam değerde olamayacağı için belli bir tolerans aralığı kullanılmalıdır.
- b) Teknik resimler doğrudan bilgisayar ortamında çizilir ve 2B'tan 3B'a dönüşüm programına aktarılır. Bunun bir zaman kaybına yol açması muhtemel olmakla beraber, çizimin hassasiyeti açısından zorunlu olmaktadır. Zaten günümüzde artık 2B'lu çizimler bilgisayar ortamında çizildiği için bunlar direk olarak 2B'tan 3B'a dönüşüm programına aktarılabilir.
2. 2B'lu çizimlerden unsurlar da doğrudan çıkarılabilir. 2B'lu çizimden unsur çıkarımı, 3B'lu çizimden çıkarmaktan daha farklı bir konudur.
 3. Katı modelin verileri daha basit BDT sistemlerinden, veri dönüşüm standartlarına (IGES, DXF vb.) aktarılıp, daha karmaşık 3B'lu çizimler yapabilen programlar yerine daha basit 2B'lu çizim yapabilen programlar kullanılabilir.
 4. Katıların BDT sistemlerinde oluşturulması için yeni ve etkileşimli bir araç yapılmış olur. Genellikle pek çok kenar ve yüzeye sahip 3B'lu şekilleri oluşturmak ekran kullanıcıları için pratik değildir. Otomatik olarak katı cismi 2B'lu projeksiyonlardan oluşturmak bu problemin bir çözümüdür.
 5. 2B'lu çizimlerden 3B'lu modeli oluşturan bir algoritma matematikle olan ilgisinin yanında çok çeşitli pratik uygulamalara sahiptir. Bu tip modeller, parçanın simülasyonunda, tasarımın doğruluğunun ispatlanmasında, analizinde, üretiminde ve döküman işlemlerinde kullanılabilir.

Her ne kadar 2B'tan 3B'ta dönüşüm işlemi, makine çizimlerinin kullanımında bazı avantajlara sahip olsa da, çoğu durumlarda pratik değildir. Çünkü parçanın yeterli sayıda görünüşünü gerektirmektedir. Çoğu çizimlerde kolay okumak ve sadelik için bazı kısa notlar ve kısaltmalar kullanılmaktadır. Bu bilgiler tasarımcının niyetini izah eder. Dönüşümlerde bu tür bilgileri algılama zorluğu vardır.

BDT tasarım sistemleri endüstride yaygın bir şekilde kullanılmaktadır, fakat parçayı tanımlayan bilgilerin seviyesi çok düşüktür. Bir BDT sisteminde parçayı tanımlayan bilgiler ister 2B'lu, ister 3B'lu olsun düşük seviyededir. Örneğin; tel çerçeve geometri, B-rep geometri ve topoloji, CSG operatörler gibi bilgilere haiz olduğu halde bunlardan hiç biri işlem planlama gibi bir uygulama sahasında doğrudan kullanılamaz. Bu yüzden yüksek seviyeli unsur bilgileri eksiktir.

Klasik 2B'lu mühendislik çizimlerinden unsur tabanlı 3B'lu parçanın oluşturulması daha ileri işlemler gerektirmektedir. Unsur tabanlı BDT, tasarım gösteriminin akıllı bir şeklidir. Burada tasarım daha sonraki uygulamalarla doğrudan ilgili olacak şekilde yüksek düzeyde tanımlamalardan oluşur.

Geleneksel BDT tasarımı, 2B'lu elemanlar şeklinde (çizgi, yay, daire) ya da 3B'lu elemanlar (tel çerçeve, yüzey, katı) şeklinde gösterilir. Fakat bu gösterimlerin tasarımı, imalat terimleri şeklinde yorumlayan bir işlem planlayıcı için çok az önemi vardır. Unsur bilgisi imalatçıya parçayı işlemek için imalat işlemlerini kesici uçları vs. belirlemesini sağlar. Bu unsur çıkarımı için ilk sebeptir⁽¹⁾.

Tasarımcı için unsurlar, işin fonksiyonu ile ilgilidir. Oysa unsurlar genellikle işlem planlamasında, uygun üretim operasyonlarını tanıma için geliştirilmektedir. Bu ise unsurlara bakış açısını tasarım ve imalat yönünden oldukça farklı kılmaktadır⁽²⁾.

Unsur tanıma, uzun vadede BDT/BDİ bütünleşmesi için çok önemlidir. Her

ne kadar ticari unsur tanıma sistemleri bulunsa da, bu yazılımlar henüz başlangıç aşamasındadır. Şu anda tasarımların çok büyük bir kısmı elle yada bilgisayarla çizilmiş 2B'lu çizimler şeklinde sunulmaktadır. Bir çok tasarım da 3B'lu tel çerçeve modeli şeklinde veya katı model şeklinde bulunmaktadır. Bu yüzden orta ve kısa vadede gerekli teknoloji, geometrik modelin bu formlarının her birinden unsur tanıma üzerine olacaktır.

Ne yazık ki, geniş bir şekilde kabul gören 2B'lu çizimler ya da 3B'lu tel çerçeve şeklinde tanımlanan parçalar, üretim unsurlarını otomatik olarak belirleme bakımından yetersizdir. Bugün hala çoğu tasarım 2B'lu çizim şeklinde bulunmaktadır. Geçmişin birikimi olan bu geometrik resimler de bugünün ihtiyaçlarına uygun olarak çözülmelidir.

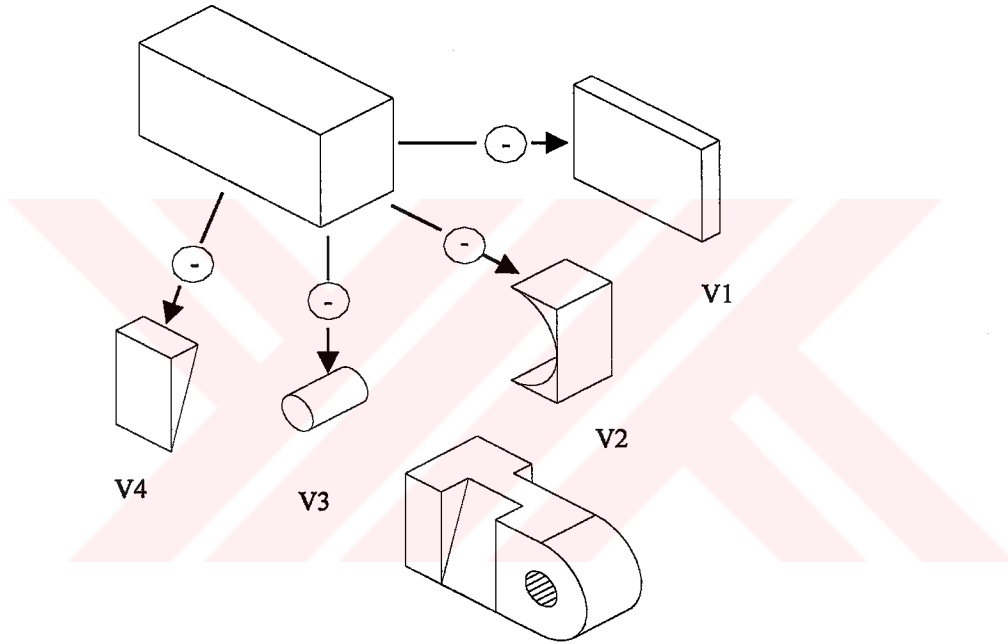
Unsur tanıma ile ilgili çoğu yayın⁽⁴⁵⁻¹⁴⁷⁾ bu probleme katı model gösterimi kullanarak yaklaşmaktadır.

Bir unsur, önceden nasıl tanımlanırsa tanımlansın ne kadar karmaşık olursa olsun tamamen bağımsız bir işleme unsuru olarak ele alınmış ve unsurlar ham malzemedan çıkarılan negatif hacimler olarak düşünülmüştür. Örneğin, Şekil 1.1'de bitmiş parça ham malzemedan V1, V2, V3 ve V4 hacimleri çıkarılarak işlenebilir. Tamamen çıkarmaya dayalı bu gösterim şekli “Çıkarımsal Katı Geometri” (Destructed Solid Geometry=DSG) ağacı olarak adlandırılır. Bu gösterim “Yapısal Katı Geometri” (Constructed Solid Geometri=CSG)'nin özel bir durumudur. Farklı olarak burada sadece “fark operatörü” kullanılmasıdır.

Burada yapılan çalışmada, cisim önce ham bir malzeme olarak kabul edilmekte ve üzerindeki işlenip çıkarılacak kısımlar, negatif unsurlar olarak tespit edilmekte ve ham malzemedan çıkarılmaktadır. Geriye kalan hacim cismin kendisi

olarak gösterilmektedir. Çıkarılan unsurların hangi tip unsur olduğu, kesicinin unsuru işlemek üzere hangi yüzeylerden girebileceği ve delik unsurların kapalı veya açık olup olmadığı tespit edilmektedir.

Geliştirilen algoritma Visual Basic dilinde programlanıp geçerliliği ispatlanmıştır. 2B'lu çizimi program içerisinde gerçekleştirebilmek için programın ön kısmına bir 2B'lu çizim modülü eklenmiştir.



Şekil 1.1. DSG yöntemiyle, ham malzemeden çıkarılan negatif unsurların gösterimi.

1.1. Literatür İnceleme

Literatür incelemesi daha çok 2B'lu çizimlerden 3B'lu modellerin elde edilmesi ve unsurların çıkarılması üzerine yapılmıştır. Bunun yanında konuyla ilgili olan 3B'lu modellerden unsur çıkarımı ve bilgisayar grafik sistemleri ile ilgili yayınlar da araştırılmıştır.

1.1.1. 2B' Çizimlerden 3B'lu Modellerin Elde Edilmesi

Prizmatik parçaların, parçanın dik görünüşlerinden yeniden oluşturulması 70'li yıllardan bu tarafa çalışılmaktadır. İlk çalışmalar parçanın özelliklerinin belirlenmesi için daha çok kullanıcı müdahalesini gerektirmekteydi. Konunun tarihsel gelişimi kullanıcının mümkün olduğunca bir çok sınırlamalardan kurtulmasıyla olmuştur. Ancak sınırlamalardan kurtulması, verilen problemin bir çok çözümünün olma ihtimalini doğurmuştur. Daha önce yapılan bu çalışmaların şekilsel geometrik tanımlamalara ve cebirsel topolojiye dayanmadığı görülmüştür. Daha gelişmiş çalışmalar bilgisayar grafik sistemler ve görünüş analizleri üzerine çalışanlar tarafından yapılmıştır.

Idesawa tarafından dik görünüşlerden bir modelin oluşturulması yaklaşımı yapılmıştır⁽⁵⁾. 2B'lu projeksiyonlardan bir tel çerçeve konstrüksiyonunu oluşturmuştur. Metot eş düzlemler (coplanar) kenarları bulmaya çalışır ve onları katı cismin şeklini oluşturmak için birbirine bağlar. Şekillerin açık ve net anlaşılır olmadığı durumlarda metot geçersizdir. Bu metodun ana adımları şöyledir.

- a) 2B'lu düğümlerden 3B'lu düğümlerin oluşturulması.
- b) 3B'lu düğümlerden 3B'lu kenarların oluşturulması.

- c) 3B'lu kenarlardan 3B'lu yüzeylerin oluşturulması.
- d) Bu yüzeylerin 3B'lu cisme montajı.

Lafue de aynı zamanda 2B'lu projeksiyonlardan katıyı oluşturmak için bir algoritma geliştirmiştir⁽⁶⁾. Bu algorithmada kenarlar yerine daha çok yüzeylerin tanımlaması gerekmektedir. Sadece 2 görünüş olduğunda değil, 3 görünüş bulunduğu durumlarda da bazı belirsizlikler çıkabilmektedir. Hazırlanan program kullanıcı desteklidir /etkileşimlidir.

Markowsky ve Wesley, tel çerçeve görünüşü şeklinde verilen (köşe ve kenarlardan oluşan) çok yüzlü cisimlerin mümkün olabilecek bütün çözümlerini araştıran ve sonuçta cismi hacimsel olarak ortaya koyan bir algoritma sunmuşlardır⁽⁷⁾. Bu algoritma makine tasarımında hayli uygulanmıştır. Bu algoritma da düz kenarlara ve düzlemsel yüzeylere sahip cisimlerle sınırlıdır.

Preiss'in metodu, her bir çizginin sayısallaştırıcıdan (digitezer) koordinatlarını okur ve ardından çizgilerin kesişmesinden mümkün olabilecek bütün düğümleri oluşturur⁽⁸⁾. Daha sonra bu düğümleri diğer görünüşlerde araştırır. Kenarları, yüzeyleri ve ardından cismin şeklini oluşturur. Algoritma görünen yüzeyleri oluşturmaktadır ve çoklu çözümlerin bulunduğu durumlarda çalışmamaktadır. Algoritmadaki adımlar şöyledir:

- a) Her doğru için verilerin girilmesi.
- b) Bu ham verilerin işlem için tekrar düzenlenmesi.
- c) Her görünüşteki kapalı alanların tanımlanması.
- d) Bu kapalı alanların üç boyuttaki şekline dönüştürülmesi.
- e) Bütün kesikli çizgiler için muhtemel kenarların oluşturulması.
- f) Kesikli çizgilerden ayrılmış, 3B'lu yüzeylerin ve kenarların oluşturulması ve

katı cismin tamamlanması.

Wesley ve Markowsky, Markowsky'nin algoritmasını⁽⁷⁾ daha sonra 2B'lu görünümler için geliştirmiştir⁽⁹⁾. 2B'lu görünümlerden bütün çok yüzlü cisimleri bulan çok genel ve komple bir algoritma sunulmuştur. Köşe ve kenarların etiketlenmesi konuyu önceki tel çerçeve algoritmasına götürmüştür. Diğer taraftan bu algoritma cismi oluşturmak için 2B'lu görünümlerin nokta ve çizgilerinden daha fazla bilgiye gerek olmadığını göstermiştir. Görünümler kesikli ya da dolu çizgilerden ibaret olabilir. Görünümlerde etiketlenmenin yapılması, problemin zor olan kısmını belirlemektedir. Cismi görünümlerden oluşturmak, cismi tel çerçeveden oluşturmaktan daha kolay olmaktadır. Algoritma çok sayıda muhtemel çözümlerin bulunması ya da, görünümlerin karmaşık olmaları durumunda bile çözüm üretmektedir. Örneğin, sadece iki görünümle verilen bilgileri kullanarak bütün muhtemel cisimleri oluşturmakta, bunun sonucunda da çok sayıda çözüm ortaya çıkabilmektedir. Bu yüzden üç ya da daha fazla görünümle fazla bilgi sağlanarak çözüme gidilmelidir. Bu algoritma sadece düz çizgilerden oluşan modelleri üretebilmektedir. Algoritma daha sonraki çalışmalar için eğri yüzeyli cisimlere de uygulanabilir gözükmektedir.

Aldefeld eşit kalınlıkta, temel elemanlardan oluşan bir makine parçasının görünümlerine dayanan bir algoritma sunmuştur⁽¹⁰⁾. Metot, 2B'lu üç dik görünüm gerektirmektedir. Aldefeld, karmaşık bir parçayı bir kaç temel alt parçaya ayırmıştır. Algoritma bu basit cisimlerin özelliklerini ve aralarındaki bağıntıları kullanmıştır. Cismin parçaları sadece eşit kalınlığa sahip olduğu takdirde kullanılabilir.

Sakurai ve Gossard, Wesley ve Markowsky'nin çalışmalarını genişleterek düz ve dairesel çizgilerle, görünen ve görünmeyen elemanları ele alabilen bir metot

sunmuşlardır⁽¹¹⁾. Köşeleri, kenarları, yüzeyleri ve sonunda cismi oluşturmak için altı adımda işlem yapılmıştır. Algoritma cismin üç tamamlanmış görünüşünü gerektirmektedir.

2B'lu bilgilerin girişinde yukarıdaki yaklaşımların yanında tamamlanmış ve tutarlı görünüşler gereklidir. Ancak, pratikte, çoğu gerçek mühendislik çizimleri, belli geometrik detayları unuttur. Kullanıcı bu detayları daha sonra bilinen standart temellere dayanarak sağlar. Yoshiura, Fujimara ve Kunii çizime eklenen yorum cümlelerinin kullanımı ile ilgili bir yaklaşım önermiştir⁽¹²⁾. Bu yaklaşım 2B'lu çizimi basitleştirmektedir. Tüm çizimin sayısallaştırılmasından sonra, metin ve geometrik kısımlar karakter tanıma tekniğiyle ayrılır ve yorum cümlelerinin manası sonradan doğal dil işleme tekniğiyle (bunlar gerekli 3B'lu bilgiler sağlayan bilgi tabanını kullanır) çıkarılır. Daha sonra, önceki yaklaşımlarda olduğu gibi, aynı işlemler uygulanır.

2B'lu mühendislik çizimleri bir makine parçasının değişik görünüşlerinden oluşur. Bu mühendislik çizimleri farklı türdeki bir çok temel hacimlerin birleştirilmesidir. Bu görünüşlerin tümü bu birleşik hacimlerin tiplerini, boyutlarını, pozisyonlarını ortaya koyar. Ho Bin, bu temel yaklaşımı kullanarak 3B'lu hacimleri oluşturmuştur⁽¹³⁾. Burada kullanılan temel hacimler 5 çeşittir. Bunlar küp, piramit, silindir, koni ve küredir. Bu temel hacimler programa kullanıcı tarafından tanıtılmaktadır. Bu nedenle algoritma yarı otomatik bir yapıdadır. Bu yaklaşımla bir çok makine parçası oluşturulabilmektedir. Daha ileri çalışmalar olarak bu temel hacimlerin yanında torusu, üçgen ya da altıgen piramitleri, kesik piramit ya da konileri, genel olarak dönele ve doğrusal süpürmeden oluşan hacimleri, iki basit cismin birleşimi olan cisimleri tanıyabilen çalışmalar yapılabilir. Bu çalışma yerinde

ve ölçekli çizilmiş iki ya da üç görünüşü gerektirmektedir.

Remi Liquette iki ya da üç görünüşten katıları oluşturan bir algoritma sunmuştur⁽¹⁴⁾. Kullanılan katılar düz, silindir, koni ya da toroid yüzeylerden oluşmalıdır. Silindir, koni ya da toroid yüzeylerin eksenleri görünüşlere paralel olmalıdır ve düz çizgileri veya daireleri kesmelidir. Algoritma ilk olarak bir tel çerçeve modeli üretir ve ardından çoğunlukla çizilmeyen teğet yüzeyleri, heuristik metot kullanarak tespit eder. Yüzeyler tel çerçeveden meydana getirilir ve algoritma katıyı oluşturmak için modelleri araştırır.

Zen Chen ve arkadaşı 2B'lu dik görünüşlerden 3B'lu katıları otomatik olarak oluşturan bir algoritma sunmuşlardır⁽¹⁵⁾. Görünüşler dolu veya kesikli olmak üzere düz çizgi, daire veya yay içermelidir. Elde edilen cisimler çok yüzlü, silindir, silindir parçası ve bunların birleşiminden oluşabilir. Cismi yeniden oluşturma işlemleri parçalama, yeniden oluşturma ve birleştirme olmak üzere üç aşamada gerçekleşmektedir. İlk olarak girilen çizim, önceden tanımlanan bir kaç alt görünüşe parçalanır. Sonra süpürme operasyonu ve ardından bir düzlemlerle kesme işlemi uygulanarak bu alt görünüşlerin her birine uygun alt parçalar elde edilir. Sonuçta bu alt parçalar, tüm parçayı oluşturmak için kapatılır. Kullanılan teknik, hacimsel esastır. Bu teknik sabit kalınlıkta olmayan çok yüzlü cisimleri de oluşturabilmektedir. Algoritma, koni, küre ve torus da dahil olmak üzere dönel simetrik parçaları içermemektedir.

Dutta ve arkadaşı eğrisel katıları, sadece iki tane düz kenarlı görünüşten elde eden bir algoritma sunmuşlardır⁽¹⁶⁾. Öncelikle verilen iki görünüşe uygun olarak, bütün üçüncü görünüşler elde edilir. Daha sonra bu görünüşlere uygun olarak da katılar oluşturulur.

You'nun geliřtirdiđi algoritma, ilk olarak çizimden tel çerçeveyi oluşturur⁽¹⁷⁾. Tel çerçeve üzerindeki kapalı döngüleri araştırarak yüzeyleri oluşturur. Tel çerçeveden bütün bulunmayan yüzeyler çıkarılır. Ardından cisim euler operatörü'ne dayanarak oluşturulur. Algoritma dik görünüşlerden muhtemel bütün manifold (her kenarı iki yüzeyin paylaştığı) cisimleri oluşturmaktadır. Üç görünüşten fazla ya da eksik çizilmiş görünüş sayılarını da ele alabilmektedir. Sonuca çabuk ulaşmak ve derinlik bilgilerini elde etmek için, her aşamada kesikli ve sürekli çizgiler kullanılmıştır. "Kenar sayma" algoritması, tel çerçevenin görünüşlerinin giriş çizimi ile aynı olup olmadığını bulmak için kullanılmıştır.

Masuda'nın çalışmasında 2B'lu dik görünüşlerden 3B'lu katı model oluşturulmaktadır⁽¹⁸⁾. Bu işlem non-manifold topolojiye (her kenarı 2 den fazla yüzeyin paylaşabildiđi) ve varsayım tabanlı (assumption-based truth maintenance system, ATMS) bir sistem temeline dayanmaktadır. Ardından eđer 2B'lu resimde bir hata varsa, hatayı bulma ve düzeltme yapmaktadır. Kullanılan sistem çok iyi bir dönüřtürme işlemi sağlamaktadır. Burada elde edilen bazı sonuçlar řu şekildedir. Bu makale hatalı çizimleri de ele alabilmek için iki metot önermiştir. Bu metotlar řu şekildedir;

- a) Birinci metot diđer görünüşlerde karşılığı bulunmayan çizgileri arařtırmaktır. Katıyı oluřturma işlemi sonunda hiç bir katı model oluřturulamazsa, ön işlem olarak uygun bir ya da iki görünüş arařtırılır ve bunlara göre muhtemel katılar oluřturulur. Daha sonra verilen dik görünüşler ile, oluřturulan katıların görünüşleri karşılaştırılır ve uygun katı bulunur. Bu katının görünüşü ile orijinal görünüşler kontrol edilerek eksik çizgiler bulunur.
- b) Diđer metot eksik çizgileri yeniden düzeltmektir. Eđer uygun hücrenel model

dik görünüşlerden elde edilemezse, yatay ya da dikey çizgiler dik görünüşlere, 3B'lu modelin en az bir kenarına uygun gelecek şekilde eklenir.

Shum'un çalışmasında altı adet dik görünüş kullanılmaktadır⁽¹⁹⁾. Bu dik görünüşler önce üç gruba ayrılmakta ardından her grup çifti içerisinde biri diğeri üzerinde süpürülerek basit alt kalınlığı olan cisimler elde edilmektedir. Ardından bu cisimler birleştirilerek 3B'lu katı cisim elde edilmektedir. Görünüşlerdeki bütün çizgiler sürekli çizgi olmak zorunda. Bu metotla tek koordinat eksenli çok yüzlü ve dik silindirler ele alınabilir. Algoritma "Autolisp" dilinde yazılıp, AutoCad ortamında çalıştırılmıştır.

Kuo'nun çalışmasında üç görünüşlü bir mühendislik çiziminden kavisli yüzeyli katıları oluşturan güçlü bir algoritma sunulmuştur⁽²⁰⁾. Başlangıçta 3B'lu tel çerçeve oluşturulur. Sonra bütün yüzeyler minimum iç açı prensibiyle tespit edilir. Ardından yüzeyler katı cisim oluşturmak için birleştirilir. Yüzeylerin yönünün doğruluğu "Moebius Kuralı" ile kontrol edilir. Bu makalenin ana katkısı şu şekilde özetlenebilir. Algoritma katıyı, işlemleri yaparken alt parçalar oluşturmadan gerçekleştirir. Bu yüzden çözüm için harcanan zaman daha kısadır. Algoritma dışarıdan direk 3B'lu tel çerçeve modeli alıp, bundan katı modeli oluşturabilmektedir.

Shin, katı modelleri üç dik görünüş içerisindeki basit geometrik elemanların özelliklerini ve aralarındaki topolojiyi kullanarak elde eden, etkili bir algoritma sunmuştur⁽²¹⁾. Algoritma 3B'lu modelleri çok hızlı bir şekilde oluşturmaktadır. Burada ele alınabilen cisimler çok yüzlü cisimler ile silindir ve torus gibi eğri yüzeyli cisimlerdir. Yüzeyler sonlu sayıda eş düzlemlili olmayan çok kenarlılardan oluşmalıdır. Kavisli yüzeyler temel eksenlerden birine paralel olmak zorundadır.

Tanaka'nın çalışmasında⁽²²⁾ 2B'lu üç görünüş içeren bir montaj resminden, içerisindeki 3B'lu cisimlerin çizimlerini katı eleman denklemleri kullanarak elde eden bir algoritma sunulmuştur. Yazarlara göre montaj resmini parçalara ayırma işlemi şu açılardan öneme haizdir.

- a) 2B'lu parçaların çizimlerinin, montaj çizimlerinden çıkarma işlemi, parçanın sayısına bağlı olarak çok zaman almaktadır.
- b) İşlemi tasarımcı yerine operatörler yaparsa bir çok hatalar meydana gelmektedir. Hataları azaltmak açısından metot önemli olmaktadır.

Bu metotta dik görünüşlerden sırayla tel çerçeve modeller, yüzey modeller ve basit katı elemanlar oluşturulmaktadır. Basit bir katı eleman, yüzeylerden oluşan kapalı bir bölgedir. Katı elemanlar ise parçanın katı modelinin bir parçasıdır. Elde edilen katı elemanlar eğer gerçek cisim içerisinde mevcut ise "doğru" eleman olarak, yok ise "yanlış" eleman olarak adlandırılıyor. Eğer problemin birden fazla çözümü varsa, bütün bu çözümleri bulabilmektedir. Metot PC'de uygulanmış ve çeşitli 2B'lu montaj resimlerinde test edilmiştir.

2B'tan 3B'ta dönüşüm yapan algoritmalar uzun bir zaman dilimine dağılmış olmasına rağmen, bütün şekilleri çözebilen genel bir algoritma henüz geliştirilememiştir. Bunun sebebi yöntemin yapısından kaynaklanmaktadır. Pratik hayatta 2B'lu çizimler bir parçayı yeterince tam anlatamadığı için, kesit görünüşlere ve metinsel açıklamalara başvurulmaktadır. Yine de yapılan her çalışma olayı değişik bir açıdan ele almış ve en azından belli özellikteki parçalara çözüm getirmiştir. Yapılan bu çalışmalardan bazılarının kısa özeti Çizelge 1.1.'de verilmiştir. Konuyla ilgili bir çok çalışma daha yapılmıştır⁽²³⁻²⁸⁾.

Çizelge 1.1. 2B'dan 3B'ta dönüşüm yapan bazı algoritmaların kısa açıklaması

Yazar	Çıktı	Yüzey Tipi	Çözüm Sayısı
Idesawa ⁽⁵⁾	Tel çerçeve	Düzlem	Tek
Lafue ⁽⁶⁾	Tel çerçeve	Düzlem	Tek
Wesley ⁽⁹⁾	B-rep	Düzlem	Çok
Preiss ⁽⁸⁾	Tel çerçeve	Düzlem	Tek
Aldefeld ⁽¹⁰⁾	CSG	Düzlem, silindirik yüzey	Tek
Sakurai ⁽¹¹⁾	B-rep	Düzlem, silindirik yüzey	Tek
Ho Bin ⁽¹³⁾	CSG	Düzlem, silindirik yüzey	Tek
Chen ⁽¹⁵⁾	CSG	Düzlem, silindirik yüzey	Tek
Lequette ⁽¹⁴⁾	Tel çerçeve	Düzlem, kuadrik yüzey	Çok
Gujar ⁽²⁹⁾	Tel çerçeve	Düzlem yüzey	Çok
Gau ⁽³⁰⁾	Tel çerçeve	Düzlem, kuadrik yüzey	Tek
Lung ⁽³¹⁾	Tel çerçeve	Düzlem, silindirik yüzey	Tek
Chen ⁽³²⁾	Tel çerçeve	Düzlem yüzey	Tek
Yan ⁽³³⁾	Tel çerçeve	Düzlem yüzey	Çok
You ⁽³⁴⁾	Tel çerçeve	Düzlem, kuadrik yüzey	Tek
Kuo ⁽²⁰⁾	Tel çerçeve	Düzlem, kuadrik yüzey	Çok
Shin ⁽²¹⁾	Tel çerçeve	Düzlem, kuadrik yüzey	Tek
Tanaka ⁽²²⁾	B-Rep	Düzlem	Çok

1.1.2. 2B'lu Çizimlerden Unsur Çıkarımı

Makine resimleri, tasarım bilgisini saklamada kullanılmasına rağmen, 2B'lu girdilerden unsur tanıma konusuna çok az ilgi gösterilmiştir⁽³⁵⁾.

Meeran ve Pratt makine resimlerinden unsur çıkaran bir sistem tanıtmaktadır⁽²⁾. Bu sistem fatura veya dikdörtgen cep gibi basit üretim unsurlarının, çizgi, daire gibi elemanların kapalı döngüleri ile en az bir görünüşte bulunabileceği kabulüne dayanmaktadır. Bu döngülerin şekil eşleştirme ile belirlenmesi ve arkasından üç görünüşten çıkarılan elemanların birleşimi, unsur tanıma metodunun temelidir. Tanıma işlemi şu adımlardan oluşmaktadır.

1. Tüm bağımsız basit unsurlar, şekil eşleştirme yoluyla tanınır.
2. Diğer birleşik elemanların, etkileşimli unsurlara ait olup-olmadığı, mantık-tabanlı üretim kuralları kullanılarak incelenir ve etkileşimli unsurlara ait olan elemanlar, daha sonra işleme tabii tutulmak üzere ayrılır.
3. Kalan birleşik elemanlar, genelleştirilmiş negatif veya pozitif unsurları başlığı altında toplanır ve gruplandırılır.

Suharitamrong ve Motavalli, 2B'lu unsur çıkarma ve teknik resim temel görüntülerinin yorumlanmasını birleştirerek, unsur temelli modeller oluşturmayı amaçlamıştır⁽³⁶⁾. Oluşturulacak unsur modeller ile, tasarım ve üretim arasında doğrudan ilişki kurulacağı gibi diğer mühendislik işlemlerinin desteklenmesi de mümkün olabilecektir.

Teknik resimlerin unsur tabanlı modellerini oluşturma konusu, Bottoni ve arkadaşları tarafından araştırılmış ve bu amaçla bir bilgisayar sistemi geliştirilmiştir⁽²⁴⁾. Geliştirilen bu sistem, kural tabanlı ve veri bağımlı işlem yapabilmektedir. Ayrıca sistem, kullanıcı tanımlı biçim unsurları kullanmakta ve

değişebilir dil muhakemesi yapabilmektedir.

Cugini ve arkadaşları, teknik resim veya 3B'lu katı modellerden unsur çıkaran grafik-gramer ve yeniden yazım kurallarına dayalı bir sistem tanıtmaktadır⁽³⁷⁾. Burada çıkartılan unsurların çıktısı, bir karışım modeldeki hacimler şeklinde temsil edilmektedir.

Aomura, makine resimlerinden üretim için otomatik 3B'lu modeller oluşturan bir tasarım destek sistemi tanıtmaktadır⁽³⁸⁾.

Hwang ve Ulman, 2B'lu serbest el çizimlerinden (kroki) unsurları tanıyabilecek bir sistem tanıtmaktadır⁽³⁹⁾. Burada girdi, 3B'lu bir parçanın 2B'lu krokisidir. Bu sistem kullanıcının etkileşimli krokiler çizmesine izin vermekte ve arkasından bu parçaya ait unsurları tanımaktadır.

Liu ve arkadaşları 2B'lu CAD çizimlerinden 3B'lu parçayı, şekil özelliklerinden otomatik olarak yeniden oluşturan bir metot sunmuşlardır⁽⁴⁰⁾. İlk olarak burada IGES formatındaki 2B'lu mühendislik çizimlerinden düğüm ve kenarlar, böl parçala metoduyla elde edilmektedir. Sonra şekil özelliklerini birleştirmek için bir kurallar dizisi geliştirilmiştir. Her bir şekil özelliğini kaydetmek için yeni bir "grafığe bağlı şekil özelliği tanımlayan" (FFAG) yapı tasarlanmıştır. Sonunda süpürme operasyonu ve hacimlerin kesişimi yaklaşımı ile kalan 3B'lu cisim hızlı bir şekilde kurulur. Bu elde edilen 3B'lu cisim FFAG prosedürünün temelini oluşturmaktadır. Bu prosedürdeki tanımlanmış bütün şekil özellikleri 3B'lu temel parçanın üzerine eklenen veya çıkarılan olarak sınıflandırılır.

Çalışmada 3B'lu parçanın oluşturulması ana fikri üç açıdan ele alınmaktadır.

- 1) Karmaşık bir parça önceden tanımlanmış basit alt parçaların birleşimi halinde

gösterilebilir. Bu temel alt parçalar, 2B'lu modellere bağlı bir sınıflandırma ile tanımlanabilir.

- 2) Bütün 2B'lu gösterimler, özelliklerine bağlı olarak basit terimler halinde tanımlanıp oluşturulabilir.
- 3) Bir 3B'lu parça, bu 3Blu parçanın boyut, tolerans, malzeme gibi özellikleri de dahil olacak şekilde, 3B'lu şekil özellikleri yapıştırma kesme gibi işlemlerden geçirilebilir.

Tyan, 2B'lu resimlerden parça üzerindeki işleme unsurlarını çıkaran bir algoritma sunmuştur⁽⁴⁾. İşleme unsurlarının 2B'lu resimlerden çıkarılmasının, 3B'luya nazaran çok daha kolay olduğunu bildirmektedir. Algoritma her unsurun beş özelliğini tespit etmektedir:

- 1) Pozitif/Negatif olma durumu,
- 2) Dışarıdan erişme yönü,
- 3) Kör veya tam delik durumu,
- 4) Tabanın kapalı/açık olma durumu,
- 5) Tabanın geometrik şekli,

Burada ele alınabilen unsurlar sadece birbirinden ayrı, bağımsız, bitişik olmayan unsurlardır. Bu algoritmanın uygulanması için bir program hazırlamışlardır (FlexiCAD). Bu program hem 2B'lu hem de 3B'lu (CSG, B-rep, Tel Çerçeve) modellerden unsurları çıkarmaktadır. 3B'lu cisimleri ele alırken önce 2B'lu görünüşleri oluşturmaktadır. Bu dönüştürme sırasında bazı bilgiler kaybolabilmektedir. Programda kullanılan unsur tanıma sistemi unsur çıkarımı ve unsur belirleme olmak üzere iki kısımdan oluşmaktadır. Birinci kısımda özel geometrik alt parçalar (silindir, küp vs.) oluşturulur. Bu alt parçalar daha sonra unsur

belirleme kısmına gönderilir ve burada unsur kütüphanesindeki unsurlarla karşılaştırılarak unsurun tipi belirlenir.

FlexiCAD programı önce üç görüntü içinden 2B'lu çerçeve modelleri çıkarır. Bu çerçeveler derinlik boyunca süpürülerek 2.5B'lu alt parçalar oluşturulur.

Unsur tanıma prosedürüne geçmeden önce giriş çizimleri ön işleme tabi tutulmaktadır. Ön işlemcide çizgilerin parçalanması ve en dış çerçevenin ortaya çıkarılması olmak üzere iki tane önemli adım vardır. Çizgilerin parçalanması işlemi görüntülerdeki her çizgi adımından parçalanarak yapılır. Dış çerçevenin ortaya çıkarılması ise iki nedenle yapılır:

1. Her görünüşü içine alan çerçevenin bulunması.
2. En dış çerçevedeki çizgilerin sayısını bulma. Bu çizgiler unsurların özelliklerini belirlemede kullanılmaktadır.

Alt parçalar girdi olarak kullanılırken iki kısma ayrılır:

1. Bitişik alt parçalar,
2. Bağımsız alt parçalar. Bitişik alt parçada eğer alt parçanın 2B'lu çerçevesi görünüşün en dış çerçevesine temas ediyorsa bu adı alır. Temas etmiyorsa bağımsız alt parça adını alır.

Sonuç olarak, bu çalışmada bağımsız ve bitişik unsurların özelliklerini bulan bir algoritma geliştirilmiştir. 2B'lu resimlerden alınan girdi önce 2.5B'lu alt parçalara dönüştürülmekte ve buradan unsur özelliklerini bulan güçlü bir algoritma geliştirilmiştir. Algoritma C dilinde yazılıp denenmiştir.

Bu metodun sınırlı kaldığı durumlar ise şunlardır:

1. Alt parçaların sabit kalınlığa sahip olması gerekir.

2. Algoritma sadece unsur kütüphanesindeki unsurları tanıyabilmektedir.
3. Bu unsurların tamamı kesişmeyen unsur olmalıdır.

Ganesan'nın yaptığı çalışmada, 2B'lu projeksiyonlardan parçanın kesişen unsurlarını çıkaran bir yeni algoritma sunulmuştur⁽¹⁾. Kullanılan algoritmanın bazı önemli tarafları vardır. Bunlar şu şekilde özetlenebilir:

1. 2B'lu resimleri girdi olarak kullanmaktadır. Bu ise algoritmayı çok daha basit ve kolay kılmaktadır. Aynı zamanda mevcut çok sayıdaki 2B'lu resimleri kullanma imkanı vardır.
2. Bu metot kesişen eğri yüzeyleri ele alabilmekte, ayrıca basit eğri unsurları çıkarabilmektedir.
3. Böl parçala usulü ile işlemleri yürütmektedir. Bunun için önce basit unsurları çıkarmakta ardından birleşik karmaşık unsurları çıkarmaktadır. Bu da çeşitli aşamalarda işlemleri basitleştirmektedir.
4. Bu metot mevcut herhangi bir BDT sisteminden destek almamaktadır. Bu nedenle kendi başına kullanılabilir ve her hangi bir ticari BDT yazılımı ile entegre edilebilir.

Burada incelenen ve 2B'lu çizimlerden unsurları çıkaran ve cismin şeklini veren makaleler basit şekiller üzerinde başarılı bir şekilde unsurları bulabilmektedir. Bu algoritmalar arasındaki farklar Çizelge 1.2. de kısaca özetlenmiştir.

2B'lu çizimden 3B'lu cisimleri oluşturma ve unsurları çıkarma konusu yaklaşık 30 yıldır incelenmekte ise de, çok az sistem kullanılabilir durumdadır. Bunun sebebi, makine parçalarının izdüşümlerinden oldukça farklı olması ve bunların farklı yorumlanabilir geometrik ifadelerle sahip olmasıdır. Unsur tanıma veya 3B'lu katı model oluşturmada karşılaşılan en önemli zorluk, 2B'lu temsil

şeklinin tam olmamasıdır. 2B’lu veri, bir nesneyi tam olarak belirleyecek anlamsal ifadeler içermemektedir. Üç görünüşün bir nesneyi tam olarak temsil ettiğinin kabulü, uygulamalarda pek gerçekçi değildir.

Çizelge 1.2. 2B’lu resimlerden unsurları çıkaran bazı algoritmalar arasındaki farklar.

Yazarı	Bağımsız / Birleşik	Çıkıntı /Çukur	Girdi
Meeran-93 ⁽²⁾	Bağımsız, Birleşik	Çıkıntı ve Çukur	DXF file
Liu-94 ⁽⁴⁰⁾	Bağımsız	Çıkıntı ve Çukur	3 görünüş
Tyan-98 ⁽⁴⁾	Bağımsız	Çukur	3 görünüş
Genesan-98 ⁽¹⁾	Birleşik	Çukur	3 görünüş

2B’lu prizmatik parçalardan unsur çıkarımın yanında, 2B’lu silindirik parçalardan da unsur çıkarımıyla ilgili bir çok çalışma yapılmıştır⁽⁴¹⁻⁴⁴⁾.

Aslan 2B’lu silindirik parçalar için bir uzman sistem geliştirmiştir⁽⁴¹⁾. Bu sistem silindirik parçaların tasarımından üretimine kadar döngüyü içeren, üretken işlem planlaması yaklaşımıyla oluşturulmuş ve iki tezgah son işlemciyi içeren bir programdır.

Aslan ve Şeker, silindirik parçalardaki işleme özelliklerinin tanımlanması amacıyla DXF dosyasından bilgi çıkarımına yardımcı olacak bir çalışmanın sonuçlarını sunmuşlardır⁽⁴²⁾. Buradan elde edilen bilgiler bir son işlemci uzman sistemin geliştirilmesi için kullanılmıştır.

1.1.3. 3B'lu Modellerden Unsur Çıkarımı

3B'lu modelleme ve unsur çıkarımı üç temel yaklaşıma dayanmaktadır.

Bunlar:

- a) Mevcut katı modellerden (CSG-yapısal katı modelleme ve B-rep-sınır temsili) unsur çıkartma,
- b) Başlangıçta unsur tabanlı tasarım,
- c) Her iki yaklaşımının birleştirilmesi.

Katı modellerden unsur tanımadaki ana amaç, üretim mühendisinin tasarım verilerini yorumlama ve imalat kararları vermesini ortadan kaldırmaktır. Katı modellerden unsur çıkartılmasında, yüzey işleme veya toleranslar gibi mevcut olmayan bilgilerin elde edilmesi mümkün değildir. Ayrıca çok basit unsurların bile elde edilmesinde kullanılan algoritmalar, oldukça karışık ve hata eğilimlidirler. Unsur tabanlı tasarım, tasarımın işlem verimliliğini artırmakla birlikte, tasarlanan yapının ifade yeteneğini kısıtlamaktadır.

3B'lu katı modellerden unsur çıkarımını aşağıdaki şekliyle kısaca özetleyebiliriz:

Yüzey-temsillerden (B-rep) unsur çıkarımı ile ilgili olarak birçok metot geliştirilmiştir. *Söz-dizimsel patern* teknikleri⁽⁴⁵⁻⁵⁰⁾, 2B'lu unsurları tanımada, kelimelerle temsil edilen bir geometrik elemanlar dizini kullanmaktadır. Bu metodun 3B'a uygulanması zordur. *Kural-tabanlı* metotlar^(46,51-56), unsur tanımada üretim kurallarını kullanmaktadır. Bir üretim kuralı; dışbükey, diklik veya bitişiklik gibi, modeldeki elemanlar için bir dizi gerekli ve yeterli şartı tayin etmektedir. Nesne, karmaşık ve bir çok yüzeye sahipse, kural kümesi ile yüzey tarama oldukça fazla

zaman almaktadır. *Grafik-tabanlı* tekniklerle^(53,57-64) unsur tanımada parçanın topolojik biçimi, bir grafik (bağlantılar grafiksel gösterimli) olarak temsil edilmektedir. Genelde bu grafiğin düğümleri, nesnenin yüzeylerini, grafiğin kenarları ise, nesnenin kenarlarını belirtmektedir. Daha sonra bu grafikte yapılacak bir arama işlemi ile, grafikte temsil edilen parçaya ait unsurlar hakkında bazı özellikler belirlenmektedir. Bu teknik, fazla bilgisayar hafızası ve zamana gereksinim duymaktadır. *Dışbükey-kabuk* algoritmaları⁽⁶⁵⁻⁷⁴⁾, nesne ve işlenmeden önceki ham nesne arasındaki farkları belirlemektedir. Daha sonra bu algoritma, parça üzerinden talaş kaldırma ile boşaltılacak hacimleri belirlemek için yeniden düzenlemektedir. Bu metodun ana dezavantajı, unsur tanımada sadece oyukları bulmasıdır. Çıkıntılarda başarısız olmaktadır. *Hücre-tabanlı* tekniklerde⁽⁷⁵⁻⁷⁸⁾, parça veya parçadan talaş kaldırılacak hacimler, temel hacimlere (hücrelere) dönüştürülmektedir. Bu hücrelerin tekrar birleştirilmesi ile orijinal parçadaki unsurlar belirlenmektedir. *Tanı-tabanlı muhakeme* metodunda⁽⁷⁹⁻⁸³⁾ sadece ipuçları veya tanımlar ve tam olarak bilinmeyen unsurlar öncelikle oluşturulur. Daha sonra unsurlar, ipuçlarının tekrar birleşimi veya tanı toplanması ile belirlenmektedir. Bu teknik, etkili unsur tanımayı mümkün kılmaktadır. Unsur tanımada *yapay sinir ağları* kullanma⁽⁸⁴⁻⁸⁹⁾, diğer tekniklere göre daha hızlıdır. Ancak burada da ağın eğitimi gerekmektedir. Bunlarla birlikte bir çok yöntem daha geliştirilmiştir. Bunlar *heuristik yaklaşım*⁽⁹⁰⁾, *hacim parçalama*^(76,77,91-97), *geri besleme*⁽⁹⁸⁻¹⁰⁰⁾ gibi yöntemlerdir.

CSG temsillerden unsur çıkarımı⁽¹⁰¹⁾, B-rep'e göre daha az ilgi çekmiştir. Bu temsilden unsur çıkartabilmek için, önce temsilin basitleştirilmesi gerekmektedir. Buradaki olumsuzluklar; CSG'de temsil tekliliğinin olması ve temel eleman ile unsurlar arası birebir ilişki eksikliğidir.

Unsur tabanlı tasarımla ilgili olarak da bir çok çalışma yapılmıştır⁽¹⁰²⁻¹¹⁶⁾. Unsur-tabanlı tasarımda, genelde tasarımcı, önceden tanımlı parametrik hacimleri unsur kütüphanesinden seçmekte ve bunlara Boolean operasyonları uygulayarak parça modellenmektedir. Bu yaklaşım, tasarımı hızlandırmakta ve tasarım sürecinde oluşan bilgilerin sonraki uygulamalara aktarılmasına izin vermektedir.

Unsur-tabanlı tasarımla ilgili diğer bir problem ise, tasarımcının unsur kütüphanesinde bulunan ve önceden tanımlı unsurlarla sınırlı uzayda tasarım yapmasıdır^(117,118). Ayrıca bu kütüphane kullanımının oldukça fazla üretim bilgisine ihtiyaç göstermesi ve erken tasarım ve üretim unsurları arasında, bire-bir ilişki olmayabilir. Makine resimlerinin, her zaman unsurlar cinsinden tanımlanamayacağı da (ki bu gerekli de değildir) bir gerçektir. Ayrıca döküm veya pres şekillendirme ile üretilcek karmaşık parçaların tam modellenmesi, unsur tabanlı tasarım tekniği ile güç olabilir.

Unsur tabanlı tasarım ve unsur tanıma yaklaşımlarının her biri, bazı güçlü ve zayıf özelliklere sahiptir⁽¹¹⁷⁾. Bu iki metottan hiç birisi, tek başına yeterli olmamakta ve böylece bu iki metodun birlikte uygulandığı bir yaklaşım, en iyi ve etkili çözüm sağlamaktadır^(85,118, 119).

Unsur çıkarımıyla ilgili benzer bir çok çalışma daha yapılmıştır⁽¹²⁰⁻¹⁴⁷⁾.

1.1.4. Çalışmanın Amacı

Yapılan bu çalışmanın amacı; 2B'lu mühendislik çizimlerinden parçanın 3B'lu modelini oluşturmak ve üzerindeki işleme unsurlarını çıkarmaktır. Çalışma, önceki benzer çalışmalardan bir çok şekilde farklılık göstermektedir.

- a) Önceki çoğu çalışmanın amacı⁽¹⁻²²⁾, 2B'lu çizimlerden 3B'lu modelleri oluşturmaktır. Bu tez çalışmasında ise 2B'lu çizimlerden, hem 3B'lu modeller oluşturulmakta, hem de modelden unsurların çıkarılması gerçekleştirilmiştir.
- b) Önceki çalışmaların bir kısmı yalnız düzlemsel yüzeyli parçaları, bir kısmı ise yalnız silindirik yüzeyli olanları, diğer kısmı ise her ikisini de kapsayacak şekilde ele almıştır (Tablo 1.1). Bu çalışma ise daha çok işleme metodu dikkate alınarak BSD dik işleme merkezlerinde üretimi yapılabilecek parçalar ile sınırlandırılmıştır. Bu parçaların taban kısmı genelde düz, üst ve yanal kısımlarda ise farklı işleme özellikleri mevcuttur. Bu farklı özellikteki yüzeyler düz, eğik ve kavisli olabilmektedir.
- c) Daha önceden yapılan çalışmalar, 3B'lu modeli, Tel Çerçeve ve Hacim Metodunu kullanarak elde etmektedir (Tablo 1.1). Bu çalışmada ise, detayları ileride verileceği gibi, 3B'lu yüzeyler, ilk defa uygulanan görüntüleme metoduyla temel görünüşteki yüzeylerden "Tabanları" oluşturmakta, daha sonra "Tavan" ve "Süpürme Taban" olarak adlandırılan yüzeyleri tespit ederek ve süpürme yöntemini kullanıp, modeldeki unsurları oluşturmaktadır. Bu unsurlar da ham prizmatik malzemeden çıkarılarak 3B'lu model elde edilmektedir.
- d) Daha önceden 2B'lu çizimden unsur çıkarımı üzerine yapılan çalışmalarda^(1,4), unsurların 5 özelliğini (pozitif/negatif durumu, dışarıdan erişim yönü, tam/kör delik durumu, sınırın kapalı/açık olma durumu, taban sınırının geometrisi) tespit etmektedir. Bu çalışmada ise unsurların

şu özellikleri tespit edilmektedir:

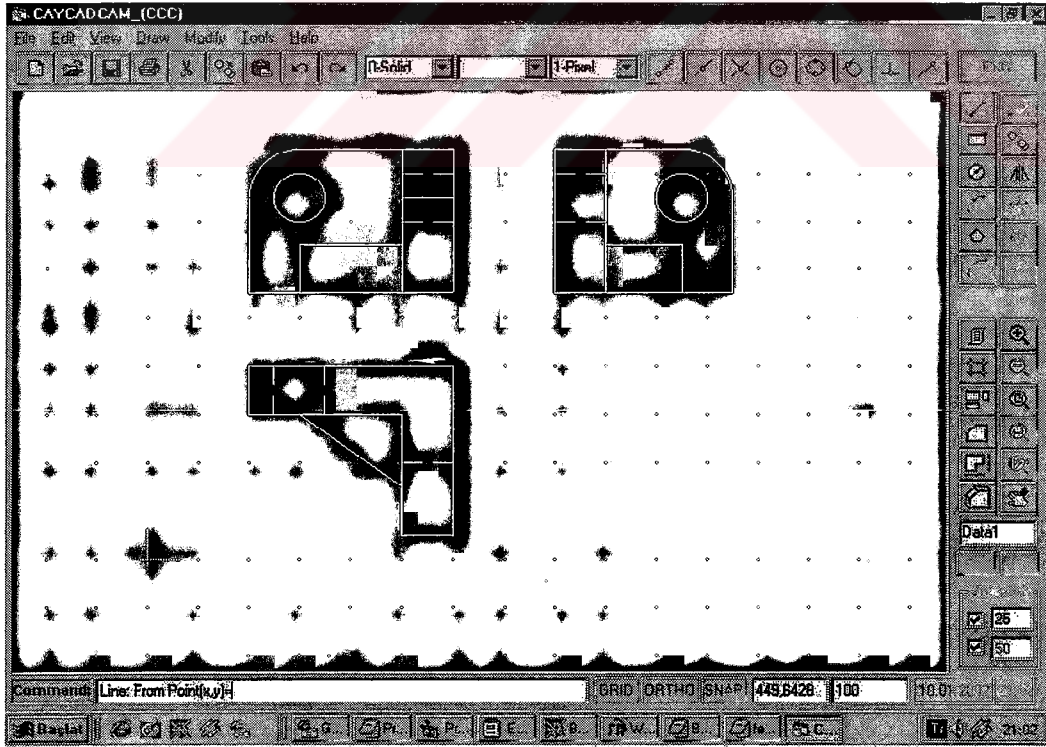
- 1) 16 çeşit unsurun tipi ve geometrisi belirlenmektedir.
 - 2) Bütün unsurlar negatif unsur olarak ele alınmaktadır.
 - 3) Unsurun üzerindeki bütün yüzeyler için dışarıdan erişimin mümkün olup olmadığı bulunmaktadır.
 - 4) Unsurun tam/kör delik durumu tespit edilmektedir.
 - 5) Unsur tabanının kapalı/açık olma durumu belirlenmektedir.
 - 6) Unsurun taban geometrisinin şekli tespit edilmektedir.
 - 7) Kendisinden önce işlenmesi gereken komşu negatif unsur varsa, onunla olan ortak yüzeyi belirlenmektedir.
 - 8) Parça üzerindeki unsurların işleme sırası tespit edilmektedir.
- e) Önceki çalışmalar cismi içine alan prizmanın uygulamalarda kullanılabileceği konusyla ilgilenmemişlerdir. Bu çalışmada ise bu prizma, pratikte parçayı işlemek için ele aldığımız ham malzeme olarak görülmüş ve ona göre programda kullanılmıştır. Eğer prizmanın sınırları 2B'lu çizimde görüntüslere eklenmezse, parçayı içine alan en küçük hacim olarak alınmaktadır. Prizmanın görüntüsleri, 2B'lu çizim esnasında görüntülerin dışına ek bölüm çizgileri çizilerek eklenmektedir.
- f) Bu çalışmada oluşturulan 3B'lu modelin geri kontrolü yapılmaktadır. Üzerinde bulunamayan unsurlar varsa onlar geri kontrolle tespit edilmektedir.
- g) Parça üzerinde karmaşık unsurların bulunduğu durumlarda, bu unsurları parçalara ayırıp, temel unsurlar cinsinden ifade edebilmektedir.

2. MATERYAL VE YÖNTEM

2.1. 2B'lu Çizim Programı

2.1.1. Programın Yapısı

Bu program 2B'dan 3B'ta dönüşümü sağlayacak algoritmanın uygulanabilmesi için, 2-B'lu çizimleri programın kendi içerisinde sağlamak amacıyla Visual Basic'te hazırlanmıştır (Şekil 2.1). Program 2B'lu temel çizim komutlarını kapsamaktadır. Kullanım şekli ve komutların uygulanışı AutoCad ile benzerdir. Kullanılan butonların resimleri de AutoCad'den alınmıştır. Böylece AutoCad kullanıcılarının programı daha rahat kullanması sağlanmıştır.



Şekil 2.1. 2B'lu çizim için hazırlanan programın ekran görüntüsü.

Programda kullanılan komutlar şunlardır:

- a) Çizim komutları (Draw): Doğru (Line), Daire (Circle), Yay (Arc), Dikdörtgen (Rectangle) ve Çokgen (Polygon)
- b) Düzeltilme komutları (Modify): Sil (Erase), Kopyala (Copy), Aynala (Mirror), Taşı (Move), Döndür (Rotate), Buda (Trim), Uzat (Extend)
- c) Görüntü komutları (View): Büyüt (Zoom Pozitif), Küçült (Zoom Negatif), Hepsini Göster (Zoom All), Pencereyi Göster (Zoom Window), Önceki Ekran (Zoom Previous), Ekran Kaydır (Pan), Ekran Tazele (Redraw).

Çizgi tipi olarak sürekli (solid), kesikli çizgi (dash), eksen çizgisi (dash dot), noktalı çizgi (dot dot) ve kesikli çift nokta çizgisi (dash dot dot=divide) çizilebilmektedir. Komutlar hem butonla hem de ekranın altındaki metin satırından girilebilmektedir. Ayrıca koordinatlar mutlak ve izafi olarak metin satırından verilebilmektedir. Snap ve Grid komutları, özel nokta yakalama komutları (uç nokta, orta nokta, kesişim, merkez) kullanılabilir. Çizgilere kalınlık ve renk verilebilmektedir.

Programda çizimi oluşturan çizgiler veri tabanında toplanmaktadır. Daha sonra veri tabanından bilgiler alınarak algoritma içerisindeki işlemlerde kullanılmaktadır. Komutlar veri tabanına yazılırken, her komutun değişkenlerinin içeriği veri tabanındaki ilgili bölümlere yazılır. Veri tabanındaki bölümler şöyledir:

- 1) Eleman: Çizim elemanın adı bu bölüme yazılmaktadır. Üç çeşit eleman vardır. Bunlar; doğru, daire ve yaydır. Dikdörtgen ve çokgen komutu ile çizilen çizgiler veri tabanına doğru olarak kaydedilmektedir.

- 2) PointX1, PointY1, PointX2, PointY2: Doğru ve yayın, başlangıç ve bitiş noktalarının koordinatları bu bölümlere yazılmaktadır.
- 3) MerkezX, MerkezY: Dairenin, yayın çokgenin merkez noktasının koordinatları bu bölüme kaydedilmektedir.
- 4) Yarıçap: Dairenin, yayın ve çokgenin yarıçap değeri bu bölüme yazılır.
- 5) Alfa, Beta: Yayın başlangıç ve bitiş açılarının pozitif x eksenine ile yaptığı açılar kaydedilmektedir.
- 6) Renk: Çizgilerin çizileceği renkler bu bölümde tutulmaktadır.
- 7) Kalınlık: Çizgilerin piksel cinsinden kalınlık değeri bu alanda tutulmaktadır.

2.1.2. Programın Algoritması

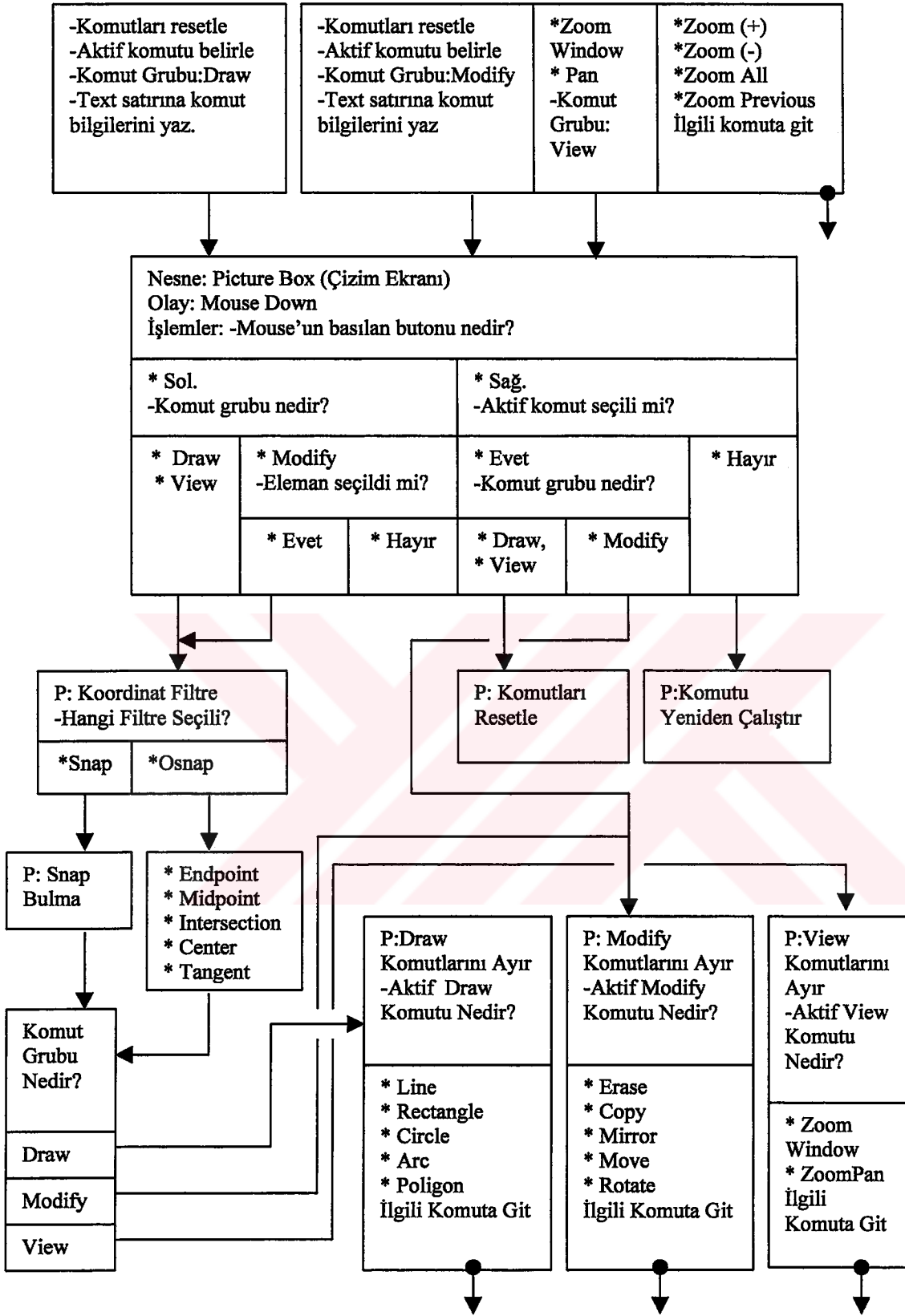
Visual Basic'te program yazılırken nesnelere kullanılmaktadır. Her nesne, üzerinde ise gerçekleştirilen "klik, çift klik, üzerinde gezme" gibi çeşitli olaylara karşı duyarlıdır. Her olayın ise kendi içerisinde çalıştırması gereken kodları bulunmaktadır. Nesne üzerinde bir olay gerçekleşirse, içerisindeki ona ait kodları çalıştırmaktadır. Örneğin, "Aç" düğmesine basıldığında dosyaları seçmek için kullanılan pencere çıkar. Burada "aç" düğmesi bir nesne, üzerine fare ile basma olayı "klik" olayıdır. Bu olayın içerisinde de pencereyi açmak için yazılan satırlar ise o komuta ait program kodlarıdır.

Visual Basic'de kullanılan her değişkenin tanımlanması gerekir. Eğer tanımlanmazsa hafızada en çok yer kaplayan 16 bitlik "variant" değişkeni olarak tanımlanır. Bu da, programın çalışma hızını ve performansını azaltır. Değişkenler programın nerelerinde kullanılacaksa, sadece o bölümleri kapsayacak düzeyde

tanımlanmalıdır. Böylece hafızada ihtiyaç kadar yer ayrılmış olur. Bunun için değişkenlerin tanımlanabileceği en alt düzeyden en üst düzeye doğru aşamalar; komut satırında tanımlama, prosedür düzeyinde tanımlama, form düzeyinde tanımlama ve proje düzeyinde tanımlama şeklindedir.

Programın algoritması Şekil 2.2’de verilmiştir.





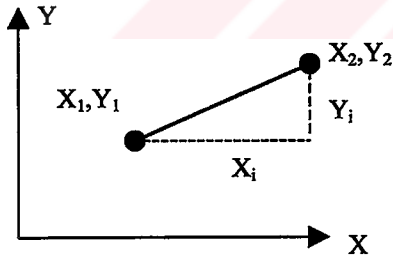
Şekil 2.2. 2B'lu çizim programının algoritması.

2.1.3. Kullanılan Komutlar

2.1.3.1. Çizim Komutları (Draw)

2.1.3.1.1. Doğru Komutu (Line)

Doğru komutu, iki uç noktasının koordinatı verilerek çizilir. Komutu çalıştırdıktan sonra farenin ilk bastığı nokta çizginin başlangıç noktası, ikinci bastığı nokta ise bitiş noktası olarak atanmaktadır. Eğer birinci noktadan sonra ikinci nokta izafi girilmek istenirse, birinci noktanın üzerine izafi değerler metin kutusundan verilerek girilir ve birinci nokta ile ikinci nokta arasında doğru çizilir. Komut Visual Basic'te Şekil 2.3'de verildiği gibi kullanılmaktadır.



Line (X_1, Y_1) - (X_2, Y_2)

$X_1, Y_1 = 1$. Noktanın mutlak koordinatları.

$X_2, Y_2 = 2$. Noktanın mutlak koordinatları.

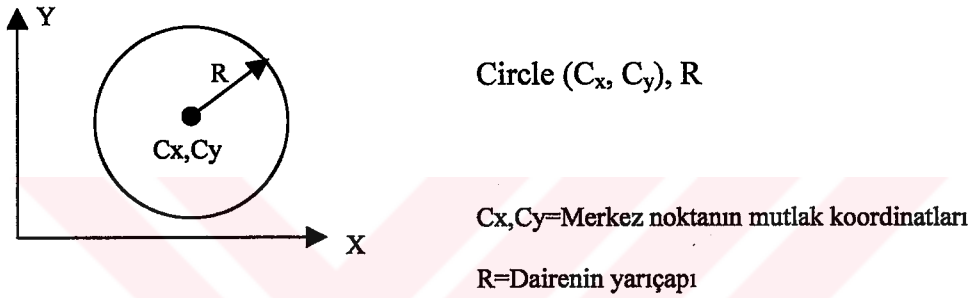
$X_i, Y_i = 2$. Noktanın 1. Noktaya göre

izafi koordinatları

Şekil 2.3. Doğru komutunun uygulanışı.

2.1.3.1.2. Daire Komutu (Circle)

Bu komut için merkez noktanın koordinatları ile yarıçapı vermek yeterlidir. Merkez ve yarıçap koordinatları fare ile ya da metin satırından klavye ile girilebilmektedir. Daire çizimi 2 veya 3 noktadan geçen dairenin çizimi olarak da gerçekleştirilebilmektedir. Komut Visual Basic'te Şekil 2.4'deki gibi uygulanmaktadır.



Şekil 2.4. Daire komutunun uygulanışı.

2.1.3.1.3. Dikdörtgen Komutu (Rectangle)

Bu komut Visual Basic'deki Doğru komutu kullanılarak çizilmektedir. Komutun sonuna B (box) harfi eklendiğinde dikdörtgen çizimi gerçekleştirilmektedir. Program, veri tabanına dikdörtgenin dört kenarını ayrı ayrı "doğru" olarak kaydetmektedir. Bu nedenle dikdörtgeni silmek için dört kenarı da seçip silmek gerekmektedir.

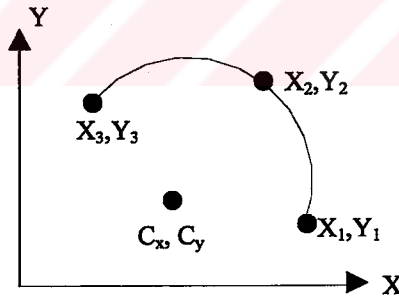
2.1.3.1.4. Yay Komutu (Arc)

Bu komut Daire komutu kullanılarak çizilmektedir. Komutun Visual Basic'te uygulanabilmesi için merkez noktasının koordinatları (C_x, C_y), yarıçap (R), başlangıç ve bitiş noktalarının x eksenini ile yaptığı açılarının (α, β) bilinmesi gerekir. Komut Visual Basic'te şu şekilde uygulanmaktadır.

Circle (C_x, C_y), R , α , β

Yay iki şekilde çizilebilmektedir.

a) Üç noktadan geçen yay: Burada 3 noktadan geçen yayı çizerken, yayın merkez noktasını ve yarıçapını bulabilmek için 3 bilinmeyenli 3 denklemini çözmek gerekmektedir. Denklem çözümüyle yayın merkez noktası (C_x, C_y) ve yarıçapı elde edilmektedir. Ardından 1. ve 3. noktalar yayın başlangıç ve bitiş noktaları kabul edilerek yayın başlangıç ve bitiş açıları hesaplanmaktadır (Şekil 2.5).



Şekil 2.5. Üç noktadan geçen yay komutunun uygulanışı.

Daha sonra bu değerler Visual Basic'teki daire komutuna uygulanarak yay çizilmektedir. Yay, daire komutunda başlangıç ve bitiş açıları verilerek çizilir (α, β).

Bu açıların hesaplanması için yayın merkez noktasının bilmesi gerekmektedir. Bu amaçla kullanılan denklemler şöyledir (2.1-2.14).

$$(X_1 - C_x)^2 + (Y_1 - C_y)^2 = R^2 \quad (2.1)$$

$$(X_2 - C_x)^2 + (Y_2 - C_y)^2 = R^2 \quad (2.2)$$

$$(X_3 - C_x)^2 + (Y_3 - C_y)^2 = R^2 \quad (2.3)$$

$$X_1^2 - 2X_1C_x + C_x^2 + Y_1^2 - 2Y_1C_y + C_y^2 = R^2 \quad (2.4)$$

$$X_2^2 - 2X_2C_x + C_x^2 + Y_2^2 - 2Y_2C_y + C_y^2 = R^2 \quad (2.5)$$

$$X_3^2 - 2X_3C_x + C_x^2 + Y_3^2 - 2Y_3C_y + C_y^2 = R^2 \quad (2.6)$$

$$(X_2^2 - X_1^2 + Y_2^2 - Y_1^2) - 2C_x(X_2 - X_1) - 2C_y(Y_2 - Y_1) = 0 \quad (2.7)$$

$$(X_3^2 - X_2^2 + Y_3^2 - Y_2^2) - 2C_x(X_3 - X_2) - 2C_y(Y_3 - Y_2) = 0 \quad (2.8)$$

$$C_x = \frac{(X_2^2 - X_1^2 + Y_2^2 - Y_1^2)}{2(X_2 - X_1)} - C_y \frac{(Y_2 - Y_1)}{(X_2 - X_1)} \quad (2.9)$$

$$C_x = \frac{(X_3^2 - X_2^2 + Y_3^2 - Y_2^2)}{2(X_3 - X_2)} - C_y \frac{(Y_3 - Y_2)}{(X_3 - X_2)} \quad (2.10)$$

$$A = \frac{(X_2^2 - X_1^2 + Y_2^2 - Y_1^2)}{2(X_2 - X_1)} \quad B = \frac{(Y_2 - Y_1)}{(X_2 - X_1)} \quad (2.11)$$

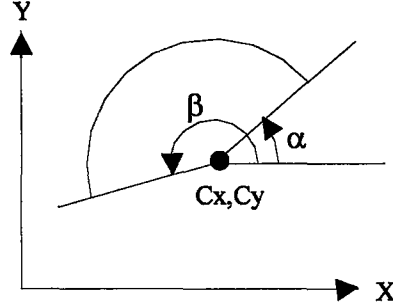
$$C = \frac{(X_3^2 - X_2^2 + Y_3^2 - Y_2^2)}{2(X_3 - X_2)} \quad D = \frac{(Y_3 - Y_2)}{(X_3 - X_2)} \quad (2.12)$$

$$C_x = A - C_y B \quad C_x = C - C_y D \quad (2.13)$$

$$C_y = \frac{C - A}{D - B} \quad C_x = A - \left(\frac{C - A}{D - B} \right) B \quad (2.14)$$

b) Merkezi, başlangıç ve son açısı bilinen yay: Bu komut için fare ile tıklayarak veya klavyeden merkez noktasının koordinatı (Cx,Cy) girilerek verilir. Açıları tesbit etmek için fare ile iki ayrı nokta daha girilir. Bu noktalar ile merkez

arasındaki çizginin x eksenine ile yaptığı açı bu yayın başlangıç ve son açısı olur (α , β açıları) (Şekil 2.6).



Şekil 2.6. Merkezi, başlangıç ve son açısı bilinen yay komutunun uygulanışı.

2.1.3.1.5. Çokgen Komutu (Polygon)

Çokgen çiziminde daire istenilen sayıda eşit açılara bölünerek yapılmaktadır. Burada önce bölme işleminin başlayacağı noktanın tesbit edilmesi gerekmektedir. Bunun için fare ile merkez ve yarıçap noktaları tesbit edilirken, basılan noktaların x eksenine ile yaptığı açı hesaplanır. Ardından diğer dilimlerin açıları bunun üzerine artışı olarak eklenir. Daha sonra bu açılara ve yarıçapa bağlı olarak her çevredeki kenar çizgisinin başlangıç ve bitiş koordinatları hesaplanır. Bu koordinatlar kullanılarak doğru çizimi yapılır ve poligonun gösterimi tamamlanmış olur. Poligon yarıçapı verilen hayali dairenin içine çizilmektedir.

2.1.3.2. Düzeltme Komutları (Edit)

2.1.3.2.1. Taşıma Komutu (Move)

Taşınacak cisimler seçildikten sonra bunların koordinatları istenen iki nokta arası kadar artırılmakta veya azaltılmaktadır. Taşınan noktanın koordinatları aşağıdaki matrisle hesaplanmaktadır (2.15).

$$T(\Delta x, \Delta y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \Delta x & \Delta y & 1 \end{bmatrix} \quad P'\{x' y' 1\} = P\{x y 1\}T(\Delta x, \Delta y) \quad (2.15)$$

2.1.3.2.2. Kopyalama Komutu (Copy)

Tıpkı taşıma komutunda olduğu gibi kopyalanacak şekil bir noktadan tutulup belli bir mesafe öteye taşınmaktadır. Bu taşıma esnasında orjinal şekil silinmemektedir.

2.1.3.2.3. Döndürme Komutu (Rotate)

Seçilen cisimler belirtilen bir nokta etrafında istenen açı kadar döndürülerek yeni koordinatları ile veri tabanına yazılmaktadır. Elemanın her noktası bir merkez etrafında döndürülürken kullanılan dönüşüm matrisi aşağıdaki gibidir (2.16). Döndürülecek her noktanın koordinatları bu matrisle çarpıldığında yeni noktalar elde edilir.

$$R(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P\{x' y' 1\} = P\{x y 1\}R(\alpha) \quad (2.16)$$

2.1.3.2.4. Aynalama Komutu (Mirror)

Aynalama yapılacak cisimler önce seçilmektedir. Daha sonra aynalama eksenini için iki nokta istenmektedir. Bu noktalar verildikten sonra veri tabanını yeniden okutup bu aynalama yapılmış elemanların koordinatları aşağıdaki matrisler vasıtası ile yeniden hesaplanıp tekrar veri tabanına aktarılmaktadır. Aynalama komutu 5 aşamada gerçekleştirilmektedir.

Adım 1: Aynalama eksenini orijinden geçecek şekilde, bütün noktaların koordinatları y eksenini doğrultusunda kaydırılarak değiştirilir (2.16). Move komutunda uygulanan prosedür bu aşamada kullanılır (Şekil 2.7.a.).

$y=mx+n$ Aynalama ekseninin denklemi.

$n=y-mx$ Aynalama ekseninin y eksenini kestiği nokta. Bütün noktalar bu değer ölçüsünde orijine taşınacaktır.

$$T(0,-n) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -n & 1 \end{bmatrix} \quad P_1\{x, y, 1\} = P\{x, y, 1\}T(0, -n) \quad (2.16)$$

Adım 2: Aynalama eksenini orijine taşıdıktan sonra eksen de dahil bütün noktalar negatif yönde orijin etrafında döndürülür (2.17). Döndürme komutunda kullanılan işlemler burada uygulanmış olur (Şekil 2.7.b).

$$R(-\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P_2\{x \ y \ 1\} = P_1\{x \ y \ 1\}T(0, -n) \quad (2.17)$$

Adım 3: Aynalama eksenini x eksenine ile çakıştıktan sonra, noktalar x eksenine etrafında “aynalama” yapılır (2.18). Aynalama matrisinin kullanıldığı esas yer burası olmuş olur (Şekil 2.7.c.).

$$M(x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P_3\{x \ y \ 1\} = P_2\{x \ y \ 1\}T(0, -n) \quad (2.18)$$

Adım 4: Daha sonra eksen ve bütün noktalar orijin etrafında Adım 2’deki yerine tekrar döndürülür. Burada Döndürme komutu pozitif açı yönünde uygulanır (Şekil 2.7.d).

$$R(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P_4\{x \ y \ 1\} = P_3\{x \ y \ 1\}T(0, -n) \quad (2.19)$$

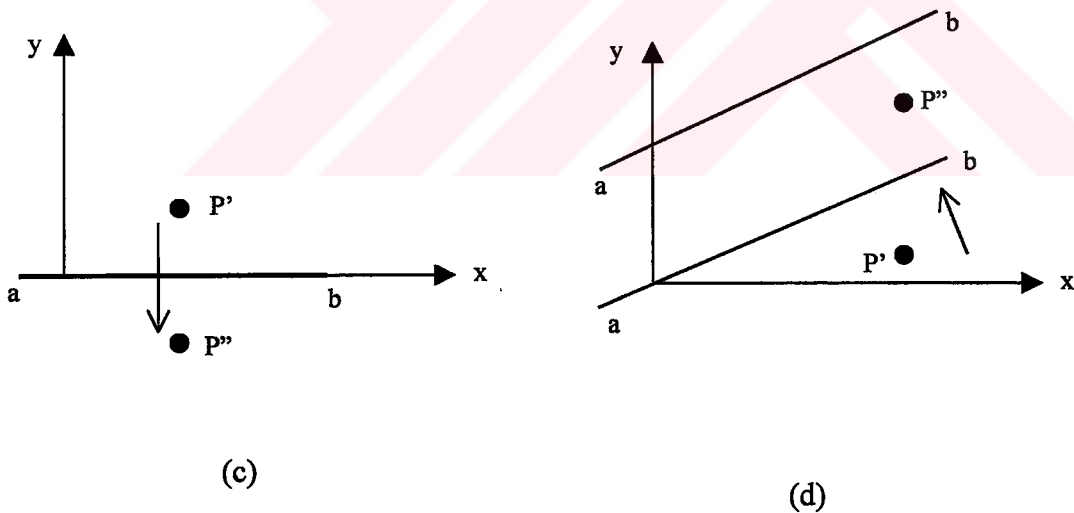
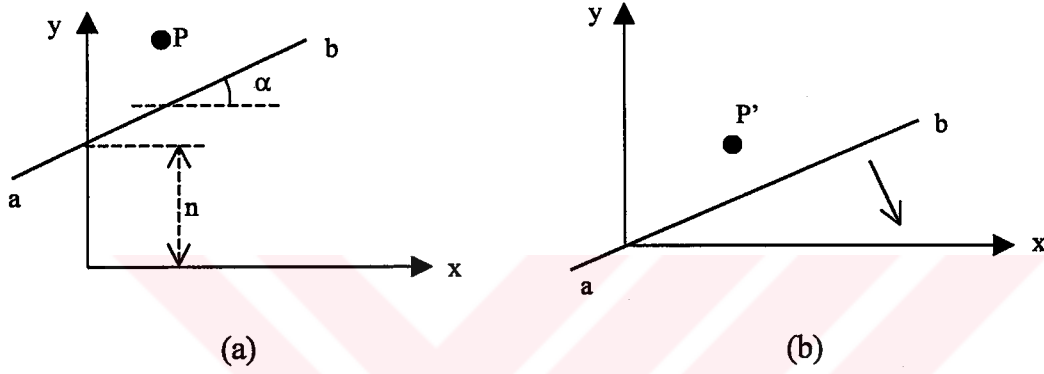
Adım 5: Son olarak aynalama eksenini ve cismin noktaları “n” kadar, y eksenine doğrultusunda ilk yerine taşınır (2.20).

$$T(0, n) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & n & 1 \end{bmatrix} \quad P_5\{x \ y \ 1\} = P_4\{x \ y \ 1\}T(0, n) \quad (2.20)$$

Bütün bu matrisler tek bir matris halinde toparlanırsa elde edilen aynalama

matrisi şöyle olur (2.21):

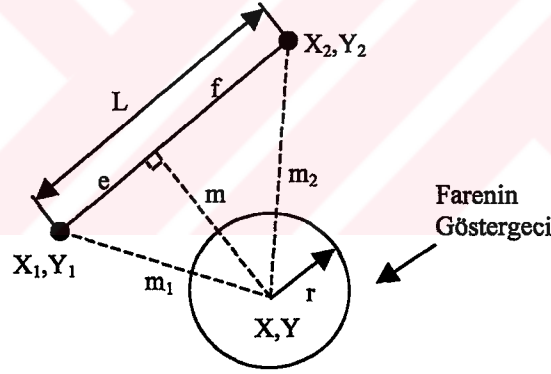
$$T=T(0,-n)R(-\alpha)M(x)R(\alpha)T(0,c)=\begin{bmatrix} \cos 2\alpha & \sin 2\alpha & 0 \\ \sin 2\alpha & -\cos 2\alpha & 0 \\ -n\sin 2\alpha & n(\cos 2\alpha + 1) & 1 \end{bmatrix} \quad (2.21)$$



Şekil 2.7. Aynalama komutunun uygulama aşamaları.

2.1.3.2.5. Silme Komutu (Erase)

Bu komut, fare ile basılan noktayı bulduktan sonra buna en yakın elemanı arařtırmaktadır. Bir elemanın basılan noktaya mesafesini ekran üzerindeki imlecin yarıçapı ile karşılařtırmaktadır. Eđer yarıçaptan küçükse bu elemanı silmek üzere kodlamaktadır ve kullanıcıya göstermektedir (Şekil 2.8). Daha sonra farenin sađ tuşuna basıldıđında veri tabanı yeniden oluşturulmakta ve silinmek üzere seçilen elemanlar veri tabanına yazılmamaktadır. Ayrıca komut, basma noktasının elemanlara uzaklıđının bulunması yanında, basılan noktanın bu elemanın ilk noktası ile son noktası arasında olup olmadıđını da arařtırmaktadır. Diđer komutlardaki eleman seçme işleminin de bu şekilde yapılmaktadır.



Şekil 2.8. Bir doğrunun fare ile seçilmesi işlemi.

Dođruya ait fonksiyonun katsayıları (2.21-2.24):

$$Ax+By+C=0 \quad (2.21)$$

$$A=(Y_2, Y_1) \quad (2.22)$$

$$B=(X_1, X_2) \quad (2.23)$$

$$C=(X_2 * X_1)/(X_1 * Y_2) \quad (2.24)$$

Doğrunun boyu (2.25):

$$L=\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (2.25)$$

Basılan noktanın doğruya olan mesafesi (2.26):

$$m = \frac{|A * x + B * y + C|}{\sqrt{A^2 + B^2}} \quad (2.26)$$

Basılan noktanın doğrunun ilk noktasına olan mesafesi (2.27):

$$m_1 = \sqrt{(X_1 - X)^2 + (Y_1 - Y)^2} \quad (2.27)$$

Basılan noktanın, doğrunun ikinci noktasına olan mesafesi (2.28):

$$m_2 = \sqrt{(X_2 - X)^2 + (Y_2 - Y)^2} \quad (2.28)$$

$$e = \sqrt{(m_1^2 - m^2)} \quad (2.29)$$

$$f = \sqrt{(m_2^2 - m^2)} \quad (2.30)$$

Bir doğrunun seçilip silinebilmesi için “r” yarıçaplı farenin göstergesinin doğrunun üzerine basması gerekir. Bunun içinde matematiksel olarak şu 3 şartı aynı anda sağlaması gerekir.

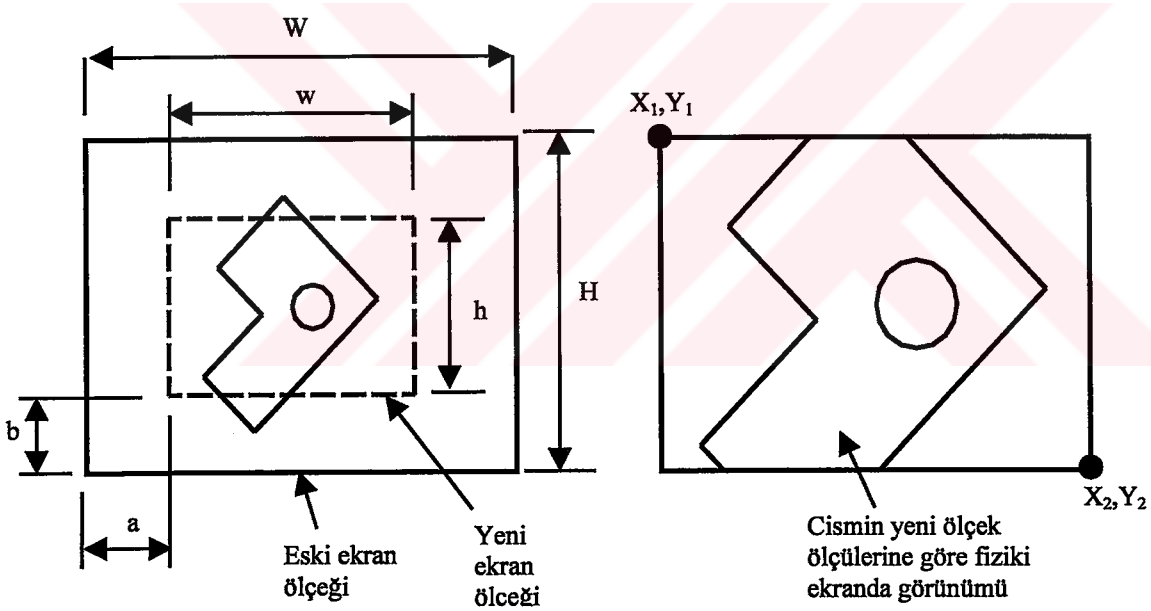
$$r < m, e \leq L, f \leq L$$

Burada “r” farenin göstergesinin yarıçapı çizim penceresinin belli bir oranındadır ve sabittir.

2.1.3.3. Görüntü Komutları (View)

2.1.3.3.1. Büyütme Komutu (Zoom Pozitif)

Bu komuta her basıpta ekran ölçeği yarıya düşürülerek işlemler yapılır. Ekran ölçeği azalınca normal boydaki çizgi büyümüş gibi görülür (Şekil 2.9). Ekran ölçeği için programda, matematiksel olarak ekranın iki köşesinin koordinatları verilir. Verilen bu koordinatlar ekranın fiziki boyutları ile orantılı olmak zorundadır. Yoksa çizgiler uzamış veya daralmış olarak gözükür. Yani daire elips gibi gözükmeye başlar.



Şekil 2.9. Büyütme komutunun uygulanışı.

Ekran ölçeğini değiştirmek için şu işlemler yapılır.

$$w = W / 2$$

W, H = Eski ölçeğe göre ekran boyu ve yüksekliği

$h = H / 2$ $w, h = \text{Yeni ölççeğe göre ekran boyu ve yükseklięi}$

$a = (W - w) / 2$

$b = (H - h) / 2$

Ekranın köşelerinin koordinat deęerleri

$\text{YeniEkranX1} = \text{EskiEkranX1} + a$

$\text{YeniEkranY1} = \text{EskiEkranY1} - b$

$\text{YeniEkranX2} = \text{EskiEkranX2} - a$

$\text{YeniEkranY2} = \text{EskiEkranY2} + h$

Visual Basicte, picture nesnesinde koordinatların uygulanışı.

`Picture.Scale (YeniEkranX1, YeniEkranY1)-(YeniEkranX2, YeniEkranY2)`

2.1.3.3.2. Küçültme Komutu (Zoom Negatif)

Bu komut büyüt komutunun uygulanışı ile aynıdır. Fakat burada yeni ekran ölççeęi büyütölmektedir. Dolayısı ile çizgiler, ölççeęi büyütölmüş ekranda küçölmüş olarak gözökecektir. Bunun için ekranın ölççeęi şu şekilde deęiştirilmektedir.

$w = W * 2$ $W, H = \text{Eski skalaya göre ekran boyu ve yükseklięi}$

$h = H * 2$ $w, h = \text{Yeni skalaya göre ekran boyu ve yükseklięi}$

$a = (w - W) / 2$

$b = (h - H) / 2$

Ekranın iki köşesinin yeni koordinatları.

$\text{YeniEkranX1} = \text{EskiEkranX1} - a$

$\text{YeniEkranY1} = \text{EskiEkranY1} + b$

$\text{YeniEkranX2} = \text{EskiEkranX2} + a$

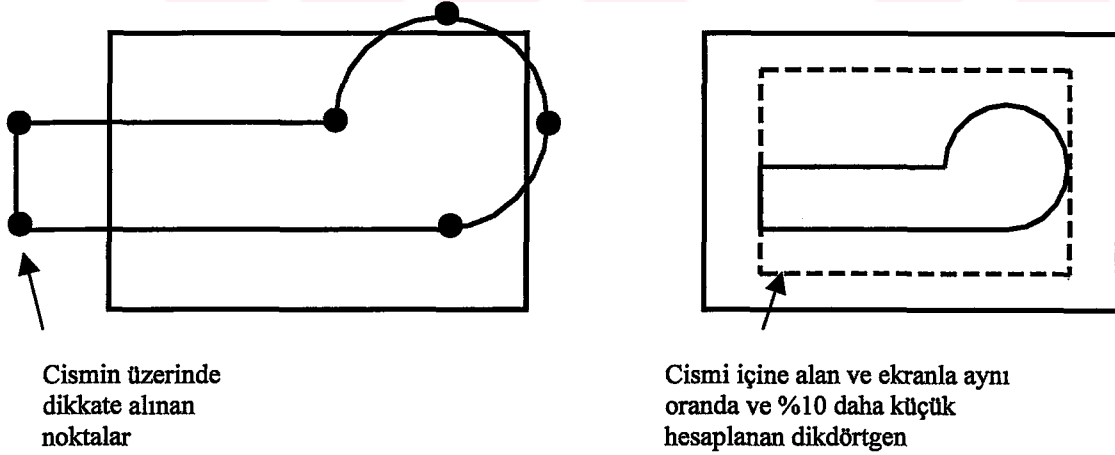
$$\text{YeniEkranY2} = \text{EskiEkranY2} - h$$

Ekranın hesaplanan yeni ölçeği.

Picture.Scale (YeniEkranX1, YeniEkranY1)-(YeniEkranX2, YeniEkranY2)

2.1.3.3.3. Hepsini Göster Komutu (Zoom All)

Bunun için veri tabanındaki bütün çizgilerin başlangıç ve bitiş noktaları araştırılmakta, daire ve yayların çeyrek noktaları tespit edilmektedir. Bütün bu noktalar bir dikdörtgen içinde varsayılmakta ve bu dikdörtgenin kenarlarının oranı ekranın kenarlarının oranı ile aynı olması sağlanmaktadır. Dikdörtgenin bütün kenarları ekranın içine girecek şekilde ekran ölçeği büyütülmekte veya küçültülmektedir. Ölçekteki bu değişiklik cisimlerin kenarında bir miktar daha boşluk kalması için %10 daha fazla alınmaktadır. Komutun uygulanışı Şekil 2.10'da gösterilmiştir.

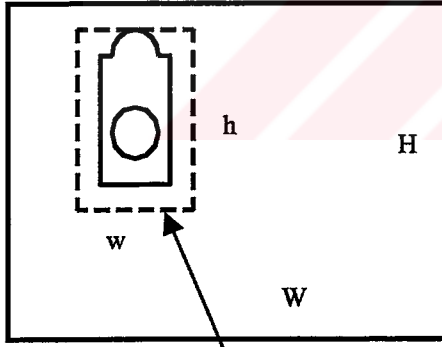


Şekil 2.10. Hepsini Göster komutunun uygulanışı.

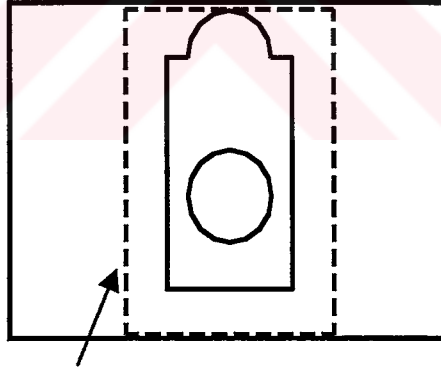
2.1.3.3.4. Pencereyi Göster Komutu (Zoom Window)

Pencere içine alınan cismi büyütmek için fare ile pencereyi belirlemek üzere iki nokta tesbit edilir ve dikdörtgen şeklindeki pencerenin boyutları çıkarılır. Daha sonra pencerenin boyu ve yüksekliği, ekranın boyu ve yüksekliği ile orantı hale getirilir. Bu oranlardan hangisi küçükse yani hangi kenarın boyu ekraninkine yakınsa o kenar ekranın kenarı ile aynı olacak şekilde büyütülür. Pencerenin kenarları arasındaki oran bozulmayacak şekilde diğer kenarda büyütülür ve ekranı tam ortalayacak şekilde yerleştirilir. Burada ekrana tam olarak yerleşen kenarın büyütme oranı aynı zamanda cismin büyütme oranı olur (Şekil 2.11). Bu şekil için büyütme oranı H/h olacaktır.

$$\text{Büyütmeoranı} = \frac{H}{h} < \frac{W}{w}$$



Cismi içine alan Zoom Window penceresi



Yüksekliklerin oranı daha küçük olduğu için pencerenin yüksekliği ekrana tam oturacak şekilde büyütülür. Bu büyütme oranı aynı zamanda cismin büyütme oranıdır.

Şekil 2.11. Pencereyi Göster komutunun uygulanışı.

2.1.3.3.5. Önceki Ekran Komutu (Zoom Previous)

En son gösterilen 10 ekran ölçeği bir değışkende tutularak saklanır. Bu komut her çalıştırıldığında daha önceden hafızaya alınan bu ölçekler, ekranı ölçekleyen komuta yüklenerek, önceki gösterilen ekranların tekrar gösterilmesi sağlanır.

2.1.3.3.6. Ekran Kaydırma Komutu (Pan)

İlk bakışta bu komut ile “Taşı” komutu görsel açıdan bir birine benziyormuş gibi görünmesine rağmen aralarında çok büyük fark vardır. Taşı komutunda cisim bir noktadan başka bir noktaya taşınmasında cismin bütün noktalarının koordinatları değıştirilir. Dolayısıyla birkaç cismin içerisinde bir tanesi taşındığında diğerleri yerinde kalır. Ekran kaydırma komutunda ise değışiklik sadece ekranın ölçeğinde yapılır. Dolayısıyla bu komut uygulandığında bütün cisimler bir noktadan başka bir noktaya taşınmış gibi görülür (Şekil 2.12).

Ekran kaydırma komutu için iki nokta gereklidir. Bu iki nokta arasındaki mesafe hesaplanarak ekranın ölçeği ona göre değıştirilmektedir.

Eksenlere göre ekranın kayacağı miktar.

$$X=X_2-X_1$$

$$Y=Y_2-Y_1$$

Kaymış olan ekranın köşe noktasının koordinatları

$$\text{YeniEkranX1} = \text{EskiEkranX1} - X$$

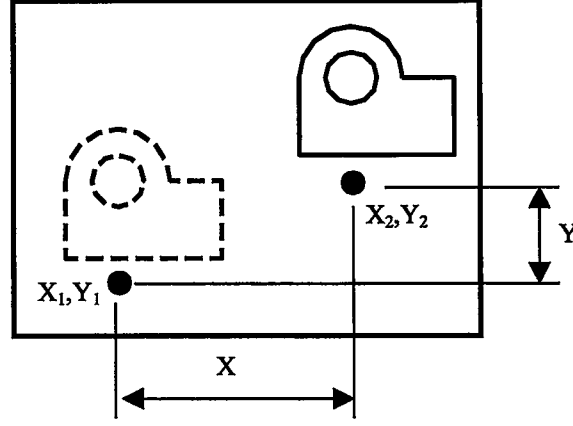
$$\text{YeniEkranY1} = \text{EskiEkranY1} - Y$$

$$\text{YeniEkranX2} = \text{EskiEkranX2} - X$$

$$\text{YeniEkranY2} = \text{EskiEkranY2} - Y$$

Yeni koordinatlara göre hesaplanan ekranın ölçeği.

Picture.Scale (YeniEkranX1, YeniEkranY1)-(YeniEkranX2, YeniEkranY2)



Şekil 2.12. Ekran kaydırma komutunun uygulanışı.

2.1.3.3.7. Yeniden Çiz Komutu (Redraw)

Bu işlem için veri tabanındaki bütün bilgiler yeniden okutulup tekrar çizilmektedir. Aynı zamanda ızgara (grid) gösterimi açık ise ızgara da yenilenmektedir.

2.2. Çizimin Algoritmaya Hazırlanması

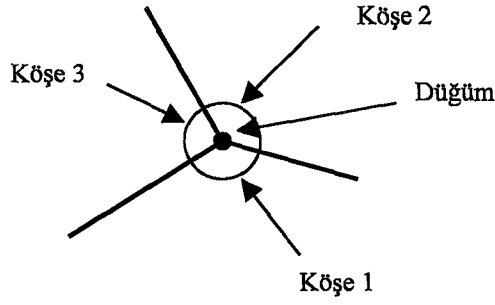
Parçanın görünüşleri 2-B'lu ortamda çizilip veri tabanına aktarıldıktan sonra, bu bilgilerin algoritma içerisinde kullanılabilir hale dönüştürülmesi gerekir. Çizimdeki elemanların kesişim noktalarından parçalara ayrılması, çizgilerin görünüşlere taksimi, elemanların özelliklerinin tespiti, elemanlar arasındaki bağıntıların tespiti gibi konular bu aşamada yapılan işlerdir.

2.2.1. Temel Kavramlar

Tezin içerisinde kullanılan bazı önemli kavramların açıklamaları aşağıda sunulmuştur. Bu kavramların bir kısmı uluslararası standartlarla aynı olarak kullanılmış, bazıları ise tez içerisindeki kullanımı dikkate alınarak açıklanmıştır.

Düğüm (Vertex): 2B'lu uzayda en az iki, 3B'lu uzayda ise en az üç kenar tarafından paylaşılan ortak noktaya denir⁽¹¹⁾. Bu kelime Türkçe'de "köşe" olarak da kullanılmaktadır. Fakat algoritma içerisinde "köşe" kavramı birazda özel bir anlam yüklenerek kullanılmıştır.

Köşe (Corner): 2B'lu uzayda yalnız iki kenarın paylaştığı ortak noktaya bu ad verilmiştir. Buna göre her düğüm kendi içerisinde bir kaç köşe bulundurabilir. Köşeler her alanın kenarları üzerinde döngü oluştururken, o alanın üzerinde oluşan köşelerin özellikleri ve açılarını saklamada kullanılan bir kavram olarak kullanılmıştır (Şekil 2.13).



Şekil 2.13. Algoritmada kullanıldığı şekliyle köşe ve düğüm kavramlarının gösterimi.

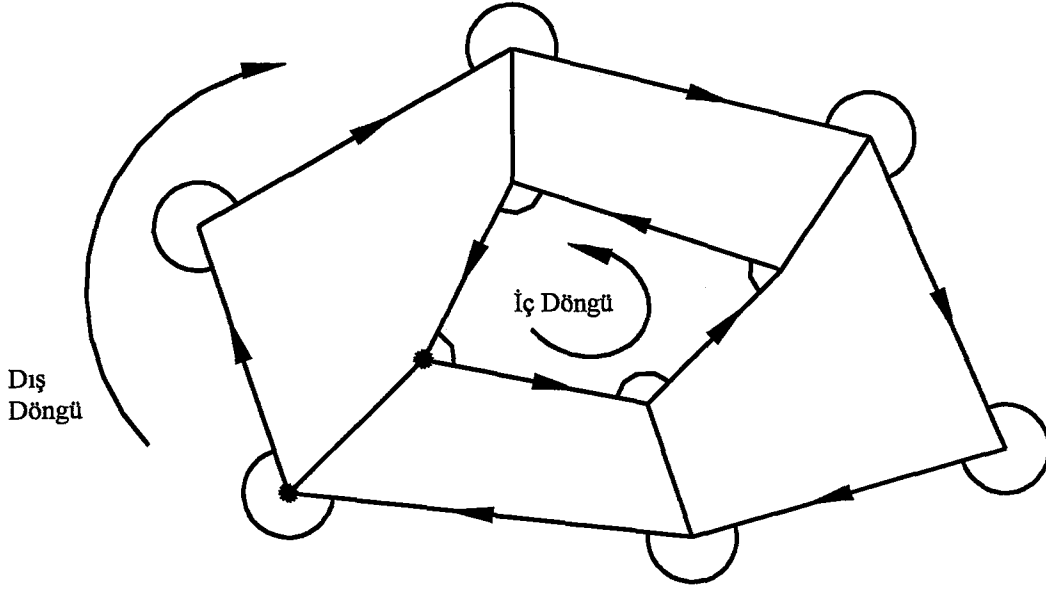
Kenar (Edge): 2B'lu uzayda alanların, 3B'lu uzayda ise yüzeylerin sınırını oluşturan ve iki alan veya yüzey tarafından paylaşılan ve uçlarında düğümlerin bulunduğu doğru ya da eğriye denir⁽¹¹⁾.

Yüzey (Face): 3B'lu uzayda cismin sınırını belirleyen, içi boş olmayan, kenar ve düğümlerden oluşan çerçevelere denir⁽¹¹⁾.

Cisim (Object): 3B'lu uzayda sonlu sayıda yüzey tarafından kapalı, içi dolu, kenar ve düğümlerden meydana gelen yapıya denir⁽¹¹⁾.

İç Döngü: Bir çerçevenin kenarları üzerinde dolaşırken, soldan itibaren süpürülen açılarının toplamı, o çerçevedeki kenarların sayısı ile 360 derecenin çarpımının yarısında daha az ise bu döngü “iç döngü” olur (Şekil 2.14).

Dış Döngü: Bir çerçevenin kenarları üzerinde dolaşırken, soldan itibaren süpürülen açılarının toplamı, o çerçevedeki kenarların sayısı ile 360 derecenin çarpımının yarısında daha fazla ise bu döngü “dış döngü” olur (Şekil 2.14).



Şekil 2.14 Algoritmada kullanıldığı şekliyle iç ve dış döngülerin gösterimi.

Sınır Kutu: 3B’lu olarak cismi içine alan en küçük kutuya denir ve cismin işleneceği ham malzemeyi temsil eder. Çizimde bölüm çizgisiyle (-----) gösterilir.

Sınır Çerçeve: Görünüşlerde cismi içine alan en küçük dikdörtgene denir. Aslında bu da sınır kutunun bir görünüşüdür ve çizgi tipi olarak yine bölüm çizgisi kullanılır.

Dış Çerçeve: Görünüşlerde parçanın en dıştaki çizgilerinin oluşturduğu çerçeveye bu ad verilir. Bütün kenarlar sürekli (solid) çizgi (———) olmak zorunda. Kenar taraması da sadece dış döngüye sahiptir.

Temel Görünüş: Algoritmayı uygulamaya başlamak için öncelikle bir görünüşten başlamak gerekmektedir. Bu görünüş “Temel Görünüş” olarak

adlandırılacaktır. Temel görünüş cismin üzerindeki yüzeylerin en iyi şekilde gösterildiği görünüştür. Pratikte parça işlenirken, tezgahın tablasına yukarıdan baktığımız görünüştür. Temel görünüş otomatik, elle ya da baştan kabul olmak üzere 3 şekilde seçilebilir.

Hayali Kenar (Silhouette-Edge): Eğri bir yüzeyin bakış yönüne teğet olması durumunda görünüşte oluşan kenarlardır. Bir silindirin yanal kenarlarının görünüşlerde oluşturduğu çizgi bu şekildedir.

Katı Modelleme: Cisimleri 3B'lu uzayda tam olarak ifade eden modelleme şekline denir. Üç çeşit katı model tekniği kullanılır ^(13, 143).

Yapısal Katı Geometri (CSG): Bu teknikte cismi oluşturan kısımlar basit hacimler (primitive) şeklinde düzenlenir ve bu hacimler cismi tanımlamak için birbirlerine eklenir ya da çıkarılır.

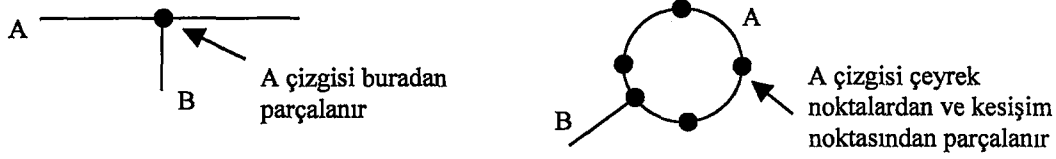
Sınır Temsili Modelleme (B-rep): Bu gösterimde cisim yüzeylerden oluşur. Her yüzey kenarlarla, kenarlar ise düğümlerle birbirine bağlanır.

Süpürme Modelleme: Ayrıca, 2B'lu bir alanın bir doğru boyunca ya da dairesel yol boyunca süpürülmesiyle de katı model tanımlanabilir.

2.2.2. Çizgilerin Parçalara Ayrılması

Görünüşlerdeki bütün çizgileri kesişim noktalarından parçalara ayırmak gerekmektedir. Yay ve dairelerin çeyrek noktalarında bir düğüm bulunmasa bile bu çeyrek noktaların izdüşümü, diğer görünüşlerde silüet çizgilerin oluşmasına neden olmaktadır (görünüşlerde silindirin yanal çizgileri gibi). Bunun gibi silüet noktaların projeksiyonlarını yakalayabilmek için daire ve yayların da çeyrek noktalardan

parçalara ayrılması zorunlu olmaktadır (Şekil 2.15). Çizgileri parçalama algoritmasının genel yapısı Şekil 2.16’da verilmiştir.



Şekil 2.15. Çizgilerin parçalara ayrıldığı noktaların gösterimi.

2B’lu çizimleri oluşturmak için Doğru, Daire, Yay olarak üç eleman kullanıldığı için ortaya çıkan kesişim olasılıkları da bunların etkileşimleri olur. Bunun için 6 adet kesişme türü ortaya çıkar. Bunlar;

- Doğru-Doğru kesişimi,
- Doğru-Daire kesişimi,
- Doğru-Yay kesişimi,
- Daire-Daire kesişimi,
- Yay-Daire kesişimi,
- Yay-Yay kesişimidir.

2.2.2.1. Doğru-Doğru Kesişimi

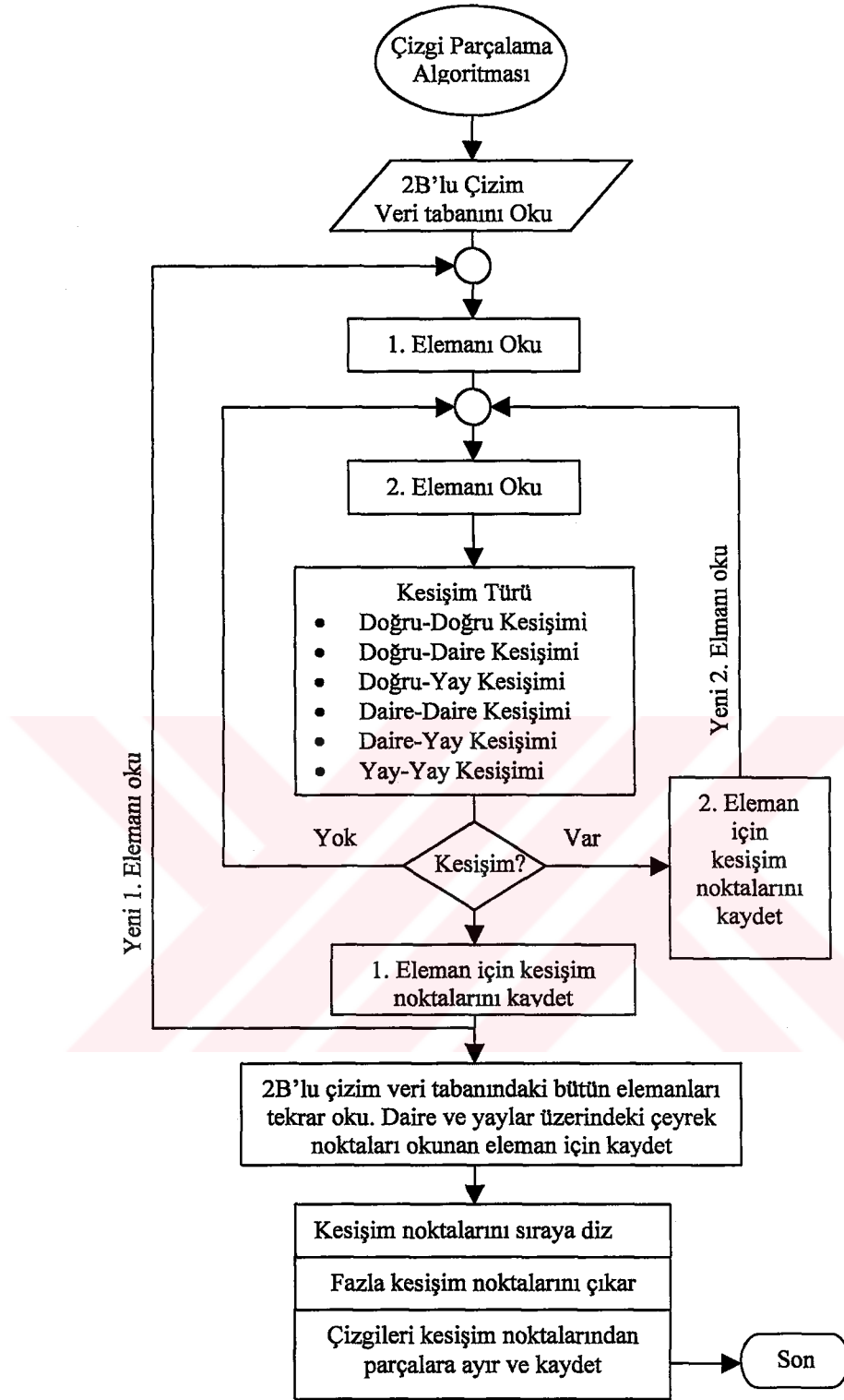
Bu elemanlar kesişirken çizgiler üç şekilde konumlanabilir. Çapraz, yatay ve dikey. Kesişim noktası ise çizginin ya uçları arasında ya da uç noktasında bulunabilir. Kesişim noktasını bulmak ve çizginin üzerinde olup olmadığını kontrol etmek için bütün bu değişimleri kontrol etmek gerekir (Şekil 2.17).

Kesişen doğruların analitik denklemleri ve katsayıları (2.31-2.34);

$$A_1X+B_1Y+C_1=0$$

$$A_2X+B_2Y+C_2=0$$

(2.31)



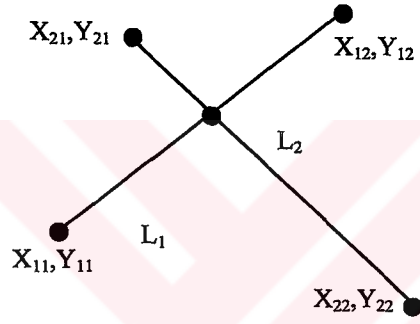
Şekil 2.16. Çizgileri parçalama algoritmasının genel yapısı.

$$A_1=Y_{12}-Y_{11} \quad A_2=Y_{22}-Y_{21} \quad (2.32)$$

$$B_1=X_{11}-X_{12} \quad B_2=X_{21}-X_{22} \quad (2.33)$$

$$C_1=X_{12} * Y_{11}-X_{11}-Y_{12} \quad C_2=X_{22} * Y_{21} - X_{21} * Y_{22} \quad (2.34)$$

Şeklindedir.



Şekil 2.17. İki doğrunun kesişimi

Çizgilerin duruş şekline göre çeşitli kesişme varyasyonları ortaya çıkar.

2.2.2.1.1. Doğru1 ve Doğru2 Çapraz

Bu kesişimin gerçekleşmesi için doğruların denklemlerinin katsayıları $A_1 \neq 0$, $B_1 \neq 0$, $A_2 \neq 0$, $B_2 \neq 0$ olmalıdır. Bu durumda her iki çizgi de kesişim noktasından parçalara ayrılır. Çizgilerin analitik denklemleri ortak olarak çözülmüşse kesişim noktasının koordinatları aşağıdaki 2.35 nolu eşitlikle bulunur.

$$X = \frac{(B_1C_2 - B_2C_1)}{(A_1B_2 - A_2B_1)} \quad Y = \frac{(-A_1X - C_1)}{B_1} \quad (2.35)$$

Bu kesişim noktasının çizgilerin üzerinde bulunması gerekir. Bunu sağlamak içinde kesişim noktasının koordinatı uç noktaların arasında kalmalıdır. Bunun için şu şartlar sağlanmalıdır.

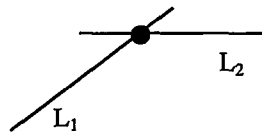
- $X_{11} < X < X_{12}$ veya $X_{12} < X < X_{11}$ ve $Y_{11} < Y < Y_{12}$ veya $Y_{12} < Y < Y_{11}$
- $X_{21} < X < X_{22}$ veya $X_{22} < X < X_{21}$ ve $Y_{21} < Y < Y_{22}$ veya $Y_{22} < Y < Y_{21}$

2.2.2.1.2. Doğru1 Çapraz, Doğru2 Yatay

Bu durumda $A_1 \neq 0$, $B_1 \neq 0$, $A_2 = 0$, $B_2 \neq 0$ dır (Şekil 2.18). Ortak kesişim noktasının koordinatları, eşitlik 2.36'daki gibi olur.

$$Y = \frac{C_2}{B_2} \quad X = \frac{-(B_1Y - C_1)}{A_1}$$

(2.36)



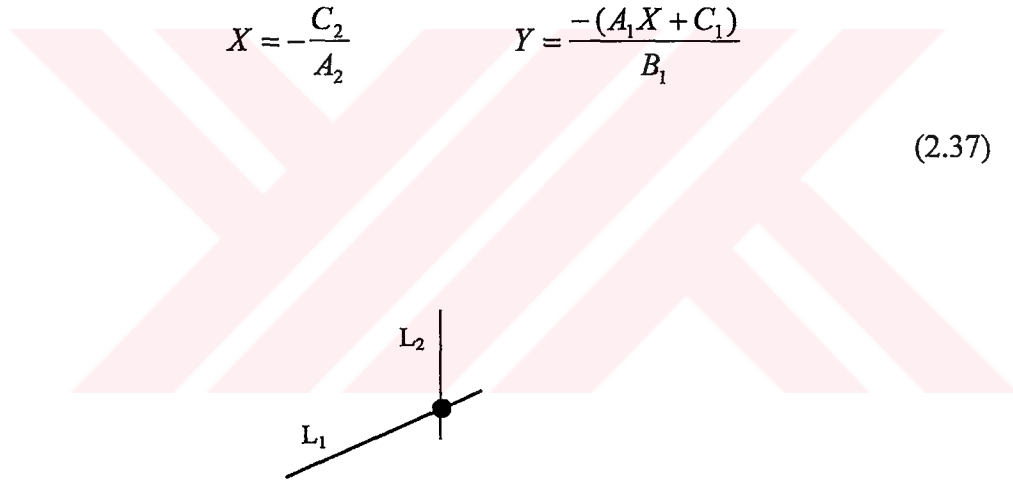
Şekil 2.18. Doğru1'in çapraz, Doğru2'nin yatay bulunduğu konumda kesişimi.

Bu kesişim noktasının çizgilerin üzerinde bulunması için şu şartların sağlanması gerekir.

- $X_{11} < X < X_{12}$ veya $X_{12} < X < X_{11}$ ve $Y_{11} < Y < Y_{12}$ veya $Y_{12} < Y < Y_{11}$
- $X_{21} < X < X_{22}$ veya $X_{22} < X < X_{21}$ ve $Y_{21} < Y$ veya $Y_{22} < Y$

2.2.2.1.3. Doğru1 Çapraz Doğru2 Dikey

Bu durumun gerçekleşmesi için $A_1 \neq 0$, $B_1 \neq 0$, $A_2 \neq 0$, $B_2 = 0$ olması gerekir (Şekil 2.19). Bu şekildeki bir kesişimde ortak noktanın koordinatları aşağıdaki 2.37 nolu eşitlikle bulunur.

$$X = -\frac{C_2}{A_2} \quad Y = \frac{-(A_1 X + C_1)}{B_1} \quad (2.37)$$


Şekil 2.19. Doğru1'in çapraz, Doğru2'nin dikey bulunduğu konumda kesişimi.

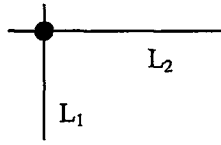
Kesişim noktasının çizgilerin üzerinde bulunması için şu şartların sağlanması gerekir.

- $X_{11} < X < X_{12}$ veya $X_{12} < X < X_{11}$ ve $Y_{11} < Y < Y_{12}$ veya $Y_{12} < Y < Y_{11}$
- $X_{21} = X$ veya $X_{22} = X$ ve $Y_{21} < Y < Y_{22}$ veya $Y_{22} < Y < Y_{21}$

2.2.2.1.4. Doğru1 Dikey, Doğru2 Yatay

Bu durumda denklemlerin katsayıları $A_1 \neq 0$, $B_1 = 0$, $A_2 = 0$, $B_2 \neq 0$ olur (Şekil 2.20). Kesişim noktasının koordinatları ise aşağıdaki 2.38 nolu eşitlikle bulunur.

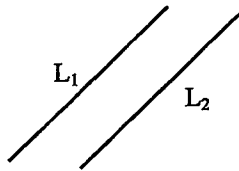
$$X = -\frac{C_1}{A_1} \quad Y = -\frac{C_2}{B_2} \quad (2.38)$$



Şekil 2.20. Doğru1'in dikey, Doğru2'nin yatay bulunduğu konumda kesişimi.

2.2.2.1.5. Doğru1 ve Doğru2'nin Paralel Olma Durumu

İki çizginin paralel olabilmesi için çizgilerin denklemlerinin katsayıları 2.39 nolu eşitliği sağlamalıdır (Şekil 2.21).



$$\frac{A_1}{B_1} = \frac{A_2}{B_2} \quad (2.39)$$

Şekil 2.21. Doğru1 ve doğru2'nin paralel olma durumu.

2.2.2.2. Doğru-Daire Kesişimi

Bu kesişim türünde doğru daireyle tek noktadan (teğet), iki noktadan (dairenin içine girerek) ve doğru dairenin içine girip diğer kenardan çıkmayacak şekilde tek noktadan kesişebilir.

Doğrunun denklemi ve katsayıları (2.40-2.43);

$$AX+BY+C=0 \quad (2.40)$$

$$A=Y_2-Y_1 \quad (2.41)$$

$$B=X_1-X_2 \quad (2.42)$$

$$C=X_2*Y_1 - X_1*Y_2 \quad (2.43)$$

Şeklindedir.

Dairenin denklemi (2.44);

$$(X-a)^2+(Y-b)^2=r^2 \quad (2.44)$$

şeklinde olur. “a” ve “b” dairenin merkez noktasının koordinatlarıdır.

Bu iki denklemin ortak çözümünden aşağıdaki 2.45 nolu eşitlikteki 2. derece denklem elde edilir.

$$X^2\left(1+\frac{A^2}{B^2}\right)+X\left(-2a+\frac{2AC}{B^2}+\frac{2Ab}{B}\right)+\left(a^2+b^2+\frac{C^2}{B^2}+\frac{2bC}{B}-r^2\right)=0 \quad (2.45)$$

Denklem daha sade yazılırsa katsayıları eşitlik 4.16-4.19’da verildiği şekilde olacaktır.

$$(A'X^2+B'X+C'=0). \quad (2.46)$$

$$A' = \left(1 + \frac{A^2}{B^2} \right) \quad (2.47)$$

$$B' = \left(-2a + \frac{2AC}{B^2} + \frac{2Ab}{B} \right) \quad (2.48)$$

$$C' = \left(a^2 + b^2 + \frac{C^2}{B^2} + \frac{2bC}{B} - r^2 \right) \quad (2.49)$$

Eşitlik 2.50'de verilen Δ 'nın durumuna göre denklemin kökleri tespit edilir.

$$\Delta = B'^2 - 4A'C' \quad (2.50)$$

Bu durumda Δ , 3 şekilde olabilir.

- a) $\Delta > 0$ ise denklemin 2 kökü vardır. Doğru ile daire iki noktadan kesişirler.
- b) $\Delta = 0$ ise denklemin 1 kökü vardır. Yani doğru ile daire tek noktadan kesişirler.
- c) $\Delta < 0$ ise denklemin kökleri yoktur. Doğru ile daire kesişmezler.

2.2.2.2.1. Doğru ile Dairenin İki Noktadan Kesişme Durumu ($\Delta > 0$)

Bu durumda kesişim noktalarının koordinatları, doğrunun duruş şekline göre 3 şekilde ortaya çıkar.

i) $A \neq 0$ ve $B \neq 0$: Doğrunun, daire eksenleri ile açı oluşturduğu bu durumda, kesişim noktalarının koordinatları aşağıdaki 2.51-2.52 eşitliklerinden hesaplanır.

$$X_1 = \frac{-B' - \sqrt{\Delta}}{2A'} \quad Y_1 = \frac{-(AX_1 + C)}{B} \quad (2.51)$$

$$X_2 = \frac{-B' + \sqrt{\Delta}}{2A'} \quad Y_2 = \frac{-(AX_2 + C)}{B} \quad (2.52)$$

ii) $A=0$ ve $B \neq 0$: Doğrunun yatay konumunda, kesişim noktalarının koordinatları 2.53 eşitliğiyle hesaplanır.

$$Y = -\frac{C}{B} \quad X_{1,2} = \frac{-B' \pm \sqrt{\Delta}}{2A'} \quad (2.53)$$

iii) $A \neq 0$ ve $B=0$: Doğrunun dikey konumunda kesişim noktalarının koordinatları 2.54 eşitliğiyle hesaplanır.

$$X = -\frac{C}{A} \quad Y_{1,2} = \frac{-B' \pm \sqrt{\Delta}}{2A'} \quad (2.54)$$

2.2.2.2.2. Doğru ile Dairenin Bir Noktadan Kesişme Durumu ($\Delta=0$)

Doğrunun duruşuna göre 3 değişik seçenek ortaya çıkar.

i) $A \neq 0$ ve $B \neq 0$: Doğrunun daire eksenleri ile açı oluşturduğu bu durumda, kesişim noktalarının koordinatları 2.55 eşitliğiyle hesaplanır.

$$X = \frac{-B'}{2A'} \quad Y = \frac{-(AX_1 + C)}{B} \quad (2.55)$$

ii) $A=0$ ve $B \neq 0$: Doğrunun yatay konumunda kesişim noktalarının koordinatları 2.56 eşitliğiyle hesaplanır.

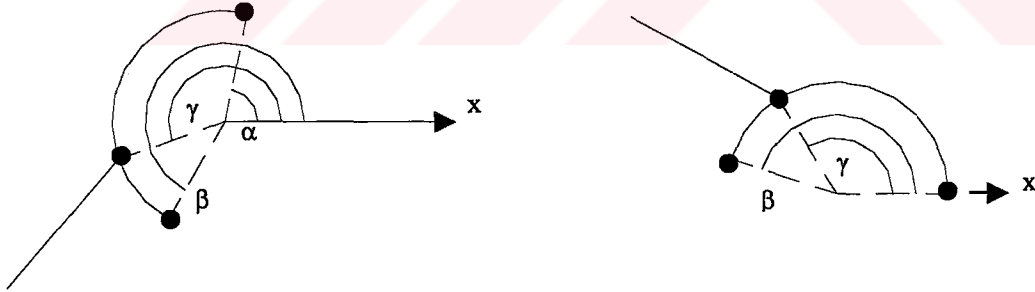
$$Y = -\frac{C}{B} \quad X = \frac{-B'}{2A'} \quad (2.56)$$

iii) $A \neq 0$ ve $B=0$: Doğrunun dikey konumunda kesişim noktalarının koordinatları 2.57 eşitliğiyle hesaplanır.

$$X = -\frac{C}{A} \quad Y = \frac{-B'}{2A'} \quad (2.57)$$

2.2.2.3. Doğru-Yay Kesişimi

Bu kesişim, doğrunun daireyle olan kesişimine benzer. Fakat burada yayın uç noktalarının doğru ile kesişip kesişmediğini de kontrol etmek gerekmektedir. Bu kontrolü yapabilmek için hem yayın uç noktalarının “x” eksenine yaptığı açı, hem de kesişim noktasının “x” eksenine yaptığı açı hesaplanır. Bütün bu üç açı ve yayın ilk noktasının açısı (α) “x” eksenine “0” olacak şekilde döndürülür. Ardından kesişim noktasının “x” eksenine yaptığı açı α ve β açılarının arasında kalıp kalmadığına bakılır. Eğer kalıyorsa bu kesişim noktası yay üzerinde demektir (Şekil 2.22). Bu çevirme işleminin yapılmasının nedeni; bazen yayın 2. Ucu 360° fazla dönmüş olup 1. bölgeye geçmiş olabiliyor. Bu durumda kesişim noktasının yerini tespit etmek imkansızlaşmaktadır. Bu nedenle bütün açılar ($\alpha=0$) olacak şekilde döndürülmektedir.



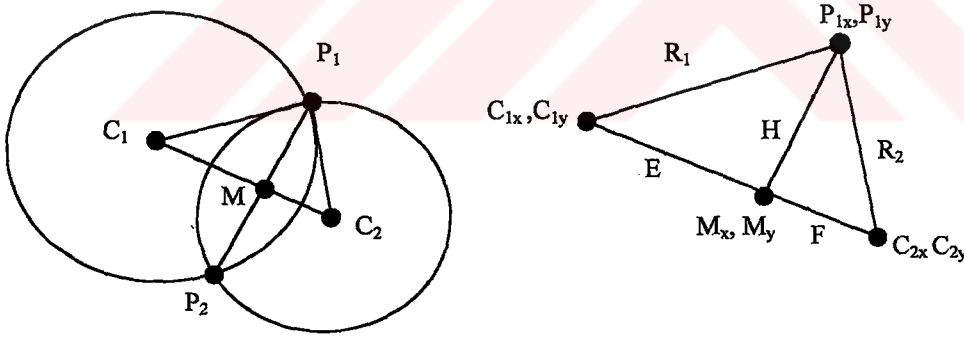
Şekil 2.22. Kesişim noktasının, yayın uçları arasında olup olmadığının tespit edilmesi (α =ilk noktanın açısı, β =son noktanın açısı, γ =kesişim noktasının açısı).

2.2.2.4. Daire-Daire Kesişimi

İki dairenin kesişebilmesi için merkezleri arasındaki uzaklık yarıçaplarının toplamından küçük olmalıdır (Şekil 2.23). Bu durumda daireler iki noktadan kesişirler. Bu kesişim noktalarından daireler yaylara parçalanır. Ayrıca daha öncede belirtildiği gibi daireler çeyrek noktalardan da parçalanır. Kesişim noktalarından daireleri parçalara ayırabilmek için bu noktaların koordinatlarının hesaplanması gerekir. Bu hesaplamayı yapabilmek için eşitlik 2.58-2.68'de verilen formüller kullanılmıştır.

Dairelerin merkezleri arasındaki uzaklık (2.58);

$$L = e + f = \sqrt{(C_{1X} - C_{2X})^2 + (C_{1Y} - C_{2Y})^2} \quad (2.58)$$



Şekil 2.23. İki dairenin kesişme durumu.

Üçgende meydana gelen E ve F çizgilerinin uzunlukları (2.59, 2.60);

$$E = \frac{(R_2^2 - R_1^2 - L)}{(-2L)} \quad (2.59)$$

$$F = L - E \quad (2.60)$$

Oluşan üçgenin yüksekliği (2.61);

$$H = \sqrt{(R_1^2 - E^2)} \quad (2.61)$$

Merkezler arası çizginin ortadan bölünme oranı (2.62);

$$N = E / F \quad (2.62)$$

M noktasının koordinatları (2.63,2.64);

$$H_x = \frac{C_{1X} + NC_{2X}}{(1 + N)} \quad (2.63)$$

$$H_y = \frac{C_{1Y} + NC_{2Y}}{(1 + N)} \quad (2.64)$$

Şeklinde hesaplanır.

($P_1 - M$) ve ($P_2 - M$) çizgilerinin yatayla yapmış oldukları açılar (α_1, α_2) program içerisinde kullanılan prosedür vasıtasıyla hesaplanırsa, P_1 ve P_2 kesişim noktalarının koordinatları için 2.65-2.68'deki eşitlikler kullanılır.

$$P_{1X} = H_x + H \cos(\alpha_1) \quad (2.65)$$

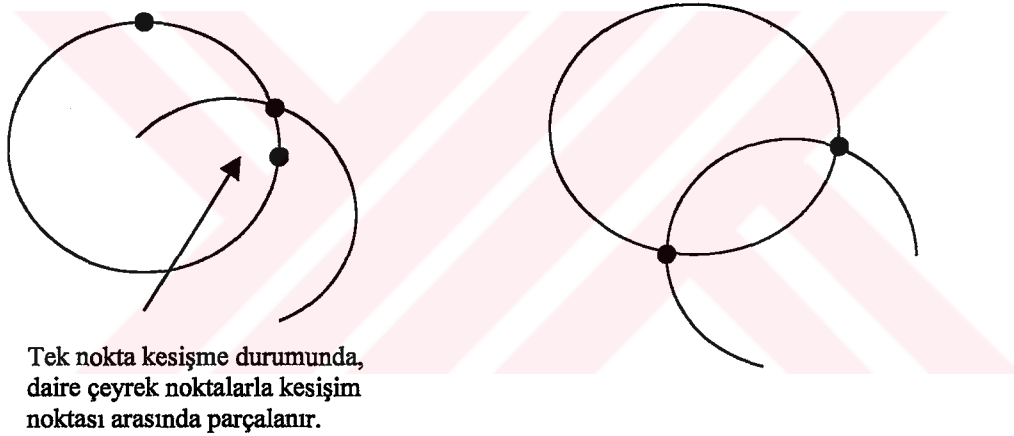
$$P_{1Y} = H_y + H \sin(\alpha_1) \quad (2.66)$$

$$P_{2X} = H_x + H \cos(\alpha_2) \quad (2.67)$$

$$P_{2Y} = H_y + H \sin(\alpha_2) \quad (2.68)$$

2.2.2.5. Daire-Yay Kesişimi

Daire-yay kesişmesi tıpkı daire-daire kesişmesine benzer. Başlangıçta her ikisi de daire kabul edilerek kesişim noktasının koordinatları yukarıdaki yöntemle hesaplanır. Daha sonra bu kesişim noktalarından hangisi yayın üzerinde ise o noktalar gerçek kesişim noktası olarak atanır ve daire ile yay o noktalardan ikiye bölünür. Gerçekte daire üzerinde 1 tane kesişim noktası olduğunda daire bu noktadan ikiye bölünemez. Fakat buradaki uygulamalarda daireler zaten çeyrek noktalardan ikiye bölündüğü için, daire üzerinde 2 den fazla bölüm noktası zaten oluşmaktadır (Şekil 2.24).



Şekil 2.24. Daire ve yayların kesişme durumu.

2.2.2.6. Yay-Yay Kesişimi

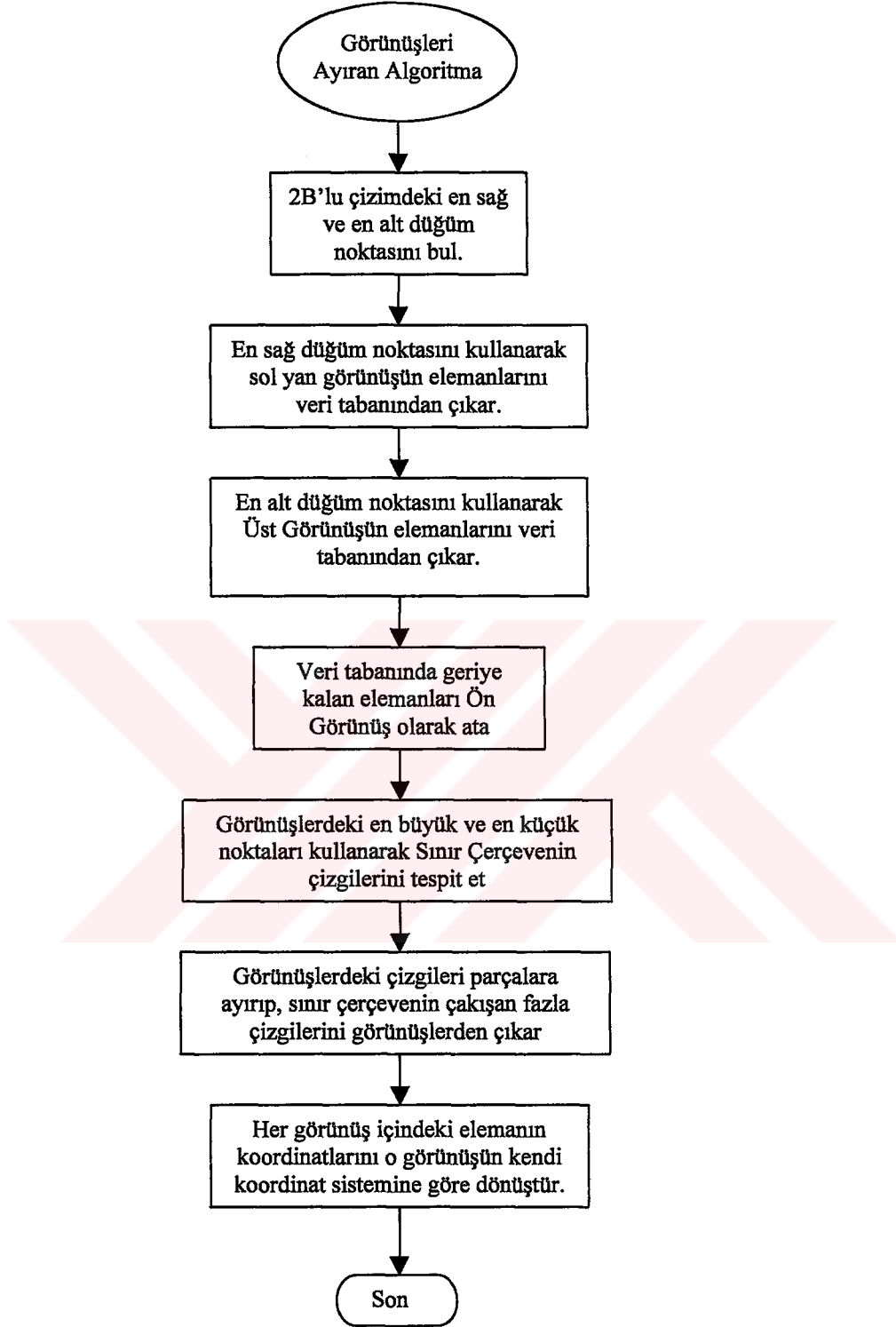
Yay-yay kesişmesi de tıpkı daire-daire kesişmesine benzer. Başlangıçta her ikisi de daire kabul edilerek kesişim noktasının koordinatları yukarıdaki yöntemle hesaplanır. Daha sonra bu kesişim noktalarından hangisi yayın üzerinde ise o noktalar gerçek kesişim noktası olarak alınır ve yaylar o noktalardan ikiye bölünür.

2.2.3. Görünüşlerin ve Sınır Çerçevenin Tespiti

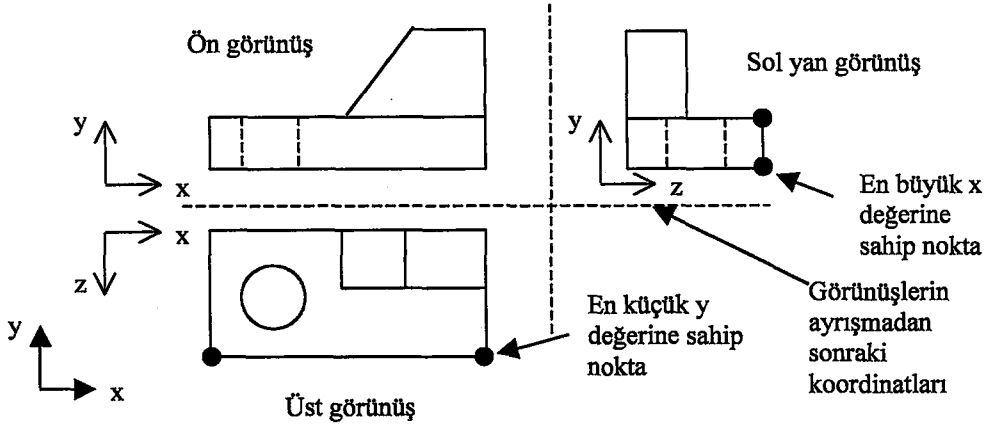
Görsel olarak ekranda 2B'lu ve üç görünüş halinde çizilen çizgilerin hepsi gerçekte veri tabanında bir arada bulunan çizgiler topluluğudur. Bunların, cismin üç dik görünüşüne (ön görünüş, üst görünüş, sol görünüş) ayrılması gerekmektedir. Bununla ilgili algoritma (Şekil 2.25)'de verilmiştir.

2.2.3.1. Görünüşlerin Tespiti

Çizimi görünüşlere ayırma ile ilgili prosedür kısaca şöyledir. En büyük "x" değerine sahip bir nokta bulunur. Bu noktaya bağlı bütün çizgiler ve kapalı döngülerin içinde kalan ada şeklindeki alanlara ait çizgilerle birlikte hepsi tespit edilir. Bu çizgilerin hepsi *Sol Görünüşe* ait olacaktır. Bunlar veritabanından çıkarılır. Geriye kalan çizgilerin içinden küçük "y" değerine sahip nokta bulunur. Yine bu noktaya bağlı çizgilerle birlikte kapalı alanların içinde kalan elemanlar tespit edilir. Tespit edilen bu çizgiler *Üst Görünüşü* oluşturur. Bunlar da veritabanından çıkarılır. Daha sonra geriye kalanlar ise *Ön Görünüşe* ait olacaktır. Her görünüşün kendi içindeki koordinat sistemi farklı olduğu için, görünüşlerin elemanları bu koordinatlara göre dönüştürülüp ilgili görünüşe kaydedilir (Şekil 2.26).



Şekil 2.25. Görüntüleri ve sınır çerçeveyi tespit eden algoritmanın yapısı.



Şekil 2.26. 2B'lu çizimin görünümlere taksimi.

2.2.3.2. Sınır Çerçevenin Tespiti

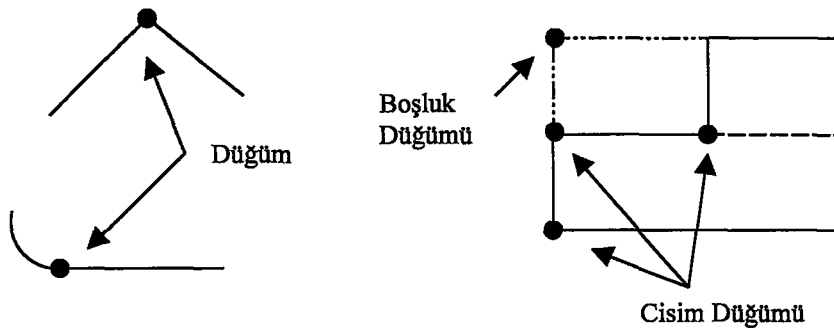
Görünümler tespit edildikten sonra her görünüş içindeki en büyük ve en küçük değerlere göre görünüşü içine alan en küçük sınır çerçevenin ölçüleri tespit edilir. 2B'lu çizim hazırlanırken sınır çerçeve görünümleri içine alacak şekilde çizime eklenirse, bu durumda sınır çerçeve cismi içine alan en küçük çerçeveden daha büyük olmuş olur. Görünümlerdeki sınır çerçeveler aslında sınır kutunun (ham malzeme) görünümlerini oluşturmaktadır. Zaten pratikte de parça işlenirken ham malzemeler işleme payları da göz önüne alınarak parçadan büyük alınır.

Programda sınır çerçevenin çizgileri bir çizgi ve iki noktadan oluşan, “bölüm” çizgisi kullanılarak gösterilmektedir. Sınır çerçevenin çizgilerini görünüş çizgilerine eklemek için, sınır çerçevenin çizgileri kesişen noktalardan parçalara ayrılır. Parçalara ayrılmış sınır çerçeveye ait çizgilerin aynısından cisme ait çizgi varsa sınır çerçevenin çizgisi burada fazla demektir ve bu çizgi görünüşten çıkarılır. Böylece kalan çizgilerden sınır çerçeveye ait olanlar görünüşe eklenmiş olur (Şekil 2.27).

2.2.4.1. Düğüm-Koordinat Dosyasının Oluşturulması

Düğümler her elemanın (doğru ve yay) uç noktalarının bulunduğu, elemanların bu noktalardan birbirine bağlandıkları yerdir. Düğümlerin koordinatları tespit edilmekte ve her düğüme bir sayıyla isim verilmektedir. Eğer düğüm cisme ait görüntüş çizgileri üzerinde ise, yani bağlı bulunduğu çizgi tipi sürekli ve kesikli ise bu köşeler “Cisim Düğümü” olarak adlandırılmaktadır. Bunun haricinde eğer düğüm, bölüm (tek çizgi, çift nokta) çizgisine bağlı ise bu düğümler “Boşluk Düğümü” olarak adlandırılmakta ve bu özelliği de dosyaya yazdırılmaktadır.

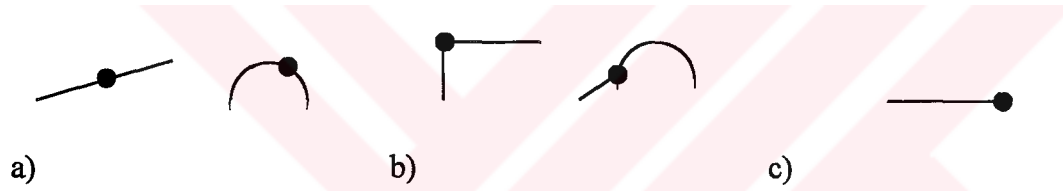
Düğümler tesbit edilirken, her görünüş içindeki bütün çizgilerin uç noktaları düğüm olarak koordinatlarıyla birlikte tek tek dosyaya kaydedilmekte, daha sonra bu dosyada aynı koordinata sahip fazla sayıdaki düğümler atılmaktadır. Eğer aynı koordinata sahip bir “Boşluk”, bir de “Cisim düğümü” varsa bu düğüm “Cisim düğümü” olarak tayin edilmektedir. Bir düğümün “Boşluk düğümü” olabilmesi için ona bağlı bütün çizgilerin “Bölüm çizgisi” tipinde olması gerekir (Şekil 2.28).



Şekil 2.28. Düğüm tipleri ve gösterimi.

Düğümler üç değişik şekilde ele alınmıştır.

1. Ln (lineer): Çizgiyi ikiye bölen, başka bir çizginin de düğüme bağlanmadığı, yayların çeyrek noktalarındakilerin hariç tutulduğu ve hiçbir görünüşte karşılığı olmayan düğümler (Şekil 2.29.a).
2. St (standart): Ayrı ikiden fazla çizginin bulunduğu ve her görünüşte karşılığı olan düğümler (Şekil 2.29.b).
3. Ep (endpoint): Çizginin bir ucundan başka hiçbir çizgiye bağlanmadığı, boşta kalan uç bu da düğüm olarak adlandırılmıştır (Şekil 2.29.c).



Şekil 2.29. Düğüm çeşitlerinin gösterimi.

2.2.4.2. Kenar-Düğüm Dosyasının Oluşturulması

Kenarlar, düğümler arasındaki bağlantıyı sağlayan, alanların sınırını oluşturan çizgilerdir. Kenarlara ait bilgiler dosyaya kaydedilirken, kenarın düğüm numarası kaydedilmektedir. Ayrıca kenar yay ise, yayın merkez noktasının koordinatları, yarıçap ve açıları da kaydedilmektedir. Bir kenar-düğüm dosyasında şu bilgiler toplanmaktadır.

Kenar No: Kenara verilen sayısal ad.

Kenar Tipi: Doğru veya yaydan hangisinin olduğunu gösteren kod..

Köşe1: Kenarın birinci ucundaki düğümün numarası.

Köşe2: Kenarın ikinci ucundaki düğümün numarası.

MerkezX: Yaylarda merkez noktasının X koordinatı.

MerkezY: Yaylarda merkez noktasının Y koordinatı.

MerkezZ: Yaylarda merkez noktasının Z koordinatı.

Yarıçap: Yayın yarıçapı.

Alfa Açısı: Yayın birinci ucunun X eksenini ile yaptığı açı.

Beta Açısı: Yayın ikinci ucunun X eksenini ile yaptığı açı.

Çizgi tipi: Çizginin sürekli, kesikli, eksen veya bölüm çizgilerinden hangisi olduğu.

Kenarları göstermek için dört tip çizgi kullanılmıştır. Bunlar sürekli, kesikli, eksen ve bölüm çizgileridir. Dairelerin eksenlerini gösteren eksen çizgisi 2B'lu çizimde çizilebilmekte, fakat algoritmada kullanılmamakta, yani yok sayılmaktadır.

1. Sürekli çizgi: Bu çizgi görüntülerde görünen kenarları temsil eder.

2. Kesikli çizgi: Bu çizgi görüntülerde görünmeyen kenarları temsil eder.

3. Eksen çizgisi: Dairelerin eksenlerini ve simetriyi temsil eder (Algoritmada kullanılmamaktadır).

4. Bölüm çizgisi: Bu çizgi görüntüleri içine alan sınır çerçeveyi temsil eder.

2.2.4.3. Dügüm-Kenar Dosyasının Oluşturulması

Her düğüm kendisine bağlı kenarlarla birlikte bu dosyaya kaydedilir. Bu dosya daha çok alanlara ait çizgileri tespit için kullanılmaktadır. Alanların çizgilerini tespit için, bir çizgiden düğüm üzerinden başka bir çizgiye geçerken, o düğüme bağlı diğer kenarların bulunması için bu dosyadaki bilgiler kullanılır. Dosyada düğümün numarası ve buna bağlı kenarların numarası tutulmaktadır. Buradaki düğüm numarası ile Dügüm-Koordinat dosyasındaki düğümlerin numarası aynı olmalıdır.

2.2.4.4. Köşe-Kenar-Koordinat Dosyasının Oluşturulması

Bir alanın içerisindeki döngüde kenarlar üzerinde dolaşırken, düğümler üzerinde o alana ait çizgilerin oluşturduğu köşeye “köşe” adı verilmiştir. Her köşe iki çizgiden oluşur. Her köşe aslında bir düğümdür. Düğümlerin üzerindeki her iki çizgi ayrıca “köşeyi” oluşturur. Köşeler bağladıkları çizgilerin tipine göre, köşenin diğer görüntülerde karşılığının olup olmama durumuna göre çeşitli tiplere ayrılmıştır. Köşelerin kullanıldığı esas yer, ileride detayları anlatılacak olan “İzdüşüm Yöntemidir”. Alanların eksenler üzerine yansıyan izdüşümlerini bulurken bu köşeler kullanılacaktır. Aşağıdaki ön görüntü için oluşturulabilecek çeşitli tipteki köşeler gösterilmiştir (Şekil 2.30). Diğer görüntülerdeki eksenler değiştiği için köşelerin adlandırılmaları da ona göre yapılmaktadır.

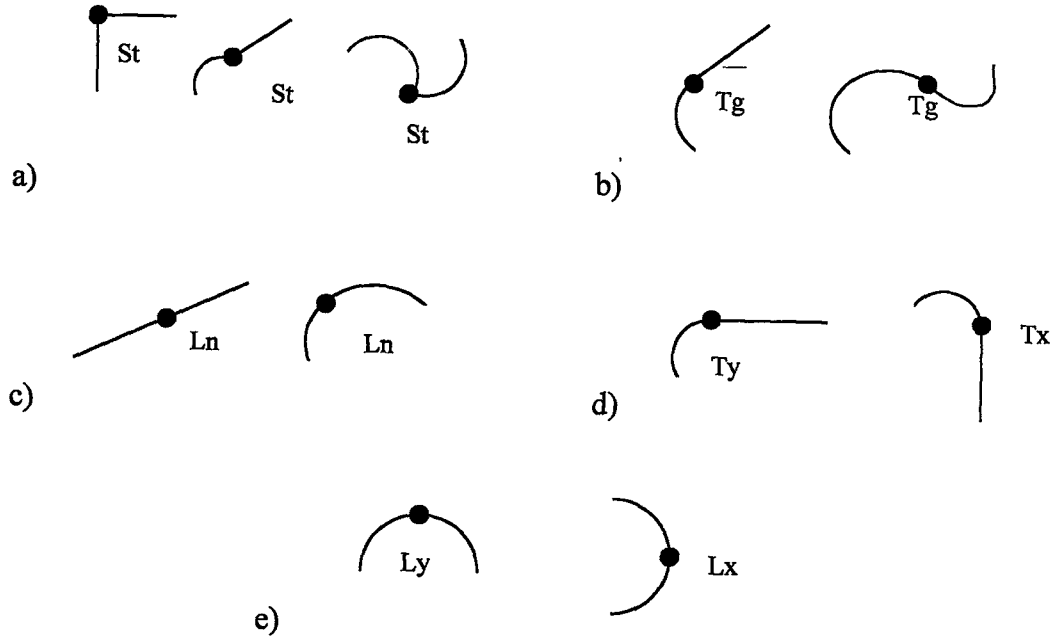
1. St (standart) köşe: Bu köşe her iki eksen üzerinde oluşturulan izdüşüm çizgisi üzerinde de nokta oluşturur (Şekil 2.30.a).
2. Tg (teğet) köşe: Bir doğru ile bir yayın ya da iki yayın teğet olarak değdiği noktalarda bu tip köşe olur. Burada teğet noktanın yayın çeyrek noktaları

haricinde olması gerekir. Bu durumda köşe diğer görünüşlerde görünmez ve izdüşümler üzerinde nokta oluşturmaz (Şekil 2.30.b).

3. Ln (lineer) köşe: Bir çizgi veya bir yay, ortasındaki her hangi bir noktadan bölünmüş ise, burada oluşan köşe lineerdir. Bu köşe yaylarda çeyrek noktalarda bulunmamalıdır. Köşe diğer görünüşlerde görünmez ve dolayısıyla izdüşümlerde nokta oluşturmaz (Şekil 2.30.c).
4. Tx (Teget-x): Bir çizgi ile yay, yayın çeyrek noktalarında teğet ise bu ismi alır. İsim verirken, köşe diğer görünüşlerde hangi eksende nokta oluşturursa o eksenin adı sonuna eklenir. Tx, diğer görünüşte x varsa o eksen üzerinde nokta oluşturur (Şekil 2.30.d).
5. Lx (Lineer-x): Bir yay çeyrek noktalarından bölünmüş ise, o yaya ait çizgiler lineer köşe oluştururlar. Bu tip bir köşenin izdüşüm noktasını oluşturduğu eksen, üzerine çizilecek teğet doğrultusuna dik olan eksenidir. O eksenin adı köşenin adının sonuna eklenir (Lx gibi) (Şekil 2.30.e).

2.2.4.5. Alanların Tespit Edilmesi

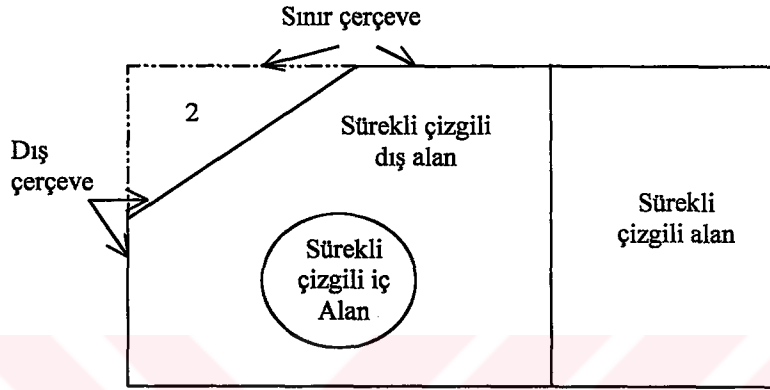
Görünümlerdeki bütün alanların tespit edilmesi ve bu alanların tipi, kenar numaraları, içinde başka bir alanın bulunup bulunmadığı, bulunuyorsa hangi alanın bulunduğu tespit edilip bir dosyada toplanır. Bununla ilgili olarak 6 çeşit alan tipi tespit edilip kullanılmıştır. Bunları belirlemek için kenar çizgilerinin tipi, alanın bitişik ya da ada şeklinde olup olmadığı özelliklerinden yararlanılmıştır. Belirlenen bu 6 çeşit alan tipi aşağıda verilmiştir (Şekil 2.31). Bu alanlar sadece temel görünüşte kullanılmış olup bütün kenarları sürekli çizgili alandır.



Şekil 2.30. Algoritmada kullanılan köşe çeşitlerinin gösterimi.

1. Sürekli Çizgili Alan: Bütün kenarları sürekli çizgiden oluşan ve sadece iç döngüye sahip alanlar.
2. Bölüm Çizgili Alan: Kenarlarından en az biri sınır çerçevesine ait olan alanlar.
3. Sürekli Çizgili İç Alan: Bütün kenarları sürekli çizgilerden oluşan fakat kendisi ada şeklinde iç alan olan alanlar. Bu alanlar başka bir alanın içinde bulunur ve başka hiç bir alanla komşu olmaz. Bu tip alanlar hem iç döngüye, hemde dış döngüye sahiptir.
4. Sürekli Çizgili Dış Alan: İçinde iç alan bulunduran ve kenarları sürekli çizgiden oluşan alanlar.

5. Dış Çerçeve: Görünüşte cisme ait en dıştaki çizgilerin oluşturduğu çerçeveye denir.
6. Sınır Çerçeve: Görünüşü içine alan ve sınır kutuya ait olan en küçük dikdörtgen çerçeveye verilen ad.

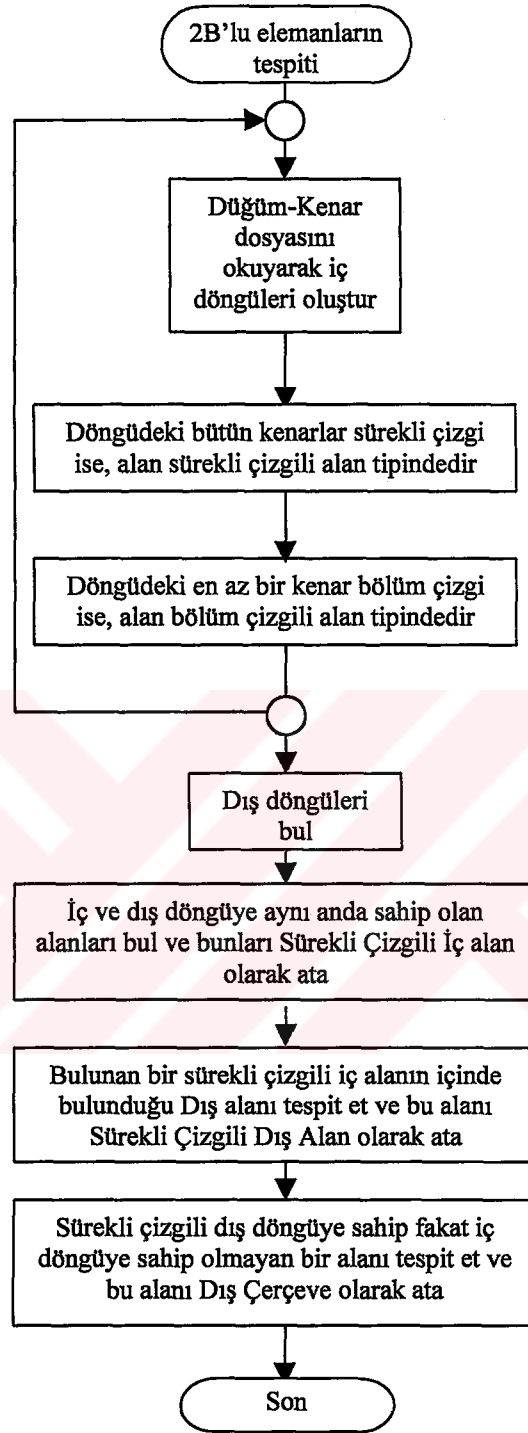


Şekil 2.31. Algoritmada kullanılan çeşitli alan tipleri.

2.2.4.5.1. Alanları Tespit İçin Uygulanan Kriterler

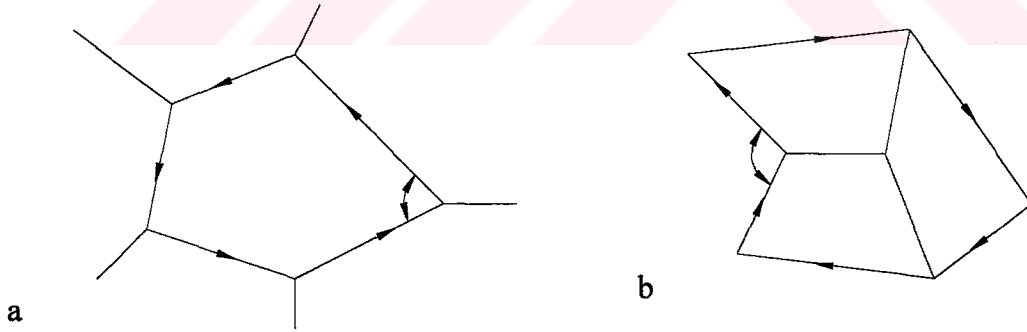
Alanları tespit eden algoritma bir kaç prosedürden meydana gelmektedir (Şekil 2.32). Her prosedürde alanların belli bir kısmı tespit edilmektedir.

Alanları tespitinde temel olarak, iç döngü-dış döngü durumu, kenar tipi, iç alan-dış alan durumu kriterler göz önüne alınır. Hangi kriterler kullanılarak hangi tip alanların tespit edildiğini aşağıdaki şekilde sıralayabiliriz.



Şekil 2.32. Alanların tespitinde kullanılan algoritmanın yapısı.

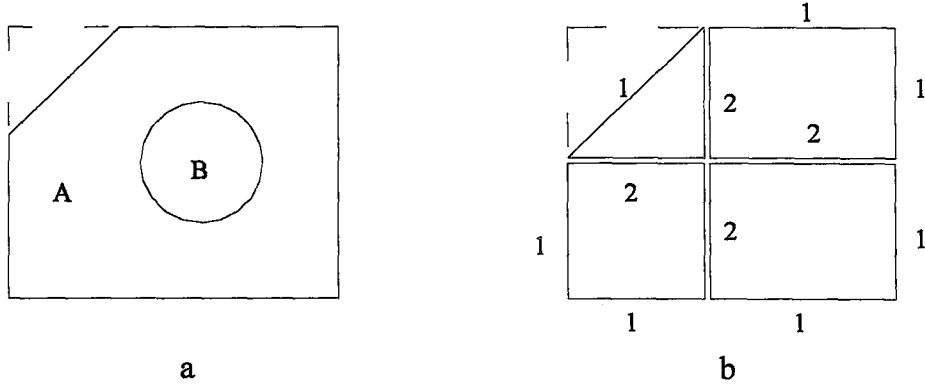
a) Bir alana ait kenarları tespit ederken, her seferinde sol taraftan en küçük açılı kenarlar alınır ve başlangıç noktasına ulaşana kadar bu işleme devam edilir. Eğer bu şekilde kenarları tespit ederken çerçevenin içerisinde dolaşmış ise burası bir “iç döngüdür” ve bir alan oluşturur. Dış kısımdan dolaşmış ise burası bir “dış döngüdür” ve bu döngünün alan oluşturması kesin değildir. Yani döngünün iç kısmında taranmayan başka çizgiler bulunabilir ve bunlar o döngünün bir alan oluşturmasını engeller. Şekil 2.33.’de her iki resimde de kenarlar üzerinde soldan dolaşıldığı halde birincisinde iç kısımdan dolaşmış ve bir “iç döngü” tespit edilip alan bulunmuştur. İkincisinde ise dış kısımdan dolaşmış ve bir alan tespit edilememiştir. İkincisinin içerisinde çizgiler olmasaydı burası da bir alan olabilirdi. Bu nedenle kenarlar üzerinde dolaşırken, dolaşımın içeriden mi yoksa dışarıdan mı yapıldığı büyük önem arz etmektedir.



Şekil 2.33. Sol taraftan en küçük açılı kenar yakalama metodu.

- b) Döngünün içten mi yoksa dıştan mı yapıldığını tespit ederken sol taraftan en küçük açılı kenarlar alınarak süpürülür ve her iki kenar arasındaki açılar toplanır. Döngünün sonuna ulaşıldığında toplanan açılar şu kriterlere uygulanır.
- i) Toplam Açı $< (\text{Kenar Sayısı} * 360) / 2$ ise bu döngü içeriden yapılmıştır yani iç döngüdür.
- ii) Toplam Açı $> ((\text{Kenar Sayısı}) * 360) / 2$ ise bu döngü dışarıdan yapılmış demektir. Dolayısıyla dış döngü bulunmuş demektir.
- c) Bir çerçevenin etrafında dolaşırken bu döngü eğer içten ise ve bunun dıştan bir döngüsü yoksa bu çerçeve görünüşteki bitişik alanlara karşılık gelir. Böyle bir döngüde bütün kenarlar sürekli çizgi ise bu alan *Sürekli Çizgili Alan* olarak adlandırılır. Eğer kenarlardan birisi bölüm çizgisi olursa bu alan *Boşluk Alan* olarak adlandırılır. Üzerinde kesikli çizgi bulunan alanlar alan olarak tespit edilmekte fakat temel alanda kullanılmadığından alan tipi belirlenmemektedir.
- d) Ele alınan bir çerçevenin hem dıştan döngüsü, hem de içten döngüsü varsa bu çerçeve bir ada şeklindeki *İç Alandır*. Yani komşu başka hiç bir alanla ortak kenarı yoktur. Mutlaka başka bir dış alanın içinde bulunur. Eğer başka bir alanın içinde bulunmazsa bu alan iç alan olarak adlandırılmaz. O takdirde *Dış Çerçeve* olarak adlandırılır. Şekil 2.34'deki ilk resimde her iki alanında içten ve dıştan döngüsü bulunduğu halde dıştaki (A alanı) *Dış Çerçeve*dir. Aynı zamanda *Sürekli Çizgili Dış*

Alan olarak da adlandırılır. İçteki ise (B alanı) Sürekli Çizgili İç Alan olarak adlandırılır.



Şekil 2.34. İç alan, Dış alan ve En Dış Çerçevenin tespiti.

- e) Eğer bir çerçevenin dıştan döngüsü var, fakat içten döngüsü yoksa ve bütün kenarları sürekli çizgi ise, bu yine *Dış Çerçeveyi* oluşturur. Eğer bu şekilde sadece dıştan döngüsü olan bir çerçeve bulunamazsa Dış Çerçeve ile aynı olan bir tane Sürekli Çizgili Alan var demektir ve bu alan Dış Çerçeve olarak kabul edilir. Bu durum görünüşte tek bir alanın olduğu durumlarda ortaya çıkar. Şekil 2.34.a.'daki alan aynı zamanda Dış Çerçivedir. Dış çerçevenin bir başka tespit yolu şöyledir. Görünüşlerdeki her çizgi iki alan tarafından kullanılır. Yalnız Dış Çerçeve ait olan en dıştaki çizgiler bir defa kullanılır. Buna göre her kenarın alanlarda kullanım sayısı tespit edilirse bunların içinden bir defa kullanılanlar Dış Çerçeveyi oluşturur (Şekil 2.34.b). Bu işlemde bölüm çizgili alanlar yok kabul edilmektedir.

- f) Eđer bir dngde dolařılan izgilerin iinde bir tane bile blm izgisi varsa o alan *Blm izgili Alandır*. Blm izgili alanlar Dıř ereve ile Sınır ereve arasında kalan kısımlardır (řekil 2.34.a).



3. ARAŞTIRMA BULGULARI

3.1. 3B'lu Modelin Oluşturulması ve Unsurların Tespiti

Buraya kadar olan bölümlerde cismin üç görünüşü çizilmiş ve bu çizgilerin algoritmada kullanılabilmesi için özellikleri, aralarındaki bağıntılar çıkarılarak dosyaya kaydedilmiş ve çizgilerin oluşturduğu alanlar tespit edilmiştir. Bundan sonraki bölümlerde ise algoritmanın 2B'dan 3B'a dönüşümü sağlayan esas kısımlarına geçilecektir. Bu bölümde adımlar kısaca şu başlıklar altında olacaktır. Temel görünüşün tespiti, projeksiyonların tespiti, taban ve tavan alanlarının oluşturulması, 3B'lu unsurların oluşturulması ve bunların tiplerinin tespiti, cismin oluşturulması ve geri kontrol işlemleridir.

3.1.1. Temel Görünüşün Tespiti

Algoritmayı uygulamaya koymak için öncelikle bir görünüşten başlamak gerekmektedir. Bu ilk olarak ele alınan görünüş "*Temel Görünüş*" olarak adlandırılacaktır. Temel görünüş cismin üzerindeki yüzeylerin en iyi şekilde görüldüğü görünüştür. Pratikte parça işlenirken, parçayı tablaya bağladığımızda ve yukarıdan baktığımız görünüştür.

Temel görünüş olarak hangi görünüşün seçileceği 3 şekilde belirlenebilir:

- 1) Baştan varsayım olarak görünüşlerden birisi kabul edilebilir. Bu üst veya ön görünüş olabilir. Ancak 2B'lu çizim esnasında cismin en uygun görünüşü, temel görünüş olarak kabul edilen görünüşün bulunduğu yere çizilmelidir.

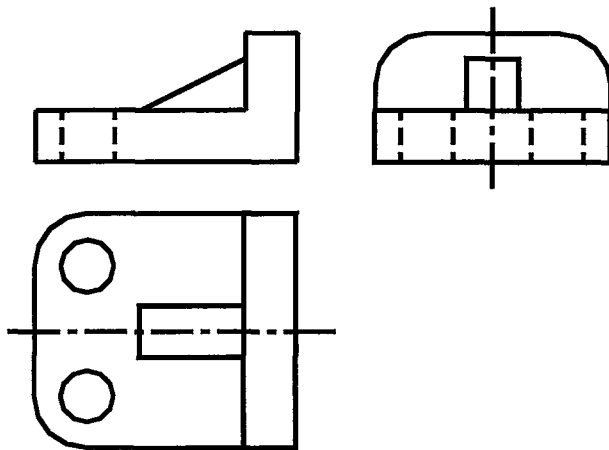
2) Hangi görünüş olduğu kullanıcı tarafından dışarıdan (kullanıcı etkileşimli) verilebilir.

3) Program tarafından belli kriterlere uyarak otomatik olarak seçilebilir.

Her üç durumda da temel görünüşün uygun bir şekilde seçilmesi gerekmektedir.

Bunun için gerekli olan kriterler aşağıda verilmiştir:

a) Parçanın simetrik olarak görüldüğü görünüş varsa, temel görünüş olarak önce o görünüş ele alınır. Eğer simetrik parçada bu yapılmazsa bakış doğrultusunun arkasındaki yüzeylerde, işlenemeyen çok sayıda hacim kalır. Yani bakış doğrultusu simetri ekseninin bir tarafından olacağı için, diğer tarafta daha işlenecek hacim kalacak demektir. Şekil 3.1'deki ön görünüş alınacak olursa, parça BSD tezgahlarında işlenirken eğik düzlemin arkasında kalan yüzey ile, parçanın L şeklinde görülen yanal yüzeyinin arkasında kalan yüzeyler işlenememiş olacaktır. Dolayısıyla seçilecek uygun görünüş zaten simetrikliği de sağlayan üst veya sol görünüştür. Bu ikisinden hangisinin seçileceğini tespit etmek için ise bir sonraki adıma geçilir.



Şekil 3.1. Temel görünüşün tespiti için en uygun görünüşün bulunması.

- b) İçerisinde en fazla yay bulunan görünüş ele alınır. Genelde BSD dik işleme merkezlerinde yaylar yukarıdan görüldüğünde tam olarak işlenir. Eğer kavisli yüzeye yandan bakılırsa, kesici bu yüzeyi işleyebilmek için kademe uygulamak zorunda kalır. Bu nedenle düzgün bir yüzey kalitesi elde edemez. Şekil 3.1'deki örnekte, simetriklikten dolayı üst ve sol görünüşten birini seçmek durumunda kalınmıştı. Bu ikisi için de en fazla radius üst görünüşte olduğu için bunu seçmek daha doğru olur.
- c) İçerisinde en az kesikli çizgi bulunan görünüş ele alınır. Kesikli çizginin az bulunması bakış doğrultusunun arka tarafında gözle görülmeyen yüzeylerin az olduğu manasına gelir. Eğer temel görünüş içerisinde kesikli çizgi bulunursa, bu kesikli çizgilere bağlı hacimler, algoritmanın temel görünüşe uygulama aşamasında tespit edilemezler. İleride anlatılacak olan geri kontrol işlemlerinden sonra tespit edilebilirler. Burada asıl amacın, parçanın işlenmesini temel görünüşte bitirmektir. Yani dik işleme merkezinde ilk bağlamada parçayı işlemektir. Şekil 3.1'deki örnekte üst görünüşte kesikli çizgi bulunmadığı için, sol görünüşe tercih edilir.
- d) Boşluk bölgesi (bölüm çizgisi bulunan alanlar) fazla olan ele alınır. Boşluk bölgesi görünüşte parçanın en dış sınırları ile, sınır kutu arasında kalan alandır. Bu bölge ne kadar fazla olursa işlenecek hacim o kadar çok demektir. Bu da parçanın en az sayıda bağlamasına yardımcı olan bir etkendir.

Bütün bu adımlar, kriter olarak sırasıyla uygulanır. En sonunda en fazla kritere hangi görüntüş sahipse o görüntüş temel görüntüş olarak ele alınır. Bu kriterler arasında net bir sınır yoktur. Eğer puanlamaya tabi tutulursa bu kriterlerin toplamından en yüksek puanı alan temel görüntüş seçilir. Puanlamanın yine önem sırasına göre büyükten küçüğe göre olması gerekir. Buna göre Şekil 3.1'deki parça, yukarıdaki kriterlerin her birine önem sırasına göre puan verilerek değerlendirilirse, temel görüntüş aşağıdaki şekilde tespit edilecektir (Çizelge 3.1).

Program otomatik olarak kendisi temel görüntüşü seçeceği zaman bu puanlama sistemini kullanmaktadır.

Temel görüntüş tespit edildikten sonra, algoritmanın daha sonraki adımlarına bu görüntüşten başlanır.

Çizelge 3.1. Temel görüntüşün tespitindeki kriterlerin örnek parçaya uygulanması.

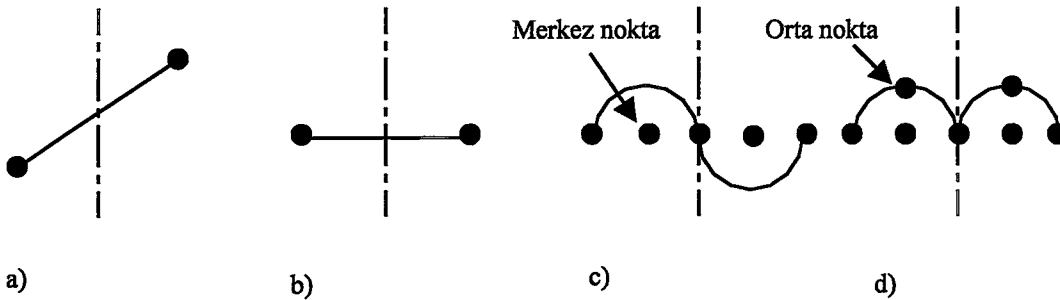
	Puan	Ön	Üst	Sol
1) Simetriklik bulunması	4	0	4	4
2) Kavisli çizgi en fazla	3	0	3+	3
3) Kesikli çizgi en az	2	0	2	0
3) Kesikli çizgi en az	1	1+	1	1
Sonuç		1+	10+	8

(+) sembolü, görüntüşlerin hangisinin içerisinde ilgili madde en fazla ise o görüntüşe eklenecek demektir. Sonuçta eşit puanlar çıkarsa artısı fazla olan öncelik sırasını alır. Bu puanlamaya göre örnek resmimizde en uygun temel görüntüş üst görüntüş olmaktadır.

3.1.1.1. Görünüşün Simetrikliğinin Bulunması

Bir görünüşte simetrikliği bulabilmek için hem çizgi hem de düğüm kontrolü yapılmalıdır. Çizgilerin tip (doğru, yay) ve çizim stili (kesikli, sürekli, bölüm) olarak kontrolleri yapılmalıdır. Yayların merkezlerinin simetrik ve yarıçaplarının da aynı olması gerekir. Simetriklik için referans alınacak eksenler, görünüşü içine alan sınır çerçevenin, kenarlarının ortasından geçen eksenlerdir. Simetrikliğin bulunması için aşağıda belirtilen hususlara dikkat edilmelidir:

- Simetri ekseninin her iki yanındaki çizgilerin simetrikliğini kontrol ederken bir taraftaki elemanlar diğer taraftaki elemanlarla karşılaştırılır. Burada çizgi tipi, çizgi stili ve uçlarının simetrikli kontrol edilir.
- Eğer bir çizgi simetri eksenine kesişiyorsa uç noktalarının eksenin her iki tarafında simetrik olarak durup durmadığı kontrol edilmelidir (Şekil 3.2.a,b).
- İki yayın simetrik olabilmesi için hem uç noktalarının hem de merkez noktalarının simetrik olması gerekir. Çoğu durumda buda yeterli olmadığı için yayın orta noktasının da simetrikliğini kontrol etmek gerekmektedir (Şekil 3.2.c,d).
- Bir doğru simetri eksenine çakışiyorsa simetrikliği bozamaz.



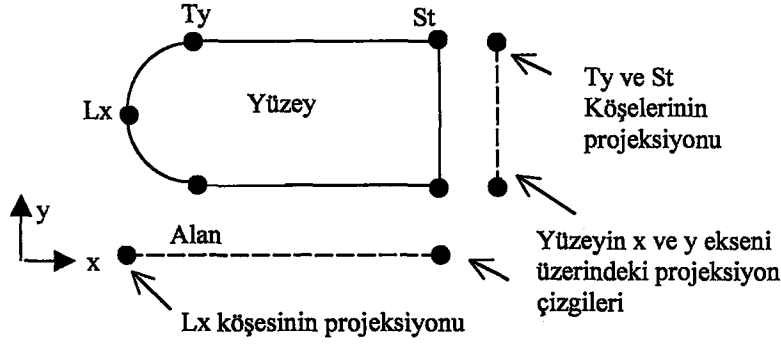
Şekil 3.2. Doğru ve yayda simetriklik durumları.

3.1.2. İzdüşüm Yöntemi

Temel görünüşteki alanların diğer görünüşlerdeki karşılığı olan çizgilerin veya alanların bulunabilmesi için “İzdüşüm Yöntemi” olarak adlandırılan bir yöntem kullanılmaktadır. Bunun için her görünüşteki alanların ve çizgilerin eksenler üzerine düşen izdüşüm çizgileri oluşturulur. Bu izdüşüm çizgilerinin üzerinde “Köşelere” karşılık gelen izdüşüm noktaları da yer alır. Daha sonra “temel görünüşteki” alanların karşılığı bulunmak üzere izdüşüm araştırmasına gidilir. Temel görünüşteki bir alanın diğer görünüşlerdeki karşılığını bulabilmek için önce bu alanın izdüşüm çizgisiyle aynı izdüşüm çizgisine sahip diğer görünüşlerdeki alan ve çizgiler araştırılır. Eğer diğer görünüşlerde aynı izdüşüm çizgisine sahip alan veya çizgi varsa, o alan veya çizgi temel görünüşteki alanın 3B’lu uzaydaki konumunu bulmayı sağlayacak çözümlerden biri olabilir. Bir kaç tane karşılığı olan çözüm çıkabileceği için bunların, detayları aşağıda verilecek olan diğer işlemler kullanarak elenmesi gerekmektedir.

3.1.2.1. İzdüşüm Çizgilerinin Oluşturulması

İzdüşüm çizgileri oluşturulurken her görünüşteki alanların, çizgilerin ve bu çizgilerin değişik birleşimlerinden oluşan grupların her birisinin izdüşümü eksenler üzerine yansıtılır. İzdüşüm alınırken gerek alanda gerekse çizgilerdeki her çizgi, eksen üzerinde düz çizgi şeklinde, ilgili köşeler eksenler üzerinde nokta şeklinde gösterilir. İzdüşümü alınan köşeler St, Tx, Lx tipindeki köşelerdir. St köşeler her iki eksene yansıtılır. Tx ve Lx köşeler sadece x ekseni üzerine, Ty ve Ly köşeler de sadece y ekseni üzerine Tz ve Lz köşeleri ise z ekseni üzerine yansıtılır (Şekil 3.3).



Şekil 3.3. Yüzeyin eksenler üzerindeki izdüşümlerinin oluşturulması.

Oluşturulan her yüzeyin, çizgilerin ve çizgilerden oluşan çizgi gruplarının izdüşüm bilgileri hepsi bir araya getirilerek dosyada toplanır. Bu dosyada izdüşüm değişkenleriyle ilgili olarak aşağıdaki bilgiler tutulur:

- Yüzey No: İzdüşümün temel görüntüdeki ait olduğu yüzey numarası.
- Görünüş: 1=Ön görünüş, 2= Üst görünüş, 3= Sol yan görünüş
- Tip: 1=Sürekli çizgili yüzey, 2=Kesikli çizgili yüzey (Bu ikisi yüzey için kullanılmaktadır), 3=Düz çizgi, 4=Eğik çizgi, 5=Kavisli çizgi (Bu üçü çizgiler için kullanılmaktadır).
- Çizgi No: Yüzey veya kenarların çizgi numaraları.
- ProX: İzdüşüm çizgisinin üzerindeki noktaların X koordinatı.
- ProY: İzdüşüm çizgisinin üzerindeki noktaların Y koordinatı.
- ProZ: İzdüşüm çizgisinin üzerindeki noktaların Z koordinatı.
- İç Alan Sayısı: Eğer yüzeyin projeksiyonu alınmıyorsa ve bu yüzey içerisinde iç alan bulunduran yüzey ise, iç alanların sayısı yüklenir.
- İç Alan No: Yüzeyin içindeki iç alanların adları saklanır.
- Eksen: Oluşturulan izdüşüm çizgisinin hangi eksende kullanılacağını gösterir.

3.1.2.2. Ortak Çözümlerin Bulunması

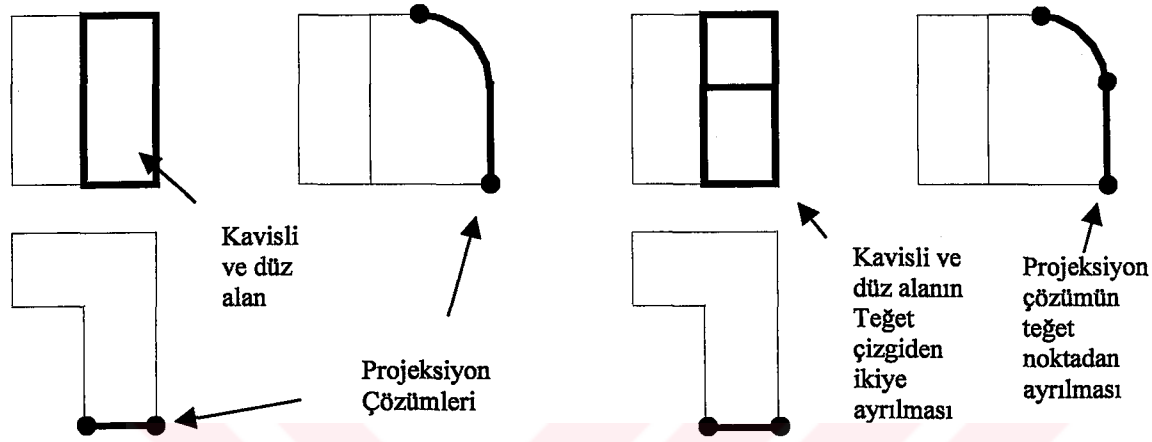
Temel görünüş içerisindeki sürekli çizgi ihtiva eden her yüzey tespit edilir. Bu yüzeylerin, eksenler üzerindeki izdüşümlerinin karşılığı, diğer görünüşlerde aynı eksenler üzerindeki izdüşümler içerisinde araştırılır. Bu araştırmada izdüşüm çizgisi üzerindeki noktaların diğer araştırılan izdüşüm çizgisi üzerinde de aynen bulunması gerekir. Bulunan izdüşüm çözümleri, temel görünüşteki sürekli çizgili alanın diğer görünüşlerdeki karşılığı olan alan veya çizgi gruplarını vermiş olur.

Temel görünüşteki yüzeyin diğer iki görünüşteki karşılığı olan çizgi veya yüzeyler bulunduktan sonra, bu çözümler içinde tekrar kendi aralarında ortak olan çözümleri araştırmak gerekmektedir. Yani temel görünüş olarak ön görünüş alınır, buradaki bir sürekli çizgili yüzeyin, üst ve sol görünüşte aynı izdüşüme sahip çözümleri bulunur. Daha sonra üst ve sol görünüşteki bu ortak çözümlerin içinde aynı ortak izdüşüme sahip olanlar tekrar bir daha değerlendirmeye alınır. Bunlar içinde ortak çözüme sahip olanlar ancak temel görünüşteki yüzeyin 3B'lu uzaydaki bilgilerini verebilir.

Temel görünüşteki bir sürekli çizgili yüzeyin diğer görünüşlerde karşılığı olan ortak izdüşüme sahip yüzey veya çizgiler dört çeşit olabilir. Bunlar “doğru, eğik çizgi (açılı duran doğru), yay ve alandır”. Yani temel görünüşteki bir yüzeyin karşılığı olan çizgiler diğer görünüşlerde bu dört şekilde bulunabilir.

3B'lu uzayda kavisli bir yüzeyle düz bir yüzeyin birbirine teğet bağlandığı durumlarda, bu yüzeylerin temel görünüşte teğet çizginin olduğu yerde yüzeyleri ayırıcı çizgi teknik resim kuralı olarak çizilmez. Bu yüzeylerin diğer görünüşlerdeki yansımaları ise “kavisli-düz çizgi” veya “iki kavisli çizgi” olmaktadır. Bu yüzeylerin

teğet birleşim yerinden parçalara ayrılması gerekmektedir. Bunun için öncelikle çizgiler bu teğet noktadan parçalara ayrılır. Daha sonra bu yüzeyler temel görünüşteki çizimine, çizilmeyen teğet çizgi eklenerek parçalara ayrılır (Şekil 3.4).

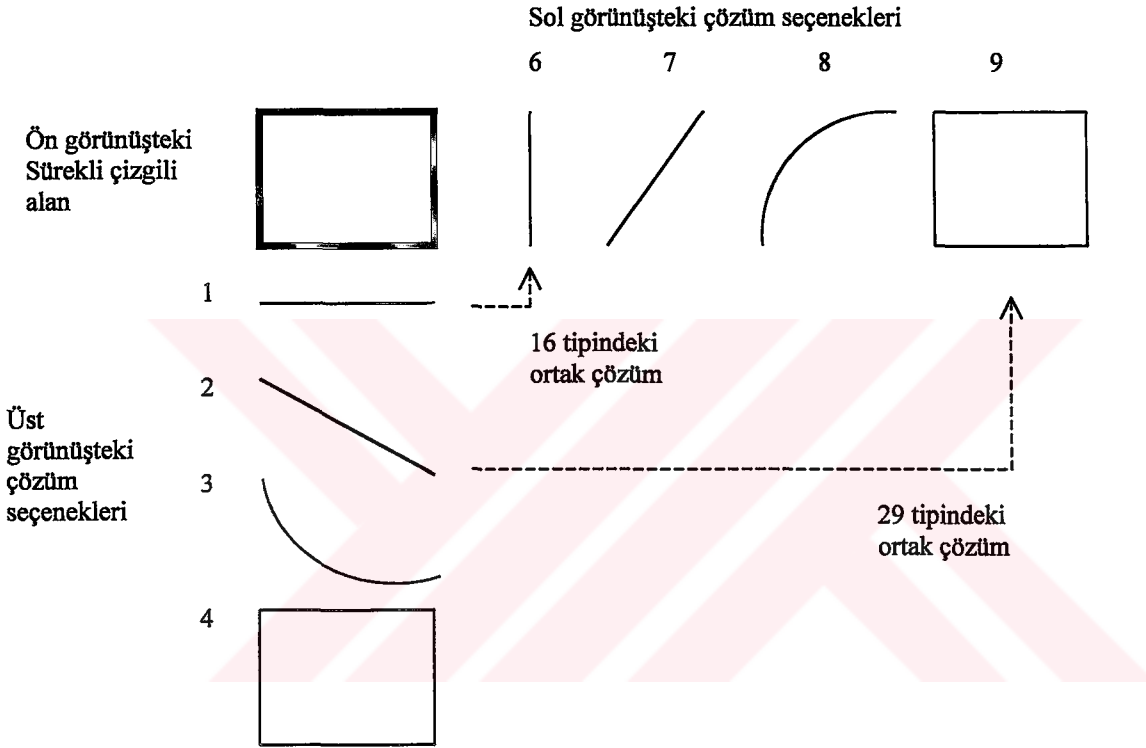


Şekil 3.4. Teğet yüzeylerin teğet çizgiden ikiye bölünmesi.

Temel görünüşün haricindeki diğer iki görünüşteki izdüşüm eşleştirmesi sonucu bulunan çözümlere bakılarak ortak çözümlerin tipleri tespit edilir. Bu çözüm tipine göre sürekli çizgili yüzeyin 3B'lu uzaydaki bilgilerinin elde edileceği prosedür işletilir. Ortak çözüm tipi için önce üst görünüşteki çözüm numarası, daha sonra sol görünüşteki çözüm numara alınır. Örneğin, Şekil 3.5'e bakılırsa üst görünüşteki çözüm olan eğik çizgi ile (2 numara) sol görünüşteki çözüm olan alan (9 numara) bir arada ortak çözümü oluştururlar ve bu çözüm tipi "29 tipindeki ortak çözüm" olarak adlandırılır.

Bazen ön görünüşteki yüzeyin 3B'lu uzayda birden fazla çözümü ortaya çıkmaktadır. Bu gibi durumlarda bakış doğrultusuna yakın olan yüzey ele alınacaktır.

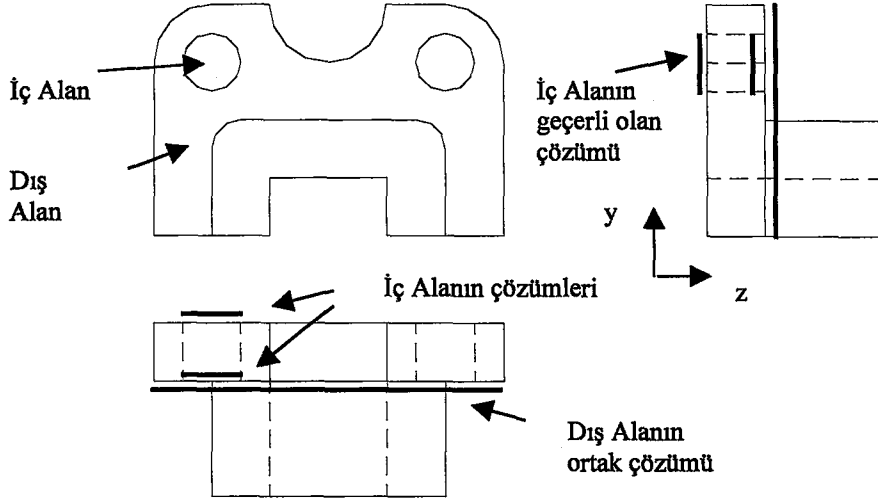
Yani temel görünüş olarak ön görünüş alınır, en büyük Z yüksekliğine sahip ortak çözüm, çözüm olarak kabul edilecektir. Eğer bir sürekli çizgili yüzeyin içinde “iç alan” olarak ikinci bir yüzey varsa bu yüzeyin yüksekliği dışında bulunan “dış alanın” kiyle aynı olamaz. Z yüksekliği bu durumda ya büyüktür ya da küçüktür. Büyükse iç alan adadır, küçükse ceptir.



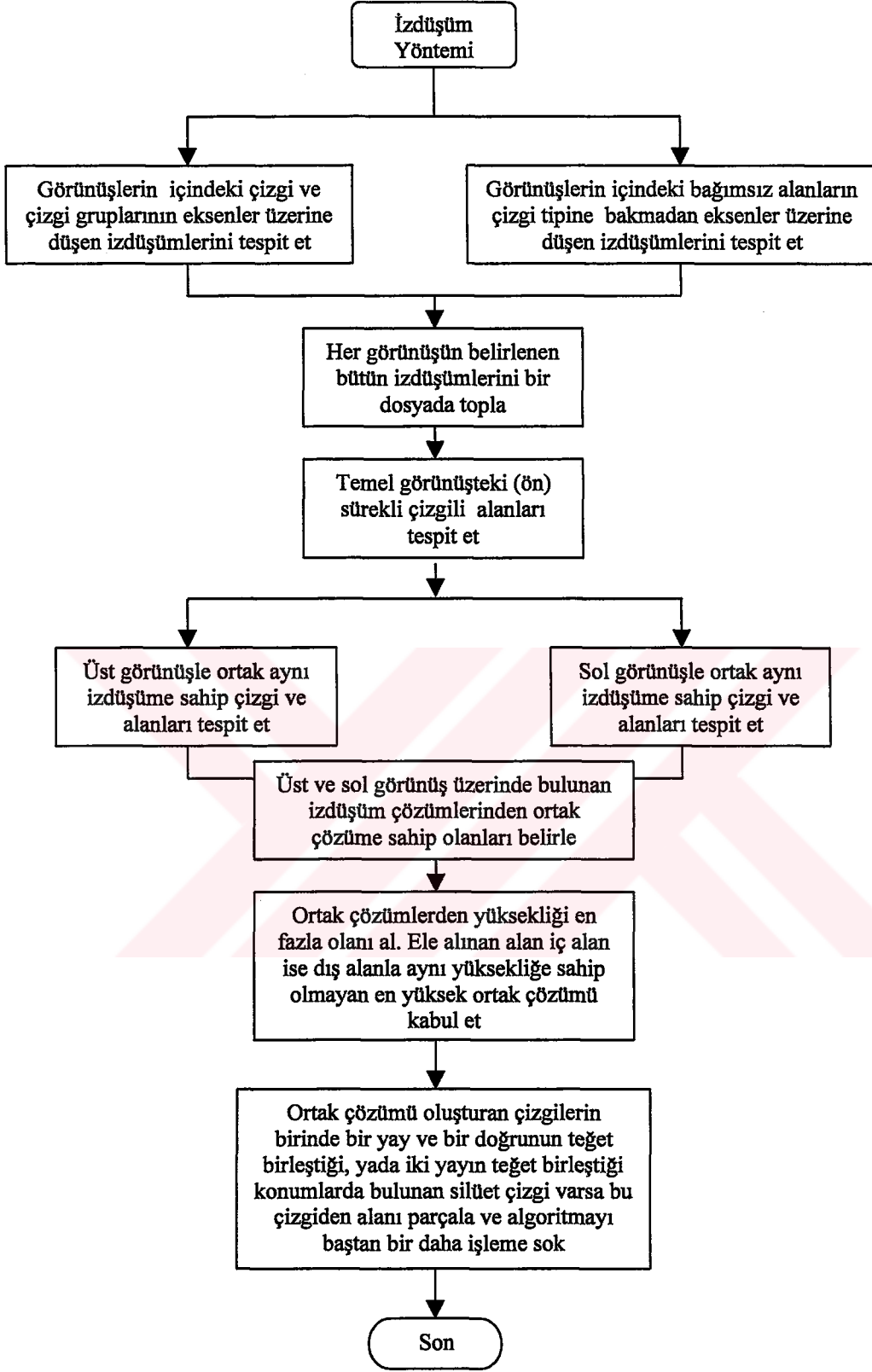
Şekil 3.5. İzdüşümlerden ortak çözümlerin bulunması.

Şekil 3.6’da daire ile gösterilen alan iç alan olarak görülmektedir. Bu alanın izdüşüm çözümüyle iki tane ortak çözümü ortaya çıkmaktadır. Bu çözümlerden birisi, dış alanın çözümü ile aynı Z yüksekliğine sahip olduğu için deliğin ortak çözümü olamaz. Dolayısıyla daha düşük Z yüksekliğine sahip olan ortak çözüm deliğin çözümü olur.

İzdüşüm yöntemiyle ilgili olarak geliştirilen algoritmanın yapısı Şekil 3.7'de verilmiştir.



Şekil 3.6. İç ve dış alanın ortak izdüşüm çözümleri.



Şekil 3.7. İzdüşüm yöntemiyle ilgili olarak geliştirilen algoritmanın yapısı.

3.1.3. Taban ve Tavan Yüzeylerin Oluşturulması

Cismin üzerindeki unsurların hacmini oluşturabilmek için, bu hacmin taban kısmını oluşturacak alana “*Taban yüzeyi*” denir. Hacmin üst yüzeyini oluşturan alana da “*Tavan yüzeyi*” denir. Bunlar algoritma içinde belirlendikten sonra “*Süpürme Taban*” olarak adlandırılan alanlar belirlenir ve hacimler bu süpürme tabanın, taban ve tavan arasında süpürülerek oluşturulmasıyla elde edilir. Taban yüzeyleri izdüşümlerin ortak çözümü sonucu oluşturulan 3B’lu yüzeylerdir. Tavan yüzeyleri ise her kullanılan eski tavandan, önceki tabanın çıkarılmasıyla oluşturulmaktadır.

Tavan, taban ve hacimlerde kenarlar anlamlı renklerle gösterilmiştir. Kullanılan renkler imalat aşamasında kesicinin yüzeylerden girip giremeyeceğini, unsurun tipini ve işlem sırasını tespit etmekte kullanılmaktadır. Kullanılan renklerle ilgili olarak detaylar aşağıda verilmiştir.

3.1.3.1. Algoritmada Kullanılan Renkler ve Anlamları

Oluşturulan unsurlarda kesicinin güvenle gireceği yüzeyleri tespit etmek ve unsurun tipini belirlemek için kenar ve yüzeylerde 3 değişik renk kullanılmıştır. Bu renk kodlaması taban, tavan, cisim ve unsurların hepsinde anlamlı olarak geçerlidir. Bu renklerin taşımış olduğu anlamlar şöyledir.

1. Kırmızı renk: Bu rengin kullanıldığı alanlardan veya kenarlardan kesici giremez. Dolayısıyla cismi oluşturan hacmin yüzeyleri kırmızı renktedir. Unsuralarda ise cisimle ortak olan yüzeyler bu rengi almaktadır.
2. Yeşil renk: Bu rengin kullanıldığı yüzeylerden kesici güvenle girebilir. Unsurun dış boşluğa bakan yüzeylerinde kullanılmıştır.

3. Mavi renk: Bu renk komşu iki unsurun ortak yüzeylerinde kullanılmıştır. Unsurlardan birisi işlendikten sonra diğer unsurun bu mavi yüzeyi yeşile dönmektedir. Mavi rengin kullanıldığı yüzeylerden kesici direk olarak giremez. Ancak komşu unsur işlendikten sonra bu yüzey yeşil renge döner ve daha sonra kesici girebilir.

3.1.3.2. Taban Yüzeylerin Tespiti

Temel görünüşte sürekli çizgilerden oluşan alanların 3B'lu uzayda yeri ve şekli, izdüşüm yöntemiyle tespit edildikten sonra, bu yüzeyler çıkarılacak hacimlerin (unsurların) alt yüzeyini belirlemede kullanılmaktadır (Şekil 3.8). Bu yüzeyler hacmi oluşturmak için durak taban yüzeyi olarak kullanılmamaktadır. Taban yüzeyi olarak, tavanlardan oluşturulan ve “Süpürme Taban” olarak adlandırılan yüzeyler kullanılmaktadır. Süpürme taban, tavan yüzeyle aynıdır, fakat Z yüksekliği ise tabanla aynıdır. Tabanların bütün kenarlarının rengi kırmızı renk olarak kullanılmaktadır.

Tabanlar 3 grupta toplanmıştır:

- a) Düz ve Altı Dolu Tabanlar: Bu tip tabanlar eğik veya kavisli olmayan düz yüzeylerdir. Tabanın altı cismin kendisini oluşturur ve dolayısıyla doludur.
- b) Eğik ve Kavisli Tabanlar: Bunlar eğik veya kavisli şekle sahip tabanlardır. Kullanımda her ikisi de aynı prosedüre göre ele alındığı için, bir arada anılmaktadır.
- c) Düz ve Altı Boş Tabanlar: Bu tabanlar kavisli veya eğik duran yüzeylerin bakış doğrultusunda izdüşümü alınmış halidir ve dolayısıyla altı boştur.

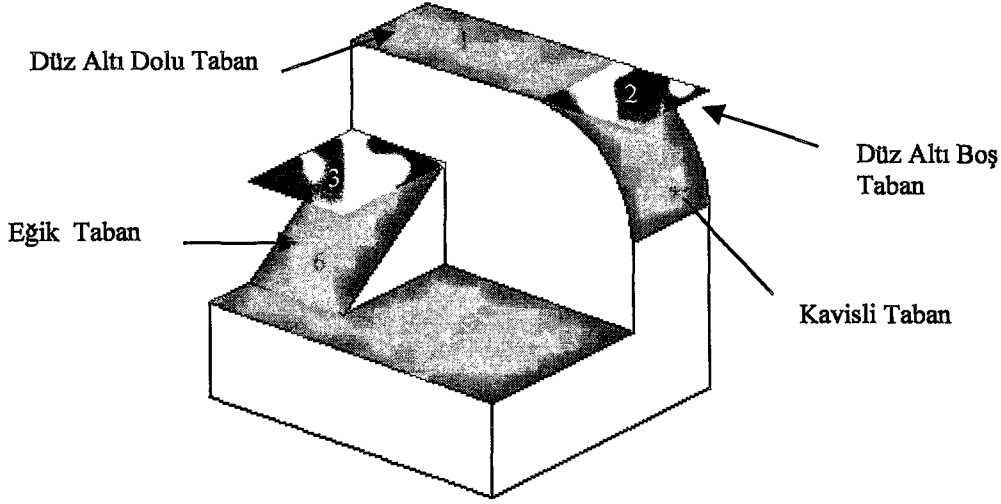
3.1.3.3. Unsurların İşlem Sırasının Tespiti

Taban yüzeyleri, çıkarılacak unsurların imalat aşamasındaki işlem sırasını da belirlemektedir (Şekil 3.8). Çıkarılacak unsurlar gerçekte, parça üzerinde işlenecek hacimler olduğu için bu hacimlerin yukarıdan aşağıya doğru belli bir düzen ve kural dahilinde imalat sırasında işlenmesi gerekir. Tabanların belli kurallar dahilinde sıralanması, imalat sırasının da tespit edilmesini sağlamaktadır. Tabanların sıralanmasında uygulanan kurallar aşağıdaki verilmiştir. Bu kurallar daha çok imalattaki tecrübeyle elde edilen bilgileri kullanmaktadır.

Burada geçen Z_1 ifadesi taban yüzeyin en yüksek Z yüksekliği, Z_2 ise tabanın en düşük Z yüksekliğidir. Dolayısıyla düz tabanlarda bu iki değer eşit ($Z_1=Z_2$), kavisli ve eğik tabanlarda ise $Z_1>Z_2$ olur. Tabii burada temel görünüş olarak ön görünüş alındığında tabanların yüksekliği Z koordinatıyla ifade edilmektedir. Temel görünüş farklı görünüşlerde ele alınırsa yüksekliği ifade eden koordinat değişir.

Tabanların sıralamasında burada Z_2 değerleri (alt yükseklikler) referans alınır. Buna göre şu kurallar uygulanır:

- a) İki düz tabandan hangisi daha yüksekte (Z_2 değeri daha büyük olan) ise o sıralamada önceliği alır.
- b) Bir düz ve bir eğik veya kavisli tabanın Z_2 değerleri eşit olduğunda öncelik düz tabana verilir. Eğik veya kavisli taban daha sonra ele alınır.
- c) İki kavisli veya eğik tabandan hangisi daha yukarıda ise (Z_2 değeri büyükse) o önceliği alır.



Şekil 3.8. Taban çeşitlerinin gösterimi (işlem sırasına göre adlandırılmıştır).

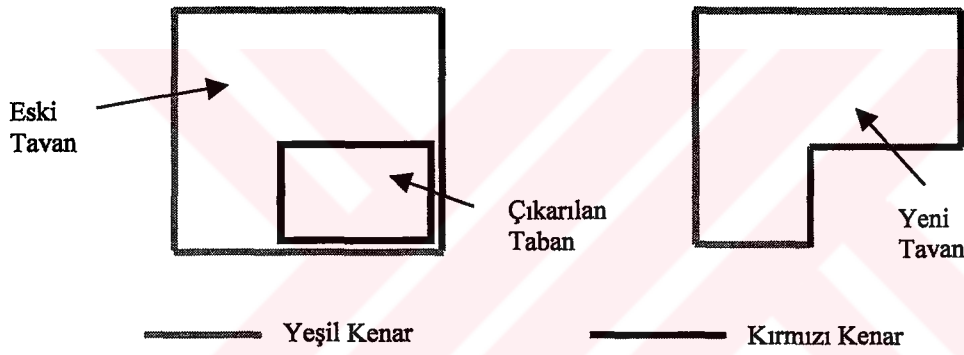
3.1.3.4. Tavan Yüzeylerin Oluşturulması

Çıkarılacak unsurların üst yüzeyini oluşturacak olan alanlar *Tavan* olarak adlandırılmıştır. Bu alanlar daha sonra anlatılacak olan süpürme tabanı oluşturmada, süpürme yüksekliğini belirlemede kullanılmaktadır. Her oluşturulan hacimden sonra bir sonraki hacimde kullanmak üzere yeni tavanlar oluşturulmaktadır. Bunun için öncelikle “ilk tavanın” oluşturulması gerekiyor. İlk tavan olarak sınır kutunun (ham malzemenin) dikdörtgen şeklindeki en üst yüzeyi alınır. İlk tavanın kenarlarının rengi yeşildir.

Tavanlar, hacim oluşturulduktan sonra unsura ait olan taban yüzeyi, yine unsura ait olan tavandan çıkarılarak oluşturulur. Bir tabanın, tavandan çıkarılabilmesi için o tavanın çerçevesinin içinde bulunması şarttır.

Yeni tavanlar oluşturulurken, çıkarılacak tabanın tipine bağlı olarak iki değişik yöntem kullanılmaktadır.

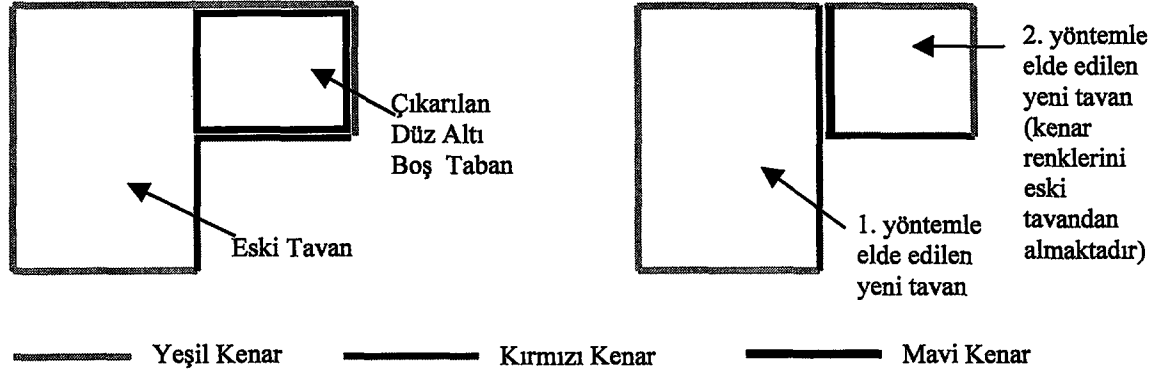
Yöntem 1: Eğer unsurun tabanı “düz altı dolu” taban veya “eğik, kavisli taban” tabandan biri ise bu yöntem kullanılır. Unsur oluşturulduktan sonra, unsurun tabanı, yine unsura ait tavandan çıkarılarak “yeni tavanlar” elde edilir. Önceki kullanılmış tavan, tavan dosyasından silinir ve yeni elde edilen tavan veya tavanlar dosyaya eklenir. Yeni eklenen tavanların Z yüksekliği ise çıkarılan tabanınkiyle aynı alınır. Bu işlemde birden fazla tavan ortaya çıkabilir. Burada yeni oluşturulan tavanın rengi, kendi üzerindeki renkler ile çıkarılan tabanın kırmızı renkleri muhafaza edilerek tespit edilir (Şekil 5.9). Oluşturulan her yeni tavan düzlemsel şekle sahiptir.



Şekil 3.9. Birinci yöntemle yeni tavanın oluşturulması.

Yöntem 2: Oluşturulan hacmin tabanı “düz altı boş” taban ise bu yöntem kullanılır. Bu yöntemde, önce yukarıdaki birinci yöntemle yeni bir tavan oluşturulur. Ardından kendisiyle aynı ikinci bir tavan daha oluşturulur. Yani en az 2 adet yeni tavan oluşturulur. Burada çıkarılan tabanla aynı olan yeni ikinci tavanı oluştururken, kenar renkleri eski tavandan alınmaktadır. Eski tavanla tabanın ortak olmayan kenarları var ise o zaman o kenarın rengi yeni tavanda mavi olarak seçilmektedir.

Yani bu ikinci yeni tavanın renklerine tabanın renkleri etki etmez (Şekil 3.10).



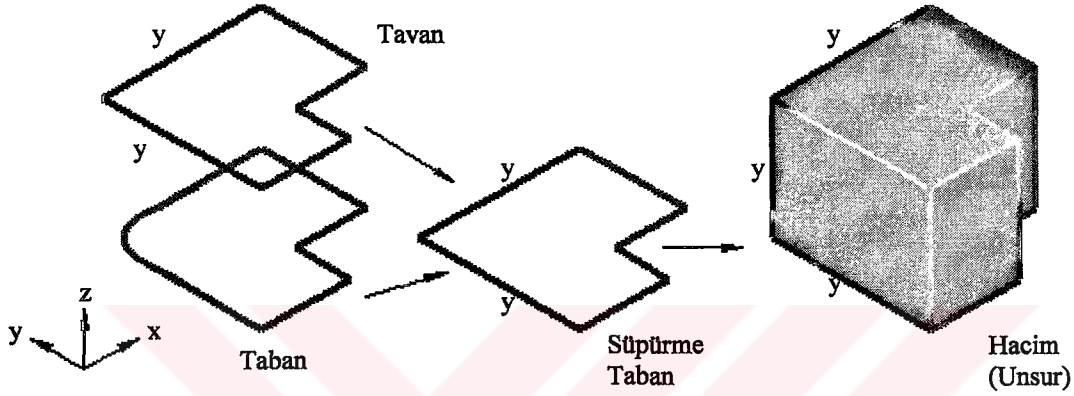
Şekil 3.10. Düz altı boş tabanın eski tavandan çıkarılmasıyla oluşturulan yeni tavanlar.

3.1.3.5. Süpürme Taban ve Unsurların Oluşturulması

Her yeni oluşturulan unsur için öncelikle iki alanın belirlenmesi gerekmektedir. Bunlardan birincisi taban, ikincisi ise tavadır. Tabanlar izdüşüm yöntemiyle oluşturulmakta idi. Tavanlar ise daha önceden oluşturulan hacimlerden sonra yeni tavanların elde edilmesiyle oluşturuluyordu. Fakat hacmi oluşturacak olan yüzey her ikisi de değildir. Bu nedenle hacmi oluşturmak için bir süpürme tabana ihtiyaç vardır. *Süpürme tabanlar (S-taban)*, eldeki tabanın düzlemsel, kavisli veya eğik olma durumuna bakarak iki şekilde elde edilirler. Ayrıca daha sonra açıklanacağı üzere süpürme taban oluşturulacak olan unsurun tipini ve kenarlarının rengini belirlemede kullanılır.

Taban Düzlemsel Durumda: Taban düzlemsel ise, hacme ait tavan yüzeyin şekli değiştirilmeden, tavanın Z yüksekliği, tabanın Z mesafesine kadar indirilir

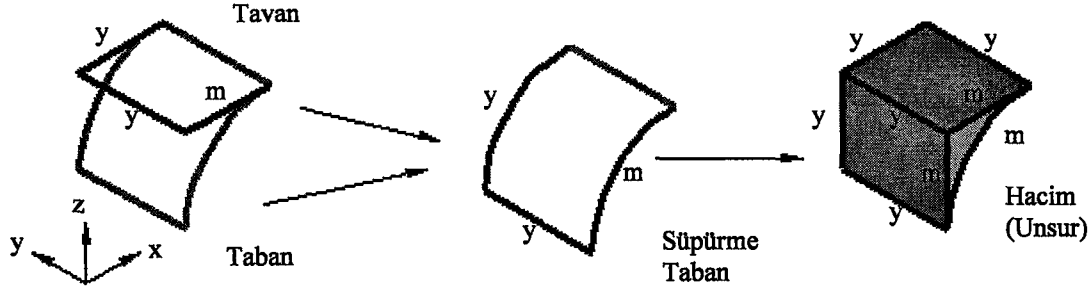
($Z_{\text{taban}}=Z_{\text{süpürmetaban}}$). Şekli ise tavanın şekliyle aynı alınır ve süpürme taban olarak kullanılır. Daha sonra bu süpürme taban bulunduğu Z yüksekliğinden tavanın Z yüksekliğine kadar süpürülerek unsurun hacmi oluşturulur. Süpürme tabanın kenar renkleri tavanın kenar renkleriyle aynı alınır ve bu renkler oluşturduğu hacme aynen yansıtılır (Şekil 3.11).



Şekil 3.11. Taban düzlemsel iken süpürme tabanın ve unsur hacminin oluşturulması (“y” yeşil kenarları göstermekte. Diğer kenarlar kırmızı renktedir).

Taban Kavisli veya Eğik Durumda: Oluşturulacak hacme ait olan taban kavisli veya eğik ise süpürme taban olarak bu durumda tabanın kendisi alınır ve yüksekliği de tabanınkiyle aynıdır. Yalnız süpürme tabanın kenarların rengi ise tavadan alınır. Yani şekil ve yükseklik tabandan, kenar renkleri ise tavadan alınır (Şekil 3.12).

Taban, tavan, süpürme taban ve unsuru oluşturma ile ilgili algoritma aşağıda verilmiştir (Şekil 3.13).



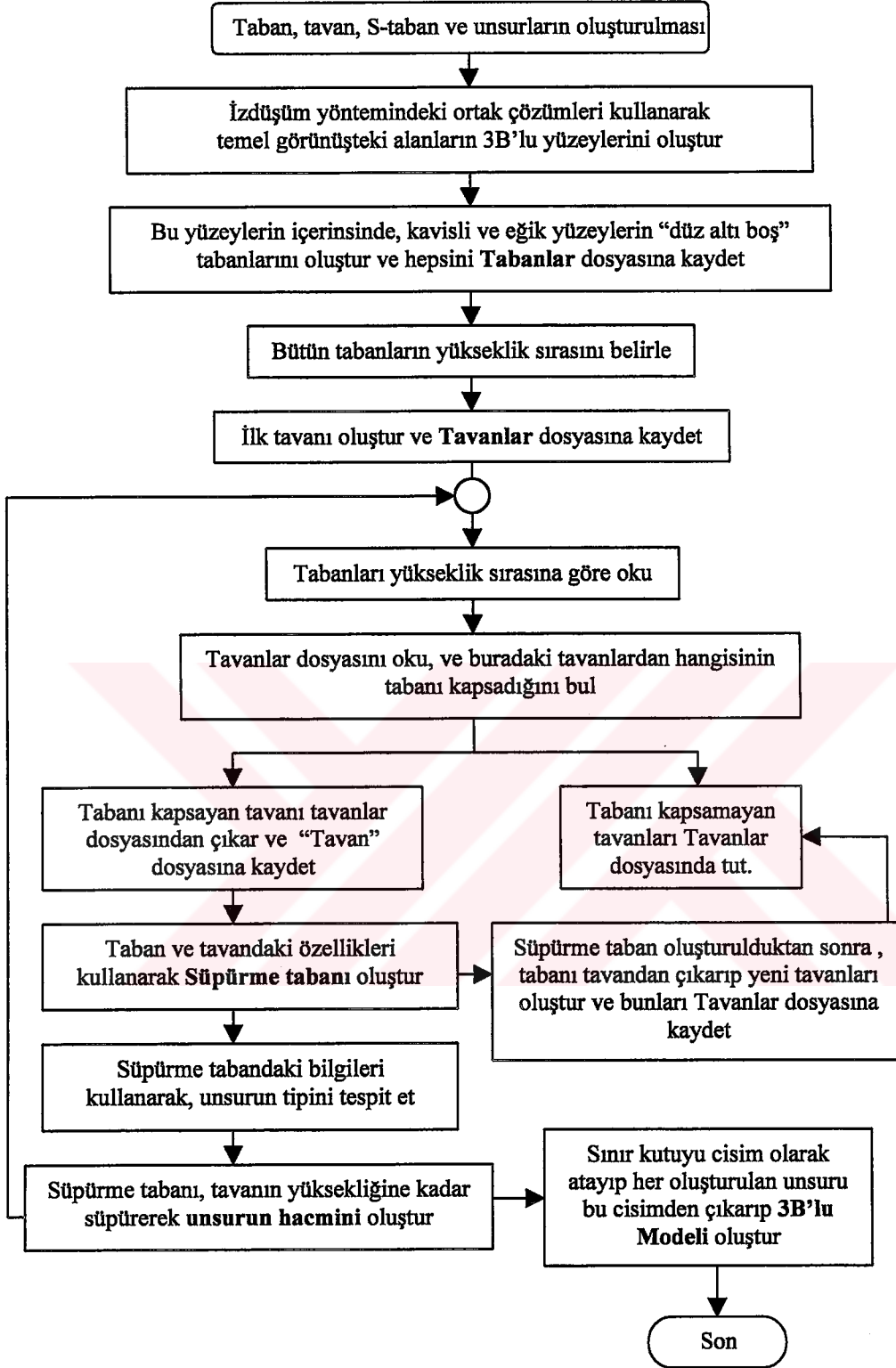
Şekil 3.12. Taban kavisli veya eğik durumda iken süpürme taban ve unsur hacminin oluşturulması (y = yeşil kenarı, m = mavi kenarı göstermektedir. Diğerleri kırmızı renktedir).

3.1.4. Unsur Tipinin Bulunması

Unsurların tasarım sonrası imalat ve işlem planlama gibi alanlarda kullanılabilmesi için unsur tiplerinin belirlenmesi büyük önem arz etmektedir. Örneğin, bir imalat işleminde bir delik unsuru için BSD’de uygun matkap seçilmesi veya bir T kanal için, uygun T kesicinin seçilmesi hep unsurların tipinin ve geometrisinin bilinmesine bağlıdır.

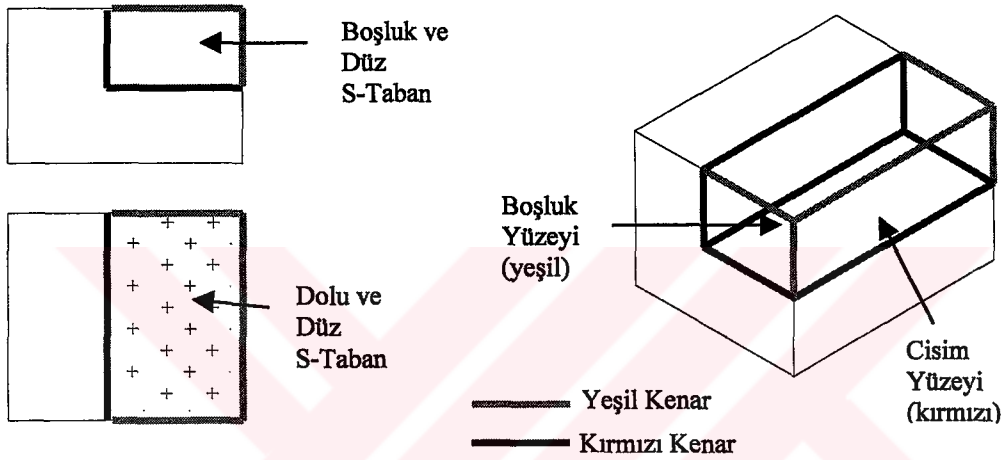
2B’lu geometriden unsur tiplerini belirlemek 3B’lu geometriye nazaran daha kolaydır. Özellikle bu çalışmada verilen metot, işlemi daha da kolaylaştırmıştır.

Burada kullanılan metotta unsur tipini belirlemek için süpürme tabanların (S-taban) özelliklerinden faydalanılmıştır. Süpürme tabanın geometrisi, kenar renkleri ve tipi (düz, kavisli ve eğik), unsurun tipini belirlemede kullanılan değişkenler olmuştur. Bu metotla 16 çeşit unsur tipi bulunabilmektedir. Bu unsurlar sayfa 111, Çizelge 3.2’de detaylıca verilmiştir.



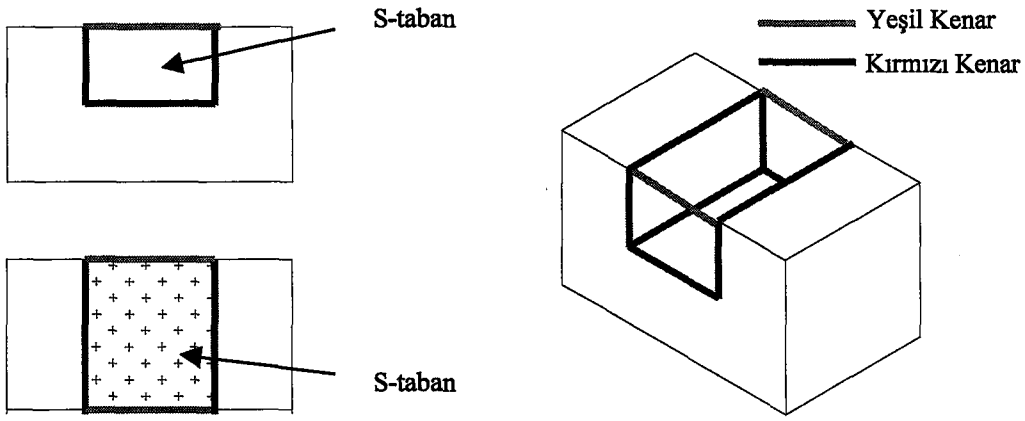
Şekil 3.13. Taban, tavan, süpürme taban ve unsuru oluşturan algoritmanın yapısı.

Kademe Unsuru: Dikdörtgen prizma şeklindeki bu unsurun 2 yüzeyi cisim tarafında (kırmızı yüzeyler), 4 yüzeyi ise boşluk tarafındadır (yeşil yüzeyler) (Şekil 3.14). Kesici karşılıklı iki boşluk yüzeyden girip çıkabilmektedir. Süpürme taban (S-Taban) temel görünüşte iki farklı şekilde görülmektedir. Bu tabanlardan birincisi boşluk taban, ikincisi ise dolu tabandır. Her ikisi de düz yüzeydir.



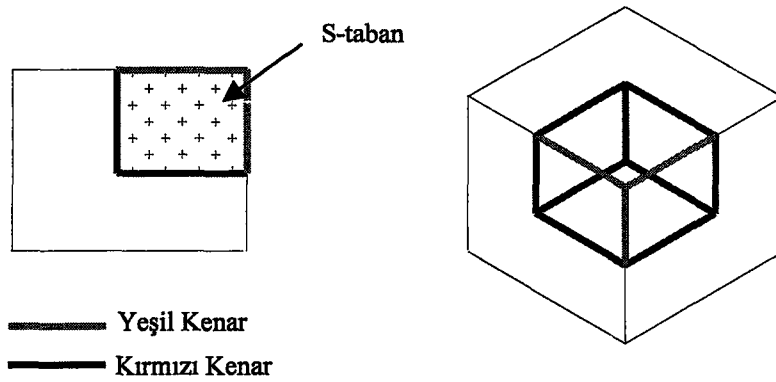
Şekil 3.14. Kademe unsurunun gösterimi.

Düz Kanal Unsuru: Bu unsur dikdörtgen prizma şeklindedir (Şekil 3.15). Unsurda üç yüzey cisim tarafında, diğer üç yüzey ise boşluk tarafındadır. Boşluk 2 yüzey karşılıklıdır. Dolayısıyla kesici bir yüzeyden girip diğer yüzeyden çıkabilir. Cisim tarafındaki karşılıklı 2 yüzeyin bulunması, kullanılacak kesicinin çapını sınırlar. Süpürme taban (S-Taban) boş ve dolu olmak üzere iki değişik şekilde ortaya çıkabilir.



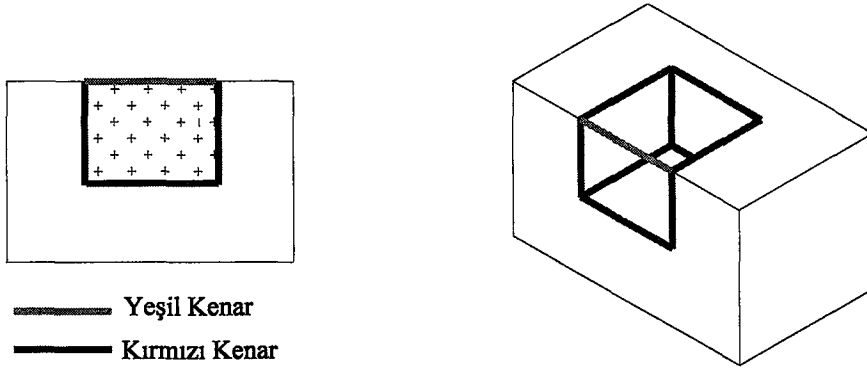
Şekil 3.15. Düz Kanal unsurunun gösterimi.

Kapalı Kademe Unsuru: Unsurun şekli dikdörtgen prizmadır (Şekil 3.16). 3 yüzeyi cisim tarafında, diğer 3 yüzeyi ise boşluk tarafındadır. Unsur kapalı olduğu için kesici bir uçtan girip diğer uçtan çıkamaz. Temel görünüşte unsurun S-Tabanı tek bir şekilde görülür ve doludur.



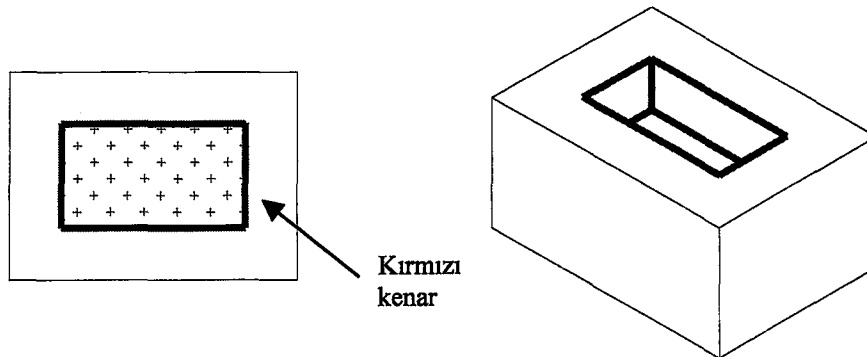
Şekil 3.16. Kapalı Kademe unsurunun gösterimi.

Arkası Kapalı Düz Kanal Unsuru: Unsurun şekli dikdörtgen prizmadır. 4 yüzeyi cisim tarafında, 2 yüzeyi boşluk tarafındadır. Unsur kapalı olduğu için kesici bir uçtan girip diğer uçtan çıkamaz. Temel görünüşte unsurun S-tabanı bir şekilde görülür ve doludur (Şekil 3.17).

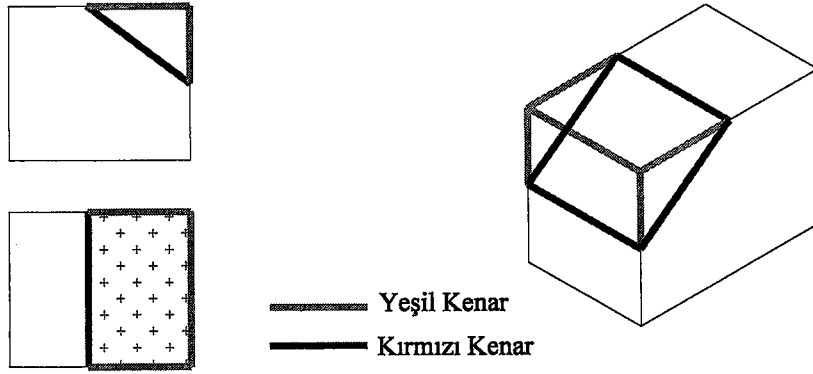


Şekil 3.17. Kapalı Kanal unsurunun gösterimi.

Cep Unsuru: Unsurun şekli dikdörtgen prizmadır. 5 yüzeyi cisim tarafında, 1 yüzeyi boşluk tarafındadır. Unsur kapalıdır. Temel görünüşte unsurun S-Taban şekli bir şekilde görülür ve doludur (Şekil 3.18).

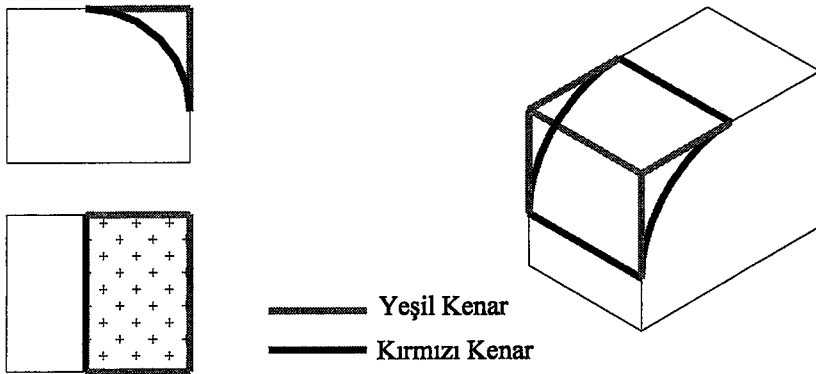


Şekil 3.18. Cep unsurunun gösterimi.



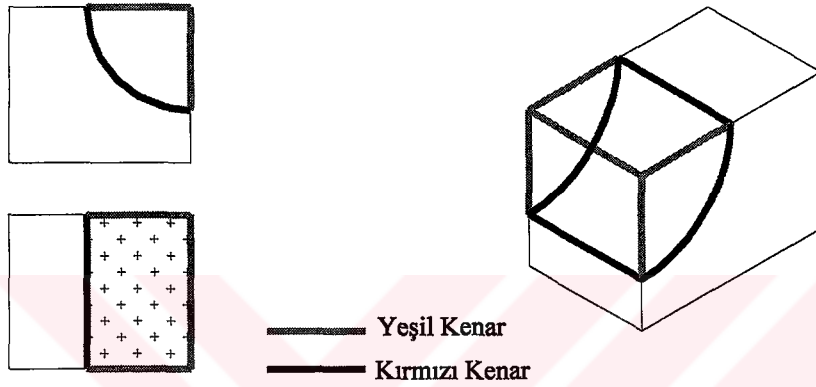
Şekil 3.20. Pah (Açılı yüzey) unsurunun gösterimi.

Dış Bükey Kavis Unsuru: Burada unsur, parçanın kenarları dış bükey olacak şekilde yuvarlatma ile oluşturulur. Unsurun taban alanı dış bükey kavisli yüzeydir. Unsurda 5 tane yüzey vardır. Bunların 1 tanesi cisim tarafında ve yuvarlaktır 4 tanesi ise boşluk tarafında ve düzdür (Şekil 3.21).



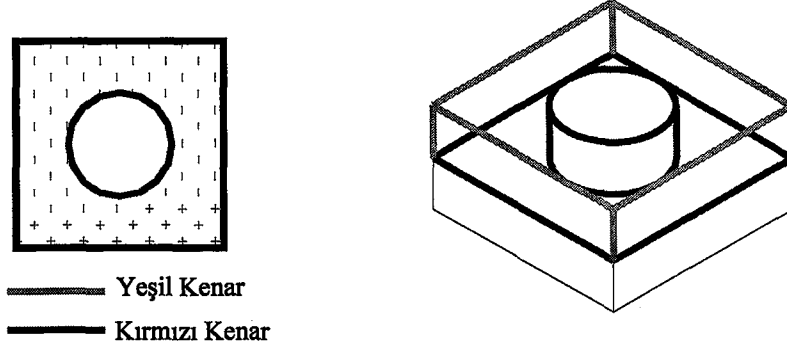
Şekil 3.21. Dış bükey kavis unsurunun gösterimi.

İç Bükey Kavis Unsuru: Bu unsurda parçanın köşesi kavisli iç bükey olacak şekilde yuvarlatılır. Unsurun aynı şekilde 5 tane yüzeyi vardır. Bunlardan iç bükey yüzey cisim tarafında, diğerleri ise boşluk tarafında bulunur. Unsurun S-Tabanı iki değişik şekilde görülür (Şekil 3.22).



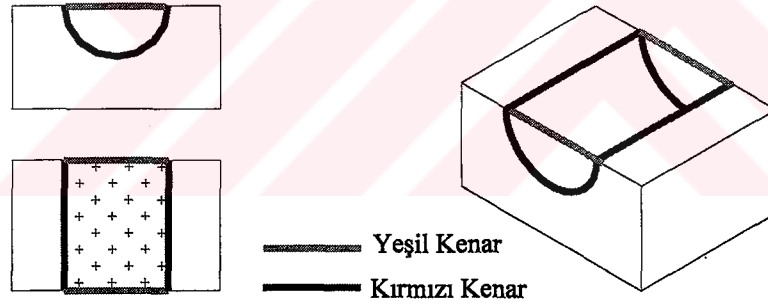
Şekil 3.22. İç bükey kavis unsurunun gösterimi.

Negatif Ada Unsuru: Bu unsur, ada şeklindeki bir çıkıntının dış kısmındaki negatif hacimden oluşur. Unsurun içerisinde bulunan adanın şekline göre de adlandırılabilir. Dairesel veya dikdörtgen negatif ada gibi. Unsurun içindeki ada kırmızı renklidir ve kesicinin o alana girmesini engeller. Unsurun dış duvarları değişik şekilde olabilir. Bunların bir kısmı cisim tarafında bulunabilir ve bu dış duvarlar kırmızı renkte olabilir (Şekil 3.23).



Şekil 3.23. Negatif Ada unsurunun gösterimi.

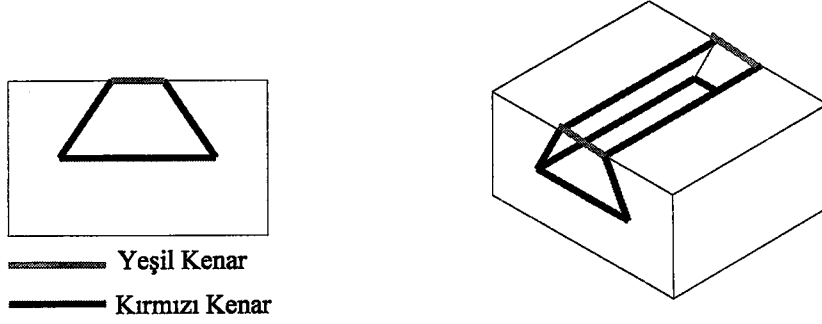
İç Bükey Yarım Silindir Unsuru: Bu unsur yarım silindir şeklindedir. Unsurun kavisli yüzeyi cisim tarafındadır ve S-tabanı iki değişik şekilde görülebilir (Şekil 3.24).



Şekil 3.24. İç Bükey Yarım Silindir unsurunun gösterimi.

Kırlangıç Kuyruğu Kanal Unsuru: Bu unsurun şekli kanal unsurun karşılıklı iki dik kenarının içeri doğru açılı durmasıyla oluşur. Algoritma unsuru ancak taban profilden yakaladığında bulabilmektedir. Unsura üstten bakıldığında,

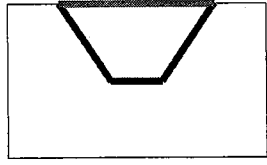
süpürülecek taban açıkça görülmediğinden unsuru bulamamaktadır. S-taban boş veya dolu bulunabilir. Eğer boş ise unsur tam delik olacaktır (Şekil 3.25).



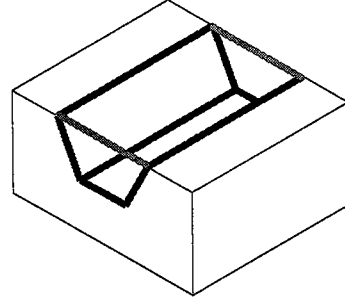
Şekil 3.25. Kırlangıç Kuyruğu Kanal unsurunun gösterimi.

V-Kanal Unsuru: Bu unsur kare kanalın dik kenarlarının dışarı doğru açılı durmasıyla oluşur. Algoritma bu unsuru ancak taban profilden gördüğünde bulabilmektedir. Unsura yukarıdan bakıldığında 3 tane solid alan olduğundan bu durumda 3 tane ayrı hacim oluşturmaktadır. Bunların ikisi düz pah, biri ise kare kanal tipinde olmaktadır. S-taban boş veya dolu bulunabilir. Eğer boş ise unsur tam delik, dolu ise kör delik olacaktır (Şekil 3.26).

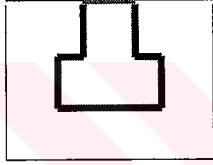
T-Kanal Unsuru: Bu unsurun taban profili T şeklindedir. Algoritma bu unsuru ancak taban profilden gördüğünde bulabilmektedir. Üstten bakıldığında süpürme taban açık olarak görülmemektedir. S-taban boş veya dolu bulunabilir. Buna göre kör veya tam delik olur (Şekil 3.27).



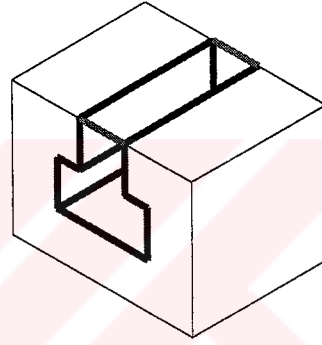
— Yeşil Kenar
— Kırmızı Kenar



Şekil 3.26. V-Kanal unsurunun gösterimi.

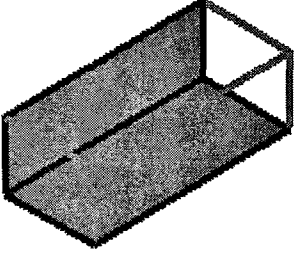


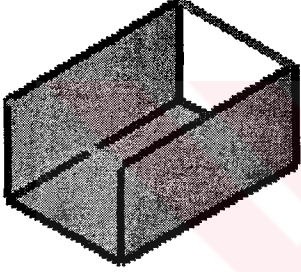


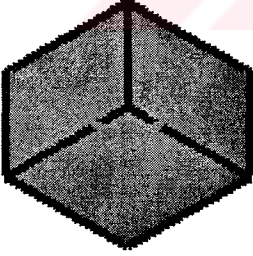
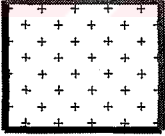
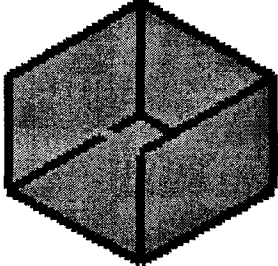
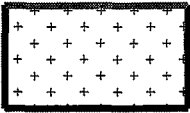


— Yeşil Kenar
— Kırmızı Kenar

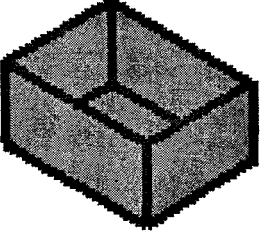
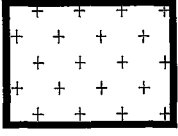
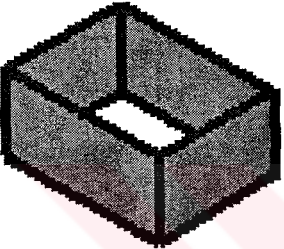

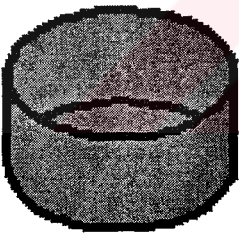
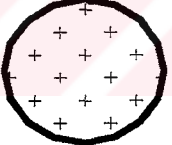
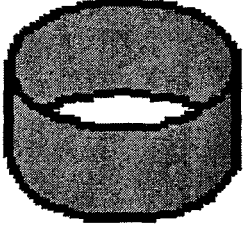
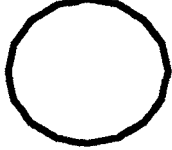


Şekil 3.27. T-Kanal unsurunun gösterimi.

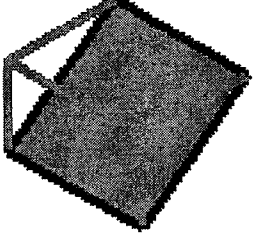

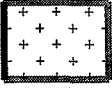
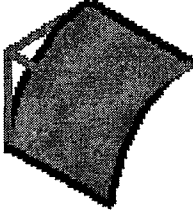

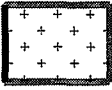
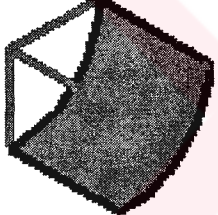

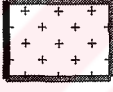
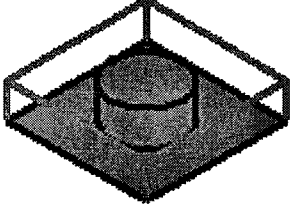
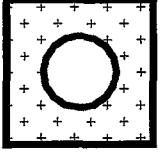
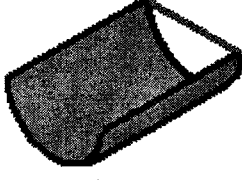

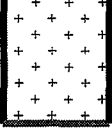
Çizelge 3.2. Unsur kütüphanesinde bulunan unsurların özellikleri (—— yeşil çizgi, ——— kırmızı çizgi).

No	Unsurun Adı, 3D'lu Şekli	Unsurun S-Tabanları	S-Taban Tipi	S-Taban Boş/Dolu
1	<p>Kademe</p> 		Düz	Boş
			Düz	Dolu
2	<p>Düz Kanal</p> 		Düz	Boş
			Düz	Dolu
3	<p>Kapalı Kademe</p> 		Düz	Dolu
4	<p>Sonu Kapalı Düz Kanal</p> 		Düz	Dolu

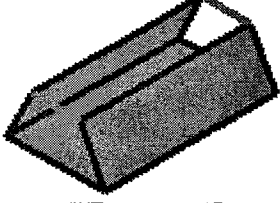

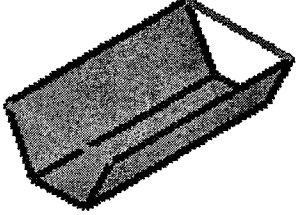

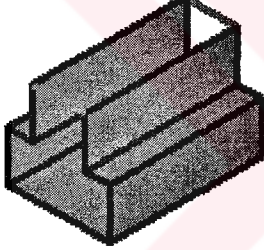

Çizelge 3.2. (devam)

5	<p>Cep</p> 		Düz	Dolu
6	<p>Prizmatik Delik</p> 		Düz	Boş
7	<p>Silindirik Kör Delik</p> 		Düz	Dolu
8	<p>Silindirik Tam Delik</p> 		Düz	Boş

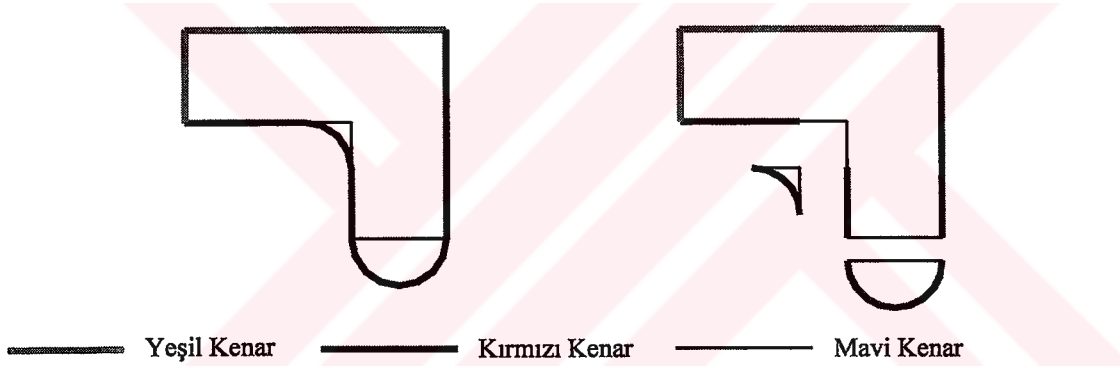
Çizelge 3.2. (devam)

9	<p>Düz Pah (Açılı Yüzey)</p> 		Düz	Boş/Dolu
			Eğik	Dolu
10	<p>Dış Bükey Kavisli Yüzey</p> 		Düz	Boş/Dolu
			Dışbükey Radüslü	Dolu
11	<p>İç Bükey Kavisli Yüzey</p> 		Düz	Boş/Dolu
			İçbükey	Dolu
12	<p>Ada Negatifi</p> 		Düz	Dolu
13	<p>İç Bükey Yarım Silindir</p> 		Düz	Boş/Dolu
			Radüslü	Dolu

Çizelge 3.2. (devam)

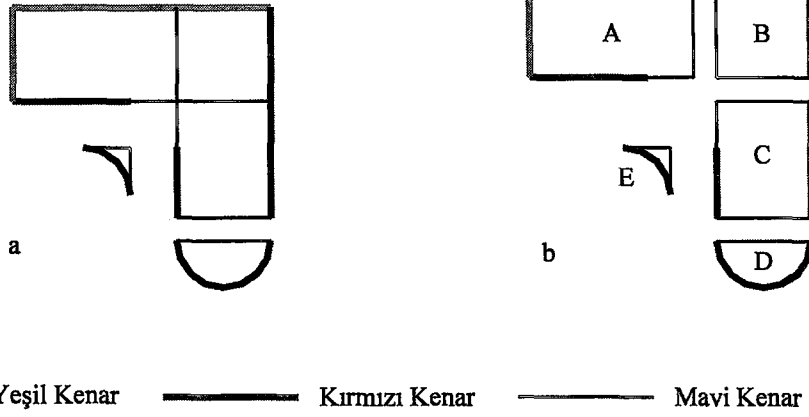
14	Kırlangıç Kuyruğu Kanal 		Düz	Boş/Dolu
15	V-Kanal 		Düz	Boş/Dolu
16	T-Kanal 		Düz	Boş/Dolu

1) Süpürme tabanın içerisinde yaylar bulunuyorsa bunların oluşturduğu unsurların tespit edilip, Süpürme tabandan çıkarılması gerekir. Bunun için yayların uç noktalarından (Algoritmada en büyük yay 90 derecedir) koordinat eksenlerine paralel olacak şekilde dik kenarlar oluşturulur. Eğer dik kenarlar süpürme tabanın içerisinde kalırsa, bu dik kenarlarla birlikte yaylar çıkarılır ve ayrı bir unsur olarak tespit edilir. Çıkarılan unsurun oluşturulan kenarları mavi olarak belirlenir. Oluşturulan dik kenarlar süpürme tabanın dışında kalırsa o zaman yayların uç noktalarıyla merkez arasında mavi çizgiler oluşturulur ve unsur bu çizgilerden ayrılarak çıkarılır (Şekil 3.29).



Şekil 3.29. Süpürme yaylarının oluşturduğu unsurların çıkarımı.

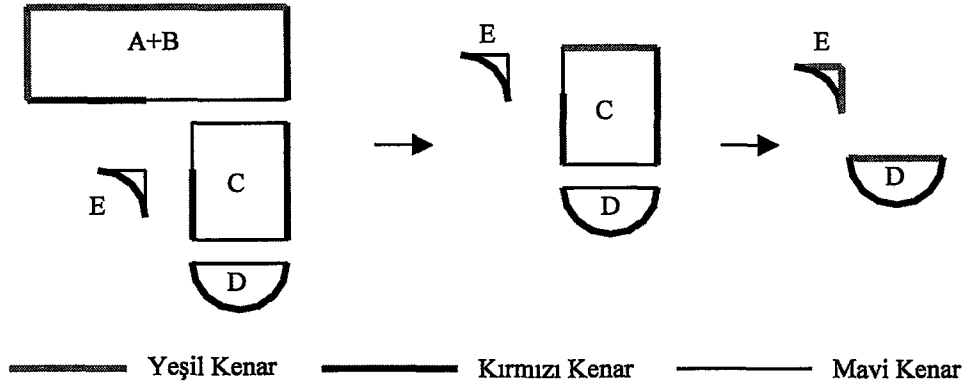
2) Yaylar ve ona bağlı olan çizgiler çıkarıldıktan sonra süpürme taban içinde geriye kalan çizgilerden kırmızı veya mavi çizgiler, süpürme tabana ait başka bir çizgi ile kesişene kadar uzatılır. Uzatılan bu çizgiler mavi olarak belirlenir. Daha sonra süpürme taban içerisindeki bütün döngülerin her biri ayrı bir alan olarak birbirinden ayrılır (Şekil 3.30).



Şekil 3.30. Kırmızı ve mavi kenarların uzatılması ve alanların bölünmesi.

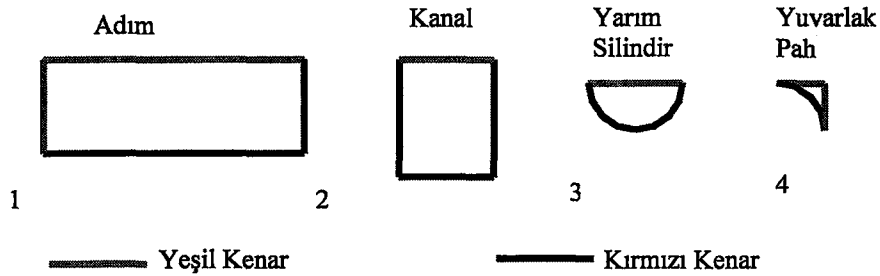
- 3) Ayrışma sonunda karşımıza unsur tipi daha kolay tespit edilebilecek alanlar çıkar. En büyük alana sahip ve yeşil kenar bulunduran alan temel bir alan olarak alınır. Bu alan süpürme taban (S-taban) olarak alınıp, unsur tipi tespit edilir. Eğer bir unsur çıkıyorsa, mavi kenarlı komşu alanları araştırılır. Bu komşu alanlarla mavi kenarlardan sırayla birleştirilerek yeniden alan oluşturulur ve unsur tipi araştırılır. Bu araştırmaya yeni bir unsur tipi bulamayana kadar devam edilir. Eğer yeni bir unsur tipi bulunmazsa, birleştirilen bu alanlar yeni bir unsurun süpürme tabanı olarak alınıp, unsur hacmi oluşturulur ve işlem sırası tespit edilerek kaydedilir ve diğer alanların içerisinde çıkarılır. Daha sonra geriye kalan alanlar incelemeye alınır. Geriye kalan alanların içinde çıkarılan unsura komşu olan mavi renkli kenarlar yeşile dönüştürülür. Tekrar yeşil kenarlı en büyük alan, temel alan olarak alınıp yeniden araştırmaya başlanır. Böylece bütün alanlar birleştirme ve çıkarma yoluyla oluşturulacak unsurların süpürme tabanları tespit edilir (Şekil 3.31). Burada yeşil kenarlı en büyük alanı ilk olarak ele almak aynı

düzlemde komşu birçok unsurun arasındaki işlem sıranı belirleme bakımından önemlidir.



Şekil 3.31. Birleştirme ve çıkarma yoluyla yeni unsurların elde edilmesi.

4) Oluşturulan unsur süpürme tabanlarının üzerinde kalan mavi çizgiler kırmızıya dönüştürülüp, unsur işlem sırası ve tipiyle birlikte dosyaya kaydedilir (Şekil 3.32). Daha sonraki aşamada bu bilgiler kullanılarak unsurun 3B'lu şekli oluşturulur.



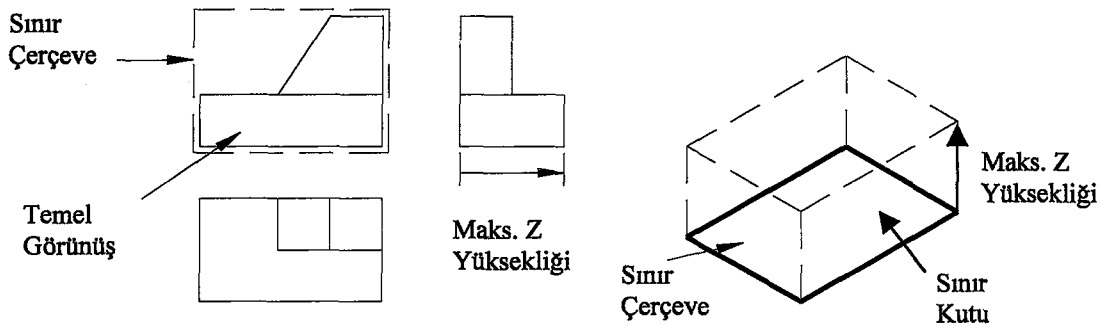
Şekil 3.32. Unsur tipi ve işlem sırası belirlenmiş süpürme tabanlar.

3.1.6. 3B'lu Cismin Oluşturulması

3.1.6.1. Sınır Kutunun Oluşturulması

Sınır kutu cismi içine alan en küçük hacimdir. Bu hacim pratikte cismi işlemek için ele alınan ham malzemenin kendisidir. Bu hacimden işleme unsurları çıkarıldığında en sona kalan hacim cismin 3B'lu hacmi olur.

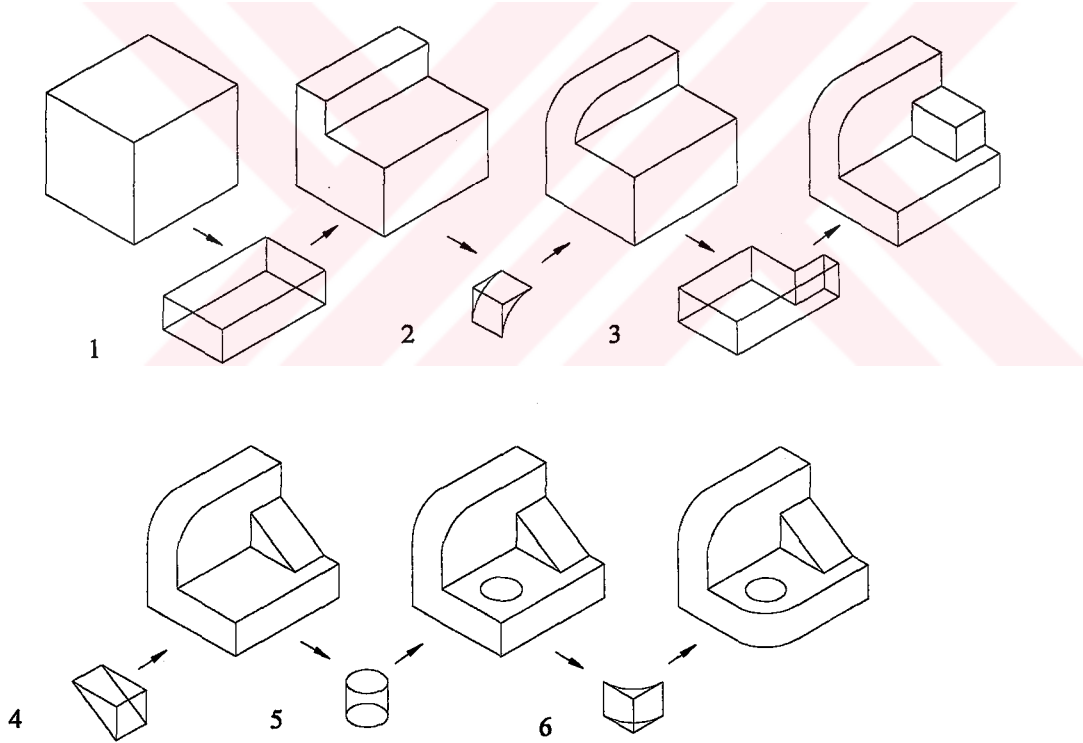
Sınır kutunun oluşturulması için öncelikle tabana ihtiyaç vardır. Taban olarak temel görünüşü içine alan “sınır çerçeve” kullanılmaktadır (Şekil 3.33). Tabanı süpürmek için gerekli yükseklik ise, diğer görünüşlerdeki kullanılmayan eksenin yüksekliğidir. Örneğin temel görünüş olarak ön görünüşü ele alırsak, ön görünüşü içine alan en küçük çerçeve sınır kutunun taban alanı olarak kullanılır. Bu taban alanı süpürmek için gerekli yükseklik ise, diğer görünüşlerden alınacak en büyük Z mesafesidir. Sınır kutu sonuçta cismin kendisi olacağı için renk olarak kırmızı kullanılır.



Şekil 3.33. Sınır kutunun elde edilmesi.

3.1.6.2. 3B'lu Cismin Oluşturulması

Cisim baştan dikdörtgen prizma şeklinde ham malzeme olarak kabul edilmektedir. Yani sınır kutunun kendisi olmaktadır. Bu ham malzemeden işlenecek negatif unsurlar çıkarıldığında bir diğer deyişle sınır kutudan unsur hacimleri çıkarıldığında, geriye kalan hacim ise 3B'lu cismin kendisi olmaktadır (Şekil 3.34). Burada hacimlerin sınır kutudan çıkarılma sırası, daha önceki bölümlerde bahsedilen tabanların yükseklik sırasına göre olmaktadır. Unsur hacimleri de bu sıraya göre oluşturulmaktadır.



Şekil 3.34. 3B'lu cismin oluşturulması.

3.1.7. Cismin Geri Kontrolü ve Eksik Unsurların Bulunması

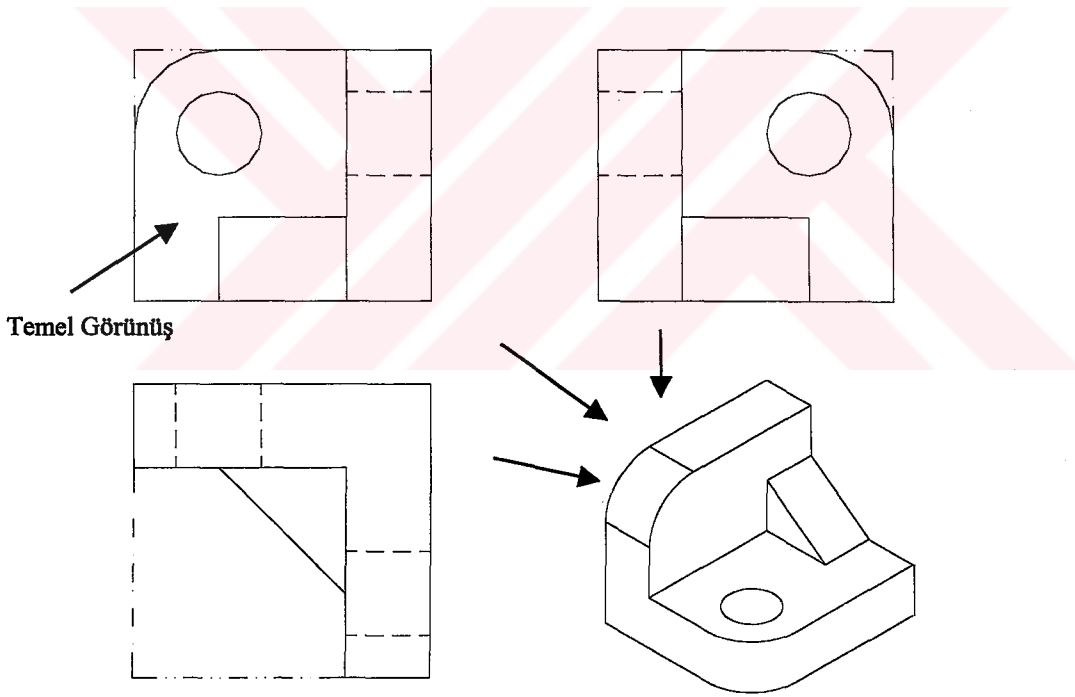
Algoritmanın ilk olarak arařtırdığı unsurlar, temel görünüşte tabanları sürekli çizgi şeklinde görülen unsurlardır. Oysa cisim üzerindeki bazı unsurların tabanları diđer görünüşlerde de sürekli çizgili alan şeklinde görülebilmektedir. Hatta bunların dışında üç görünüşte de tabanları sürekli çizgili alan şeklinde olmayıp, kesikli çizgi şeklinde olabilen unsurlar bulunabilmektedir. Burada kullanılan algoritma üç görünüşte de tabanları sürekli çizgi şeklinde bulunan unsurları arařtırmaktadır. Eğer unsurun tabanı kesikli çizgi şeklinde görülüyorsa bu unsurları bulamamaktadır. Bu durumu pratik hayatla karşılařtırsak, bu tip unsurlar parçanın tablaya bağlandığında parçanın altında kalan unsurlar olmaktadır. Şayet algoritmaya 3 görünüş yerine bu unsurların tabanlarının sürekli çizgili alan şeklinde görüleceđi diđer görünüşlerde (sađ, arka, alt) eklenirse, o zaman algoritma bu durumdaki unsurları da bulabilmektedir.

Program ilk temel görünüşe algoritmayı uyguladıktan sonra 3B'lu cisimi oluřtırmaktadır (Şekil 3.33). Fakat bu oluřturulan cisimden bütün negatif unsurların çıkarılıp çıkarılmadıđından emin olmak gerekmektedir. Bu amaçla cismin geri kontrolü yapılmalıdır.

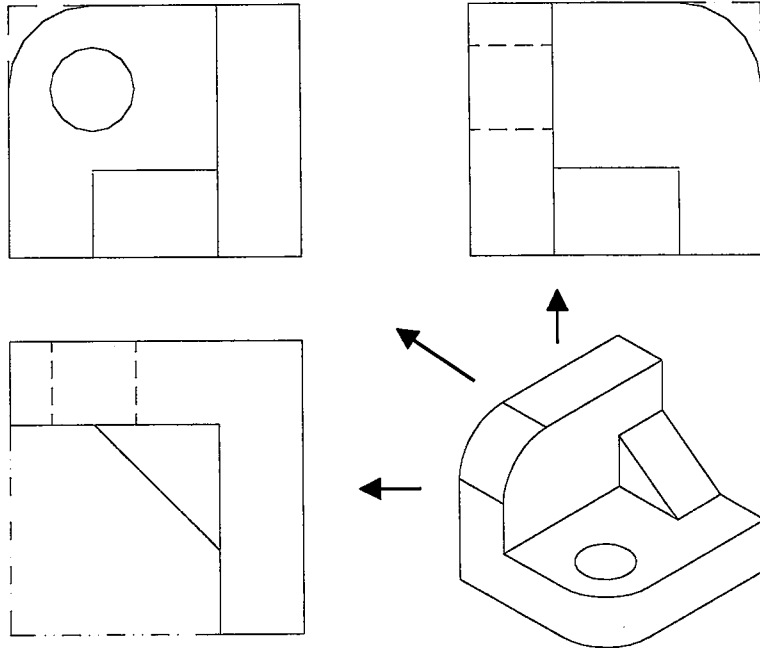
Oluřturulan 3B'lu cismin 2B'lu olarak üç temel görünüşü tekrar oluřturulur (kontrol görünüşleri) (Şekil 3.34). Bu görünüşler orjinal görünüşlerle karşılařtırılır. Eğer orjinal çizimde, oluřturulan görünüştekilerden daha fazla çizgiler bulunursa, cismin üzerinde çıkarılmamış hacim bulunuyor ve henüz unsur çıkarım işlemini tamamlanmamış demektir.

Bu amaçla orijinal üç görünüşün hangisinde eksik sürekli çizgili alan varsa, o görünüş temel görünüş olarak kabul edilip tekrar algoritma uygulanır. Bu uygulamada sınır kutu tekrar baştan prizma şeklinde alınır ve unsurlar birer birer çıkarılır. Çıkarılan bütün unsurlar atılıp içerinden sadece araştırılan alanın oluşturduğu unsur alınır. Bu unsur, daha önceden oluşturulan hacimden çıkarılır (Şekil 3.35). Eğer bu aşamada temel görünüşte başka alanlarda varsa, onların oluşturduğu hacimler de cisimden çıkarılır.

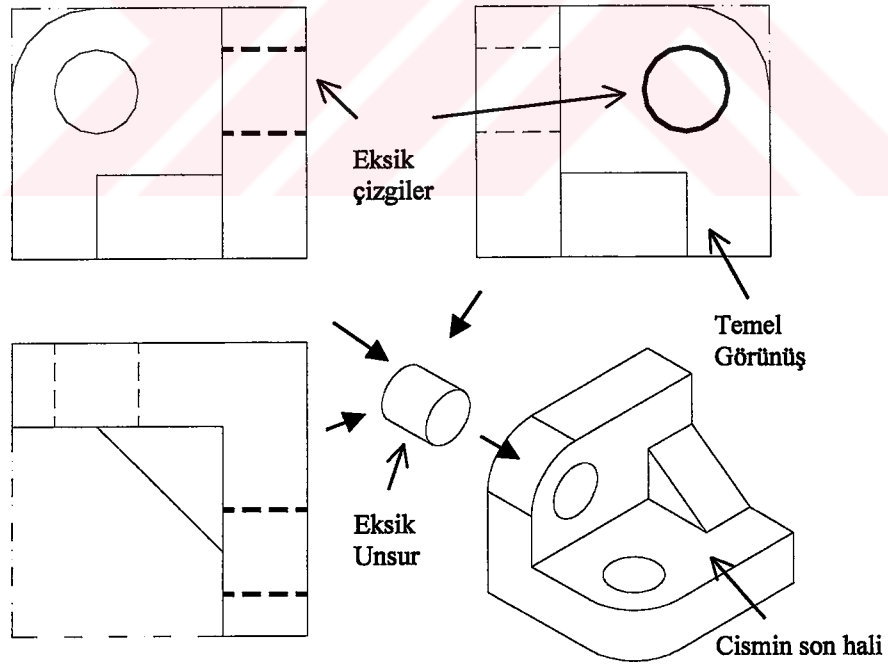
Ardından tekrar geri kontrol yapılır. Başka, sürekli çizgili eksik alan bulunursa, o görünüşten elde edilen unsurlar da cisimden çıkarılır.



Şekil 3.35. Orijinal çizimden elde edilen 3B'lu cisim.



Şekil 3.36. Elde edilen 3B'lu cisimden tekrar kontrol görüntülerinin çıkarımı.



Şekil 3.37. Temel görünüşün sol görünüş alınmasıyla bulunan eksik unsur.

4. TARTIŞMA VE SONUÇ

Makine parçalarının en yaygın gösterim şekli olan 2B'lu teknik resimler, doğrudan bilgisayar tabanlı otomatik üretim işlemlerinde kullanılmadığı için, 2B'lu çizimlerden parçanın 3B'lu şeklinin oluşturulması ve üzerindeki unsurların tespit edilmesi büyük önem arz etmektedir.

Bu tez çalışmasında parçanın 3 temel görünüşü ele alınmış ve bu görüntülerden parça üzerindeki negatif unsurlar çıkarılıp, parçanın 3B'lu modeli kullanıcıya sunulmuştur. Görünüşler üzerinde herhangi başka bir bilgi ve detay (kesit görünüşü, yardımcı görünüş, ölçülendirme vs) bulunmamaktadır. Daha çok 2.5B'lu sayılabilecek ve BSD'li dik işleme merkezlerinde işlenebilecek türden parçalara uygulanabilmektedir. Bu parçaların üzerinde düz, eğik ve kavisli yüzeyler bulunabilmektedir. Parçanın alt kısmında kalan unsurlar çıkarılamamaktadır. Geliştirilen algoritma Visual Basic dilinde programlanıp geçerliliği ispatlanmıştır. 2B'lu çizimi program içerisinde gerçekleştirebilmek için programın ön kısmına bir 2B'lu çizim modülü eklenmiştir.

Burada ele alınan metotta öncelikle parça ham bir malzemeyi temsil eden bir prizma olarak ele alınmaktadır. Daha sonra malzemedен çıkarılacak unsurlar tespit edilmektedir. Çıkarılacak unsurları tespit etmek için taban, süpürme taban ve tavan olarak adlandırılan 3 tane yüzey kullanılmaktadır. Tabanların tespitinde ise “izdüşüm yöntemi” olarak adlandırılan yeni bir yöntem kullanılmıştır.

Tabanlar, unsurun alt sınırını, metod içinde oluşturulan Tavanlar ise unsurların üst yüzeyini, taban ve tavanlardan elde edilen Süpürme Tabanlar (S-taban) ise unsurun alt yüzeyini oluşturmaktadır. S-tabanlar daha sonra, tabanla tavan arasında süpürülerek unsurun hacmi oluşturulmaktadır.

d) Bu çalışmada aşağıdaki unsur özellikleri tespit edilmektedir:

- 1) 16 çeşit unsurun tipi ve şekli belirlenmektedir.
- 2) Bütün unsurlar negatif unsur olarak ele alınmaktadır.
- 3) Unsurlar üzerindeki bütün yüzeyler için dışarıdan erişimin mümkün olup olmadığı belirlenmektedir.
- 4) Unsurun tam/kör delik durumu tespit edilmektedir.
- 5) Unsur tabanının kapalı/açık olma durumu belirlenmektedir.
- 6) Unsurun taban geometrisinin şekli tespit edilmektedir.
- 7) Kendisinden önce işlenmesi gereken komşu negatif unsur varsa, onunla olan ortak yüzeyi belirlenmektedir.
- 8) Parça üzerindeki unsurların işleme sırası tespit edilmektedir.

e) Bu çalışmada cismi içine alan prizma, pratikte parçayı işlemek için ele aldığımız ham malzeme olarak görülmüş ve ona göre programda kullanılmıştır. Eğer prizmanın sınırları 2B'lu çizimde görüşlere eklenmezse, parçayı içine alan en küçük hacim olarak alınmaktadır. Prizmanın görüşleri, 2B'lu çizim esnasında görüşlerin dışına ek çizgiler çizilerek eklenmektedir.

f) Bu çalışmada oluşturulan 3B'lu modelin geri kontrolü yapılmaktadır. Üzerinde bulunamayan unsurlar varsa onlar geri kontrolle tespit edilmektedir.

g) Parça üzerinde karmaşık unsurların bulunduğu durumlarda, bu unsurları parçalara ayırıp, temel unsurlar cinsinden ifade edebilmektedir.

Algoritma bazı açılardan sınırlı kalmaktadır. Bunlar şu şekilde özetlenebilir.

- 1) Üç görünüş içinde kesikli çizgilerle görünen ve cismin arka tarafında kalan hacimleri çıkaramamaktadır.
- 2) Algoritma küresel ve toroid yüzeyleri de tanıyamamaktadır.

Bu algoritmayla ilgili olarak ileride şu çalışmalar yapılabilir.

- 1- 2B'lu çizimlere kesit görünüşleri ve yardımcı görünüşler eklenerek, cismin üzerindeki negatif unsurlar tespit eden ve 3B'lu hacmi oluşturan bir algoritma yapılabilir.
- 2- 2B'lu çizim içerisindeki kesikli görünmez çizgilerde tanıyan bir algoritma geliştirilerek, cismin arkasındaki unsurlar da tespit edilebilir. Gerekirse bunun için cismin 3 görünüşü yerine 6 görünüşü istenebilir.
- 3- Cismin üzerindeki negatif unsurlar tespit edildikten sonra, BSD'li tezgahlar için otomatik kesici yolunu çıkaran modül eklenerek, parça programı 2B'lu çizimden doğrudan tezgaha aktarılabilir.
- 4- Algoritmanın ön kısmına, tarayıcıdan okunan 2B'lu resimlerin üzerindeki çizgileri tanıyacak bir modül eklenerek, ve bu modülden cismin görüntüleri alınarak, daha önceden çizilmiş fakat kağıt üzerinde bulunan resimlerden de cismin unsurları ve 3B'lu hacmi oluşturulabilir.
- 5- Parçaların kamera ile çekilen dik görünüşleri, resim işleme programlarına aktarılıp çizgiler burada sayısallaştırıldıktan sonra, parçanın 3 görünüşü tespit edilebilir. Daha

sonra görünüřlerden parçanın üzerindeki unsurlar ve 3B'lu řekli çıkarılabilir. Bu konu üretim hatlarında robotların parçaları yakalamak için parçanın řekli hakkında bilgi sahibi olması gerektiđi durumlarda, yada üretimi istenen bir model parçanın imalatını yapabilmek için kullanılabilir.

Geliřtirilen algoritma birçok örnek parça üzerinde denenmiřtir. Bunlardan 2 tanesi Ek1 ve Ek2 de verilmiřtir. Burada ele alınan parçaların görünüřleri parçayı 3 görünüř içerisinde tam olarak anlatabilmektedir. Parçalar üzerinde düz, eğik ve kavisli yüzeyler vardır. Parçalar daha çok BSD dik işleme merkezlerinde işlenebilecek türden parçalardır. Parça tablaya bađlandığında bütün yüzeyler tezgahta üstten işlenebilecek řekildedir. Parçaların karakteristik řeklini veren ya da işlenecek yüzeylerin en iyi görüldüđü görünüř ön görünüř olarak çizilmiřtir.

KAYNAKLAR

1. R. Genesan, and V. Devarajan, CAD, **30**, 863(1998).
2. S. Meeran, and M.J. Pratt, CAD, **25**, 7(1993).
3. J. J. Shah and M.T. Rogers, **22**, 515(1988).
4. L.W. Tyan and V. Devarajan, CAD, **30**, 357(1998).
5. M. Idesawa, Bull. of the Japon Soc. of Mech. Engng., **16**, 216(1973).
6. G. Lafue, Comput. Graph., **2**, 103(1976).
7. G. Markowsky and M.A. Wesley, IBM J. of Res. & Dev..., **24**, 582(1980).
8. K. Preiss, Comput. in Ind., **2**, 133(1981).
9. M.A. Wesley and G. Markowsky, IBM J. of Res. & Dev., **25**, 934(1981).
10. B. Aldefeld, CAD, **15**, 59(1983).
11. H. Sakurai and D.C. Gossard, Comput. Graph., **17**, 243(1983).
12. H. Yoshiura, K. Fujimura and T.L. Kunii, IEEE Comput. Magz., 32(1984).
13. Ho Bin, CAD , **18**, 147(1986).
14. R. Lequette, CAD, **20**, 171(1988).
15. Z. Chen and D.B.Perng, Pattern Recogn., **21**, 439(1988).
16. D. Dutta and Y.L. Srinivas, CAD, **24**, 149(1992).
17. C.F. You, S.S. Yang, Comput. & Graph., **20**, 275(1996).
18. H. Masuda and M. Numao, CAD, **29**, 177(1997).
19. S.S.P. Shum, W.S. Lau, M.M.F. Yuen and K.M. Yu, Comput. & Graph., **21**,
787(1997).
20. M.H. Kuo, CAD, **30**, 517(1998).
21. B.S. Shin and Y.G. Shin, CAD, **30**, 63(1998).

22. M. Tanaka, K. Iwama, A. Hosoda and T. Watanabe, *CAD*, **30**, 37(1998).
23. C. Kim, M. Inoue and S. Nishihara, *J. of Information Processing*, **15**, 510(1992).
24. P. Bottoni, U. Cugini, P. Mussio, C. Papetti and M. Protti, *Machine Vision and Appl.*, **8**, 326(1995).
25. D. Dori and K. Tombre, *CAD*, **27**, 243(1995).
26. S.K. Ghosh, Y.B. Li, A.K. Munns, X.C. Wang, *J. of Materials Processing Technology*, **54**, 211(1995).
27. D.B. Lysak, P.M. Devaux and R. Katuri, *Pattern Recognition*, **28**, 393(1995).
28. D. Dori, M. Weiss, *Engng. Appl. of Artificial Intelligence*, **9**, 53(1996).
29. U.G. Gujar and I.V. Nagendra, *Comput. & Graph.*, **13**, 505(1989).
30. J.T. Gau, *Automatic Construction of B-rep Solid Models from 2-D Engng. Drawings*, MS Thesis, National Taiwan University, Taiwan, R.O.C. 1991.
31. C.Y. Lung and K.C. Fan, *Proceed. of The Ninth National Conf. of The Chinese Society of Mechanical Engineers*, Kaohsiung, Taiwan, R.O.C., 203(1992).
32. Z. Chen, D.B. Perng, C.J. Chen and C.S. Wu, *Int. J. of Comput. Integrated Manufac.*, **5**, 2(1992).
33. Q.W. Yan, C.L.P. Chen and Z.H. Tang, *CAD*, **26**, 699(1994).
34. C.F. You and S.S. Yang *Proceed. of Automation'95*, Taiwan, R.O.C., **2**, 1020(1995).
35. H.R. Börklü, *Mühendis ve Makine*, **39**, 35(1997).
36. V. Suharitdamrong and S. Motavalli, *Ind. Engng. Res. Conf. Proc.*, IIE, Norcross, GA, USA, 183(1995).
37. U. Cugini, B. Falcidieno, F. Giannini, P. Mussio and M. Protti, *Proc. of AI in Design'91 Conf.*, Edinburgh, UK, 407(1991).
38. S. Aomura, H. Homma and M. Ikawa, *Proc. of Int. Conf. On Engng. Design*

- ICED'95, Praha, Czech Republic, **4**, 1513(1995).
39. T.S. Hwang and D.G. Ullman, Proc. of ASME Con. On Comp. In Engng., **1**, 67(1994).
40. C.H. Liu, D.B. Perng, Z. Chen, Comput. Ind. Engng., **26**, 689(1994).
41. E. Aslan, A Postprocessor Development with Expert System Approach, PhD Thesis, Gazi University, Institute of Science and Technology, Ankara, Turkey, 1995.
42. E. Aslan, U. Şeker, Alpdemir N., J.of Engineering and Environmental Science, **23**, 339(1999).
43. E. Aslan, U. Şeker, International Congress-Gear Transmissions, Sofia, Bulgaria, 157(1995).
44. E. Aslan, M.N. Alpdemir, 7. International Machine Design and Production Conference, METU, Ankara, 455(1996).
45. M. Flasiński, CAD, **27**, 403(1995).
46. B.K. Choi, M.M. Barash, CAD, **16**, 81(1984).
47. B.K. Choi and M.M. Barash, CAD, **17**, 162(1985).
48. S.M. Staley and D.C. Anderson, ASAIE Comput. Mech. Engng., **2**, 61(1983).
49. R.K. Li, Int. J. Comput. Integr. Manufact., **1**, 1451(1988).
50. H.P. Wang, Int. J. Prod. Res., **26**, 119(1988).
51. P.D. Stefano, CAD, **29**, 761(1997).
52. D.T. Pham and S.S. Dimov, Proc. Instn. Mech. Engrs. Part B, **212**, 13(1998).
53. V. Allada and S. Anand, Int. J. of Prod. Res., **34**, 1791(1996).
54. M.R. Henderson, Extraction of feature information from three dimensional CAD data, Ph.D. Thesis, Purdue University, West Lafayette, IN, 1984.
55. M.R. Henderson and D.C. Anderson, Comput. Ind., **5**, 329(1984).

56. M.R. Henderson and G.J. Chang, In Procc. of the 1988 ASME Int. Comput. in Engng. Conf. and Exhibition, San Francisco, CA, **1**, 529(1988).
57. S. Gao and J.J. Shah, CAD, **30**, 727(1998).
58. Q. Ji, M. M. Marefat, CAD, **27**, 435(1995).
59. I.K. Kyprianou, Shape classification in Comput.-aided-design, Ph.D. Dissertation, University of Cambridge, July, 1980.
60. S. Joshi and T.C. Chang, CAD, **20**, 58(1988).
61. S. Joshi, N.N. Vissa and T.C. Chang, Int. J. Prod. Res., **26**, 863(1988).
62. S. Joshi and T.C. Chang, **1**, 1(1990).
63. H. Sakurai and D.C. Gossard, IEEE Comput. Graph. & Appl., **10**, 22(1990).
64. P. Gavankar and M.R. Henderson, CAD, **22**, 442(1990).
65. F. Pariente and Y.S. Kim, CAD, **28**, 589(1996).
66. X. Xu and S. Hinduja, CAD, **30**, 503(1998).
67. E. Wang and Y.S. Kim, Comput. & Ind. Engng., **27**, 123(1994).
68. Y.S. Kim, In Procc. of the 1991 ASME Comput. and Engng. Conf., New York, **61**(1991).
69. Y.S. Kim. CAD, **24**, 461(1992).
70. Y.S. Kim, In Procc. of the 1992 ASME Comput. in Engng. Conf., USA, **233**(1992).
71. T.C. Woo and J.M. Hammer, Proc. 9 th CIRP Conf., **247**(1977).
72. T.C. Woo, In Proceed. of the Conf. CAD/CAM Tech. in Mechanical Engng., Cambridge, **76**(1982).
73. T.C. Woo, Comput., **44**(1984).
74. J.C.E. Ferreira and S. Hinduja, CAD, **22**, 41(1990).
75. R. Bidarra, K.J.D. Kraker, Bronsvoort, W.F., CAD, **30**, 301(1998).

76. Y.J. Tseng and S.B. Joshi, CAD, **26**, 667(1994).
77. H. Sakurai, CAD, **27**, 833(1995).
78. P. Dave, H. Sakurai, Proc. of ASME Conf. On Comp. In Engng., 553(1995).
79. M.M. Marefat and R.L. Kashyap, IEEE Trans. Pattern Anal. Mach. Intel., **12**, 949(1990).
80. J. Vandenbrande, A.A.G. Requicha, IEEE Trans. Pattern Anal. Mach. Intel., **15**, 1269(1993).
81. A.A.G. Requicha, Commun. ACM, **39**, 71(1996).
82. V. Jagannathan, R. Dodhiawala and L.S. Baum, Academic Press, Inc., Inter. Ed. 1989.
83. J. Pearly, Morgan Kaufmann, Palo, Alto, CA, 1988.
84. K. Nezis and G. Vosniakos, CAD, **29**, 523(1997).
85. J.H. Han and A.G. Requicha, CAD, **29**, 393(1997).
86. C.A. Chang and C.Y. Tsai, Comput. in Ind., **34**, 27(1997).
87. S. Prabhakar and M.R. Henderson, CAD, **24**, 391(1992).
88. M.C. Wu, J.R. chen and S.R. Jen, Inter. J. of comp. Integrated Manufac., **7**, 261(1994).
89. C.C. Chan, G.W. Fisher, Proc. of the 1996 ASME Inter. Mech. Engng. Congress and Exposition, Atlanta, GA, 731(1996).
90. J. Dong and S. Vijayan, CAD, **29**, 475(1997).
91. J. Dong and S. Vijayan, CAD, **29**, 427(1997).
92. A. Trabelsi and S. Meeran, Int. J. of Prod. Res., **34**, 2701(1996).
93. H. Sakurai and P. Dave, CAD, **28**, 519(1996).
94. Y. Shen and J.J. Shah, Adv. Des. Autom. ASME, **1**, 575(1994).
95. J. Dong, H.R. Parsaei and T. Gornet, In Proceed. of the 2nd Ind. Engng Res.

- Conf., Institute of Ind. Engineers, Los Angeles, CA, 11(1993).
96. J. Dong, Design and implementation of a prototype feature-based automated process planning (FBAPP) system, Ph.D. Thesis, University of Louisville, 1993.
97. Y.J. Tseng and S.B. Joshi, *Int. J. of Production Res.*, **32**, 1063(1994).
98. M.A. Chamberlain, A. Joneja and T.C. Chang, *CAD*, **25**, 19(1993).
99. M.T. Wang, A generative reasoning methodology for Manufac., Ph.D. Thesis, Purdue University, IN, 1990.
100. M.T. Wang and T.C. Chang, In *Proceed. of the Manufac. Int. Conf.*, ASME, New York, **2**, 49(1990).
101. D.B. Perng, Z. Chen, D.E.R. Li, *CAD*, **22**, 285(1990).
102. C.M.Hoffmann and R.J. Rinyo, *CAD*, **30**, 321(1998).
103. F. Öztürk, *Makine Tasarım ve İmalat Dergisi*, **3**, 51(1996).
104. C.C.P. Chu and R. Gadh, *CAD*, **28**, 321(1996).
105. X. Chen and C.M. Hoffman, *CAD*, **27**, 905(1995).
106. J.C. Cavendish, *CAD*, **27**, 703(1995).
107. O.W. Salomons, F.J.A.M. Houten and H.J.J. Kals, *J. Manufact. Sys.* **12**, 113(1993).
108. N.N.Z., Gindy, *Int. J. Prod. Res.*, **27**, 2089(1989).
109. K. Case, J.X. Gao and N.N.Z. Gindy, *J. Engng. Manuf. I. Mech. E.*, **208**, 71(1994).
110. T. Laakko and M. Mantyla, In *ACM Siggraph 2nd Symposium on Solid Modelling and Appl.*, Montreal, 333(1993).
111. M. Ranta, M. Mantyl, Y. Umeda and T. Tomiyama, *CAD*, **28**, 371(1996).
112. A.A.G. Requicha and J.H. Vandenbrande, In *Artificial Intelligent: Implications for CIM*, New York, 301(1988).

113. A.A.G. Requicha and J.H. Vandenbrande, In *Comput. in Engng.*, ASME, Anaheim, CA, 47(1989).
114. J.J. Shah, M.T. Rogers, P.C. Sreevalsan, D. Hsiao, A. Mathew, A. Bhatnagar, A. Liou and D.W. Miller, In *ASME Comput. in Engng. Conf.(CIE)*, 233(1990).
115. C.T. Martino, B. Falcidieno, B. Giannini, B. Hassinger and J. Ovtcharova, *CAD*, **26**, 646(1994).
116. J.C.E. Floriani and E. Bruzzone, *CAD*, **21**, 602(1989).
117. C. Kim and P.J. Grady, *CAD*, **28**, 451(1996).
118. Z. Fu, *Geometric Reasoning For Feature-Based Design Using A Graph Grammar*, Ph.D. Thesis, University of Leeds, December 1991.
119. A.R.O. Guerra and S. Hinduja, *CAD*, **29**, 343(1997).
120. W.C. Regli, S.K. Gupta, D.S. Nau, *CAD*, **29**, 37(1997).
121. D.B. Perng and C.F. Chang, *CAD*, **29**, 687(1997).
122. J.H. Han, A.A.G. Requicha, *CAD*, **30**, 453(1998).
123. A. Joneja, *Comput. & Ind. Engng.*, **33**, 489(1997).
124. M.C. Wu and C.R. Liu, *CAD*, **28**, 603(1996).
125. I.H. Filiz, T. Dereli, Ş. Çetinkaya, 7th Int. Machine Design and Prod. Conf., METU, Ankara, 113(1996).
126. H. Suh and R.S. Ahluwalia, *CAD*, **27**, 627(1995).
127. R.K. Li, C.Y. Lin and H.H. Wu, *Int. J. of Prod. Res.*, **33**, 549(1995).
128. S.K. Gupta and D.S. Nau, *CAD*, **27**, 323(1995).
129. S.H. Chuang and S.F. Huang, *CAD*, **28**, 473(1995).
130. X. Chen and C.M. Hoffman, *CAD*, **27**, 695(1995).
131. D.W. Perng, C.T. Cheng, *Comput. & Ind. Engng.*, **26**, 423(1994).
132. M.C. Fields and D.C. Anderson, *CAD*, **26**, 803(1994).

133. C.L.P. Chen and C.L.P., LeClair, CAD, **26**, 59(1994).
134. L.C. Sheu, J.T. Lin, CAD, **25**, 333(1993).
135. D.H. Lentz and R. Sowerby, CAD, **25**, 421(1993).
136. W. Duan, J. Zhou, K. Lai, CAD, **25**, 29(1993).
137. J. Ovtcharova, G. Pahl and J. Rix., Comput. & Graph., **16**, 187(1992).
138. R. Gadh and F.B. Prinz, CAD, **24**, 583(1992).
139. J.C. Cavendish and S.P. Marin, IEEE Comput. Graph. & Appl., **7**, 20(1992).
140. J.J. Shah and A. Mathew, CAD, **23**, 282(1991).
141. J. J. Shah, CAD, **23**, 331(1991).
142. A.J.P. Gomes and J.C.G. Teixeira, Comput. & Graph., **15**, 217(1991).
143. J. Corney and D.E.R. Clark, CAD, **23**, 658(1991).
144. N. Brandli and M. Mittelstaedt, CAD, **21**, 87(1989).
145. J.J. Woodwark, CAD, **25**, 19(1988).
146. Y.C. Lee and K. Fu, IEEE Comput. Graph. & Appl., 20(1987).
147. T. Kurtay, H. Kocabaş, MAMKON'97 İ.T.Ü. Makine Fakültesi 1. Makine Müh. Kongresi, 268(1997).

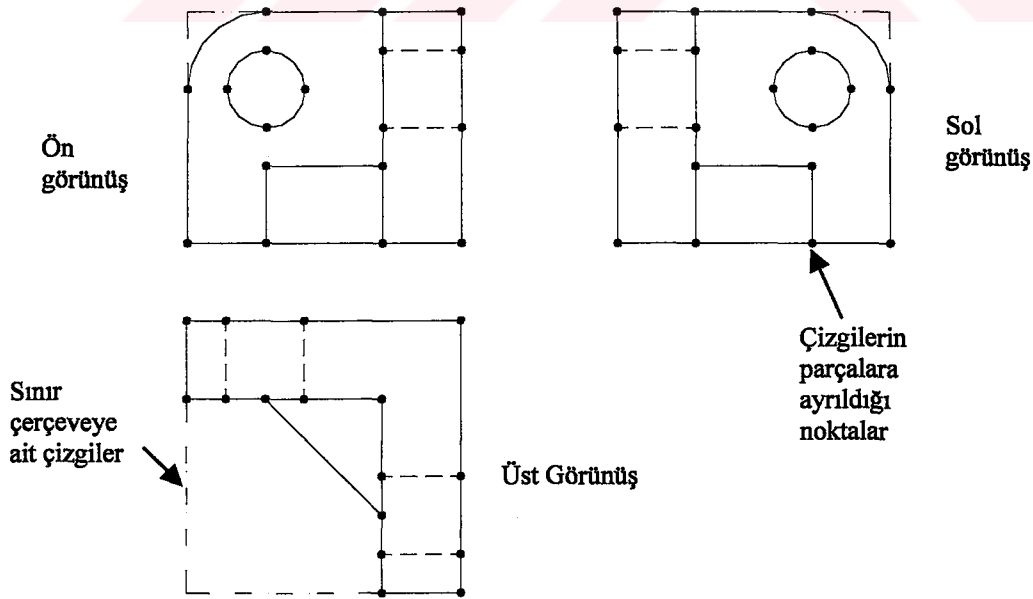
ÖZGEÇMİŞ

1968 yılında Çankırı'nın Kurşunlu ilçesinde doğan İbrahim Çayırođlu, İlk, Orta ve Lise öğrenimini Ankara'da tamamladı. 1987-1991 yılları arasında İ.T.Ü Sakarya Mühendislik Fakültesi Makina Mühendisliđi Bölümünde Lisans Öğrenimini bitirdi. 1991-1993 yılları arasında piyasada mühendis olarak çalıştıktan sonra, 1993 yılında Kırıkkale Üniversitesi Mühendislik Fakültesinde Araştırma Görevlisi olarak çalışmaya başladı. 1994-1996 yılları arasında Yüksek Lisans eğitimini Kırıkkale Üniversitesi Fen Bilimleri Enstitüsünde yaptı. 1996 yılında yine aynı Enstitüde Doktora öğrenimine başladı. Evli ve 2 çocuk babası olup halen Kırıkkale Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliđi Bölümünde çalışmaktadır.

EK 1. GENEL UYGULAMA PARÇASI

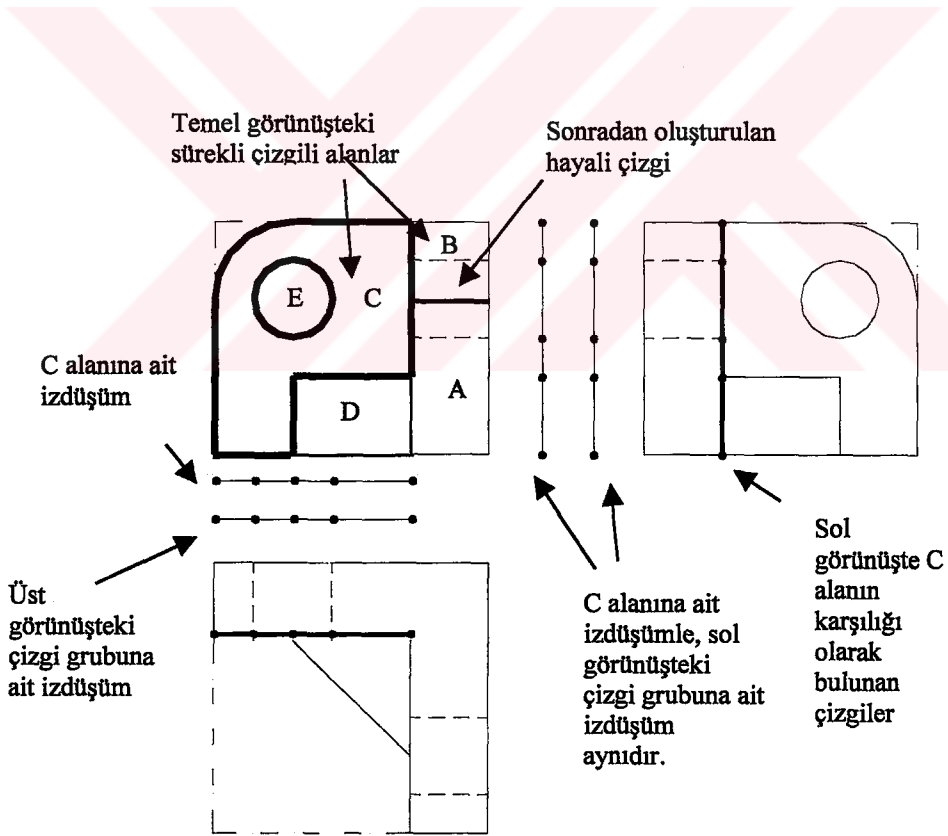
Ele alınan bu parça, pratikteki kullanımından daha çok, algortimanın özelliklerini yansıtabilecek düzeyde özelliklere sahip olması nedeniyle seçilmiştir. Algortimanın bütün aşamaları bu parça üzerinde maddeler halinde açıklamalı ve çizimli olarak aşağıda verilmiştir.

1. İlk olarak parçanın 3 temel görünüşü, program içerisindeki çizilmektedir (Şekil Ek 1.1).
2. Çizimdeki bütün çizgiler kesişen her noktadan parçalara ayrılır. Daire ve yaylar ise çeyrek noktalardan da parçalara ayrılır. Daha sonra database’de bir arada bulunan bütün çizgilerin görünüşlere taksimi yapılır. Taksimden sonra her görünüşün sınır çerçevesi oluşturularak, çerçeveye ait çizgiler kesişim yerlerinden parçalara ayrılarak görünüşlere eklenir (Şekil Ek 1.1).



Şekil Ek 1.1. Örnek parçanın üç görünüşü ve çizgilerin parçalara ayrıldığı noktalar.

3. Çizgiler görünürlere ayrıldıktan sonra her görünüş içerisindeki elemanların, düğüm-koordinat dosyası, kenar-düğüm dosyası, düğüm-kenar dosyası, köşe-kenar-koordinat dosyası ve alan-kenar dosyaları oluşturulup kaydedilir.
4. Daha sonra bütün görünümlerdeki çizgi, çizgi grupları ve sürekli çizgili alanların eksenler üzerine düşen izdüşümleri tespit edilir. Ardından temel görünüşteki sürekli çizgili alanların izdüşümleri ile diğer görünümlerdeki çizgi veya alanların izdüşümleri eşleştirilerek, sürekli çizgili alanların diğer görünümlerde karşılığı olan çizgiler veya alanlar tespit edilir. Eğer düz yüzeyle kavisli yüzeyin birleşim yerlerindeki hayali çizgiler çizilmemişse algoritma bu gibi yerlerdeki hayali çizgileri otomatik olarak oluşturmaktadır (Şekil Ek 1.2).

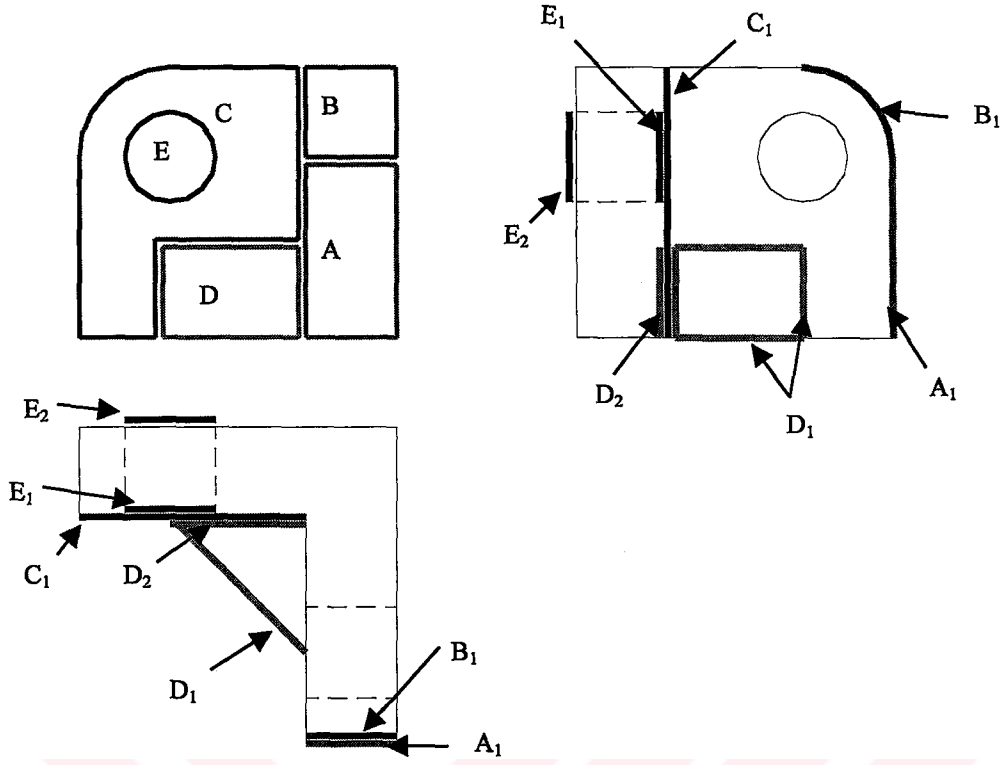


Şekil Ek 1.2. C-yüzeyinin diğer görünümlerdeki izdüşüm karşılığının bulunması.

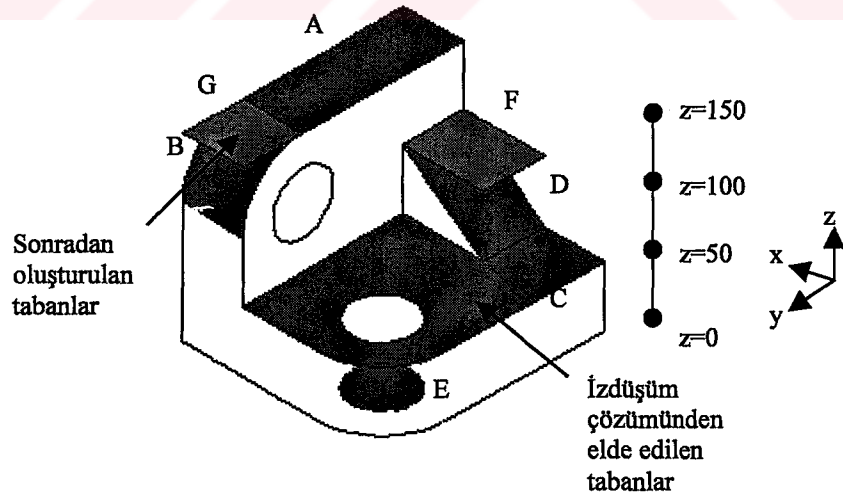
5. Temel görünüşteki her sürekli çizgili alanın izdüşümlerinden hareketle diğer görünüşlerdeki karşılığı olan çizgiler bulunduktan sonra bu çizgilerin ortak çözümü bulunur. Bazen birden fazla ortak çözüm ortaya çıkabilir (D alanının iki tane ortak çözümü vardır). Bu gibi durumlarda Z koordinatı olarak en üstteki ortak çözüm alınmaya çalışılır. Eğer ortak çözümü aranan alan iç alan ise bu ortak çözüm dış alanınkiyle aynı olamaz. Onun dışında en yüksek Z koordinatına sahip olan alınır. E iç alanın birinci çözümü, C dış alanınkiyle aynıdır. Bu nedenle diğer çizgiler ortak çözüm olarak kullanılır.

Ortak çözüm olarak her alan için şu çizgiler çözüm kabul edilir. A, B ve C alanları için tek çözüm vardır (A_1 , B_1 , C_1 çözümleri). D alanı için ise iki çözüm vardır. Bu çözümlerden en üstteki alan olan D_1 çözümü çözüm olarak kabul edilir. E iç alanın ise iki çözümü vardır, fakat bu çözümlerden E_1 çözümü C dış alanın C_1 çözümüyle aynı çizgi üzerindedir. Bu nedenle E alanı için E_1 çözüm olarak alınmaz ve E_2 çözüm olarak kabul edilir (Şekil Ek 1.3).

6. Temel görünüşteki her sürekli çizgili alan için bulunan çözüm çizgilerinin tipine ve koordinatlarına bakılarak, alanın 3-boyutlu uzaydaki şekli ortaya çıkarılır. Bunlar taban alanlarının ilk şekilleridir. Daha sonra bu taban alanlarından eğik ve kavisli olanların üst kısımlarında sonradan o alanın üst izdüşüm alanı olacak şekilde yeni taban alanları oluşturulur. Böylece bunların katılımıyla bütün taban alanları oluşturulmuş olur. Taban alanların Z koordinatına göre daha önceden açıklandığı şekilde yükseklik sırası belirlenir. İleride bu sıralamaya göre unsurlar oluşturulacaktır. Tespit edilen bütün tabanların yükseklik sırası A, G, F, B, C, D, ve E şeklindedir.



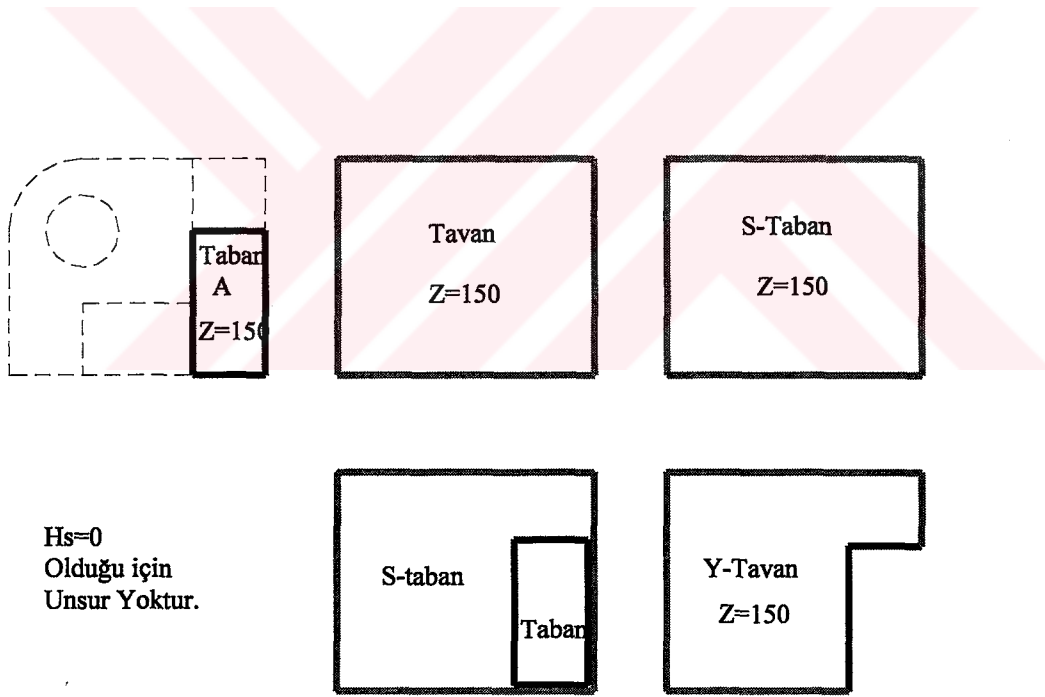
Şekil Ek 1.3. Temel görünüşteki alanların, diğer görünüşlerdeki ortak çözümleri.



Şekil Ek 1.4. Tabanların gösterimi.

7. Bütün tabanlar tespit edildikten sonra unsurların oluşturulmasına geçilir. Bunun için öncelikle “ilk tavan” tespit edilir. İlk tavan temel görünüşün sınır çerçevesinin şeklindedir ve Z yüksekliği en yüksek seviyededir. Her taban kullanımında tavan tabanın seviyesine indirilerek “süpürme taban” (S-Taban) olarak kullanılır. S-Tabanla tavan arasındaki mesafeye ise “süpürme yüksekliği” (H_s) denir ve bu yükseklik “0” dan farklı ise unsur oluşturulur ($H_s=H_{\text{tavan}}-H_{\text{S-taban}}$). Her tabanın oluşturduğu unsur sırasıyla aşağıda açıklanmıştır:

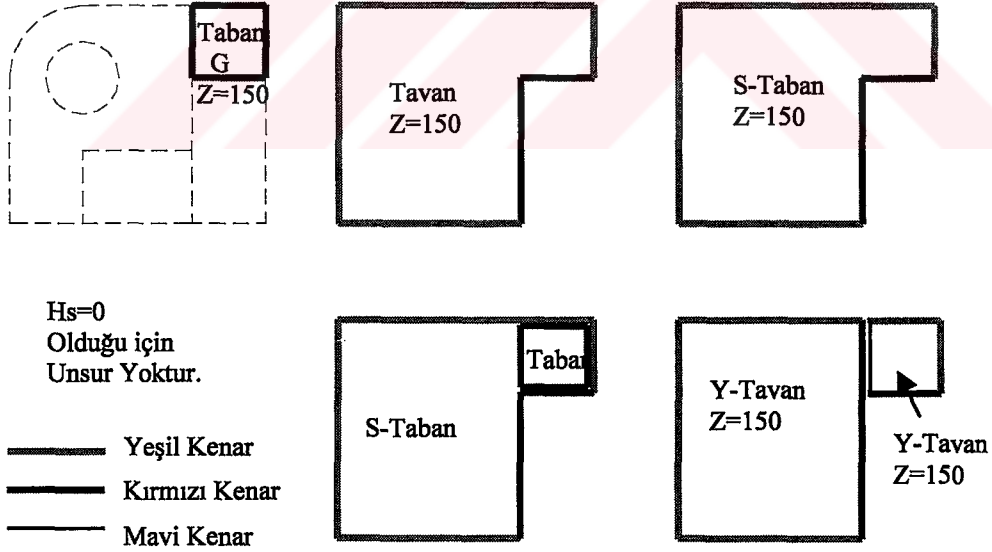
1) **Taban-A:** Bu taban düzdür. İlk kullanılan taban olduğu için ona ait olan tavan ise ilk tavadır. S-Taban ile Tavan arasındaki yükseklik ($H_s=0$) olduğu için unsur oluşmamıştır (Şekil Ek 1.5).



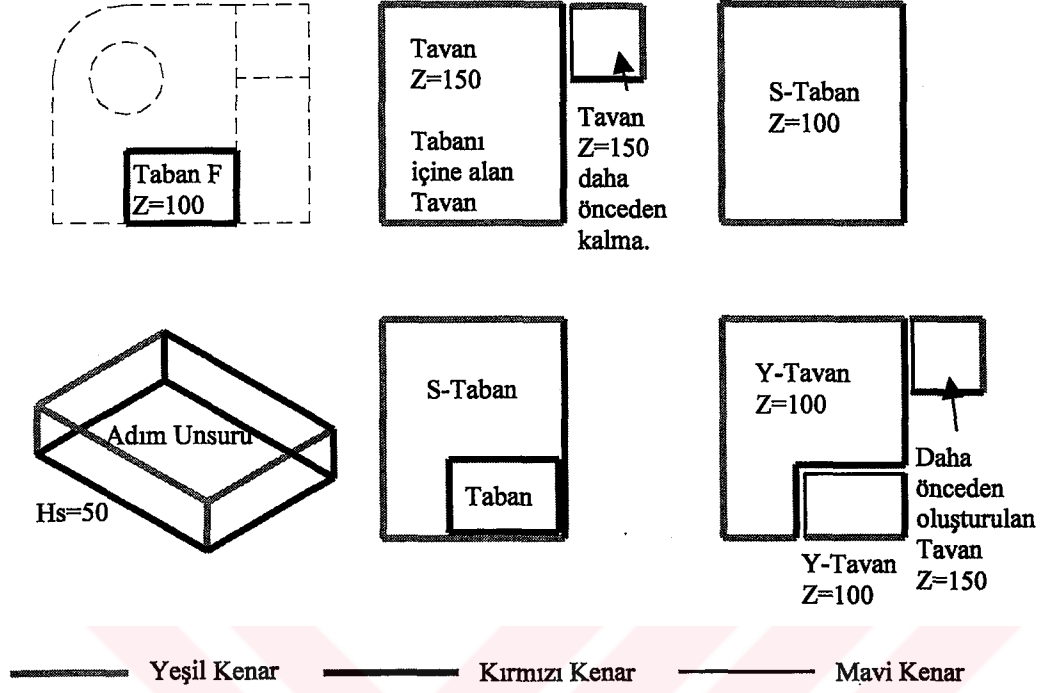
Şekil Ek 1.5. A-tabanına ait tavan, S-taban ve oluşturulan Y-tavanın gösterimi.

2) **Taban G:** Bu taban “düz altı boş” tabandır. Bu nedenle taban, yeni taban oluşturma amacıyla S-tabandan çıkarılırken tabanın kendisiyle aynı fakat S-tabanın renklerini taşıyan yeni bir ikinci taban daha oluşturur. Bu tavan ile komşu yeni tavan arasındaki ortak kenar ise mavi olur. Süpürme yüksekliği ($H_s=0$) olduğu için unsur oluşmamıştır (Şekil Ek 1.6).

3) **Taban F:** Bu taban da “düz altı boş” tabandır. Bu nedenle yeni tavan (Y-Tavan) oluştururken iki tane yeni tavan oluşturulur. Daha önceden de kullanılmayan bir tavan olduğu için üç tane tavan olmuş olur. S-tabanın tipinden oluşturulan unsurun tipi adım (step) olarak belirlenmiştir. Unsurun süpürme yüksekliği ($H_s=50$) dir (Şekil Ek 1.7).

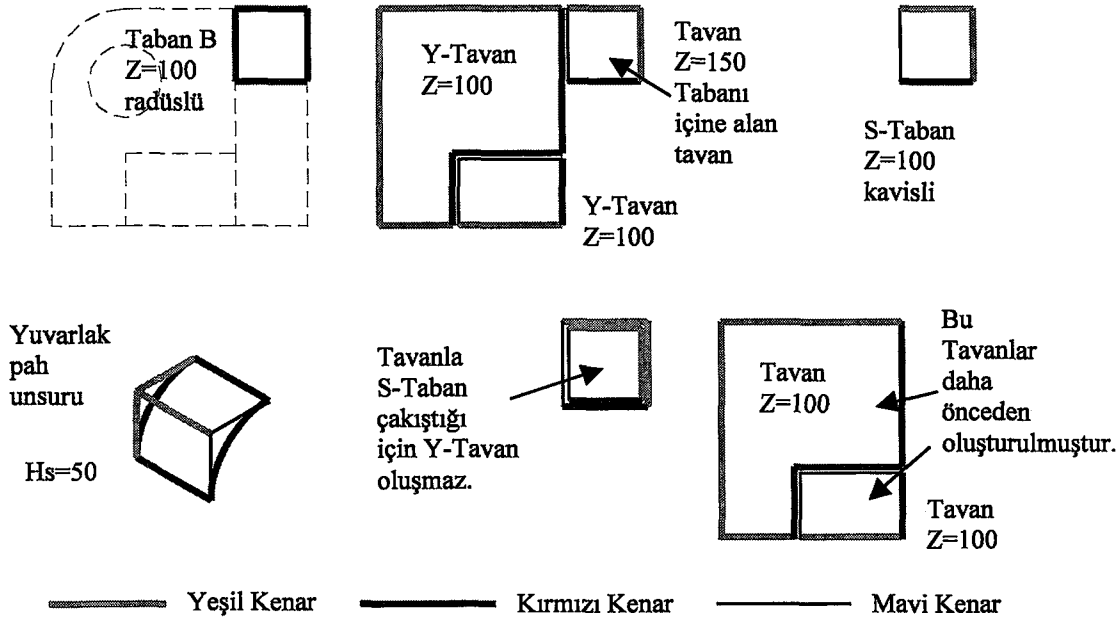


Şekil Ek 1.6. G-tabana ait tavan, S-taban ve oluşturulan Y-tavanların gösterimi.



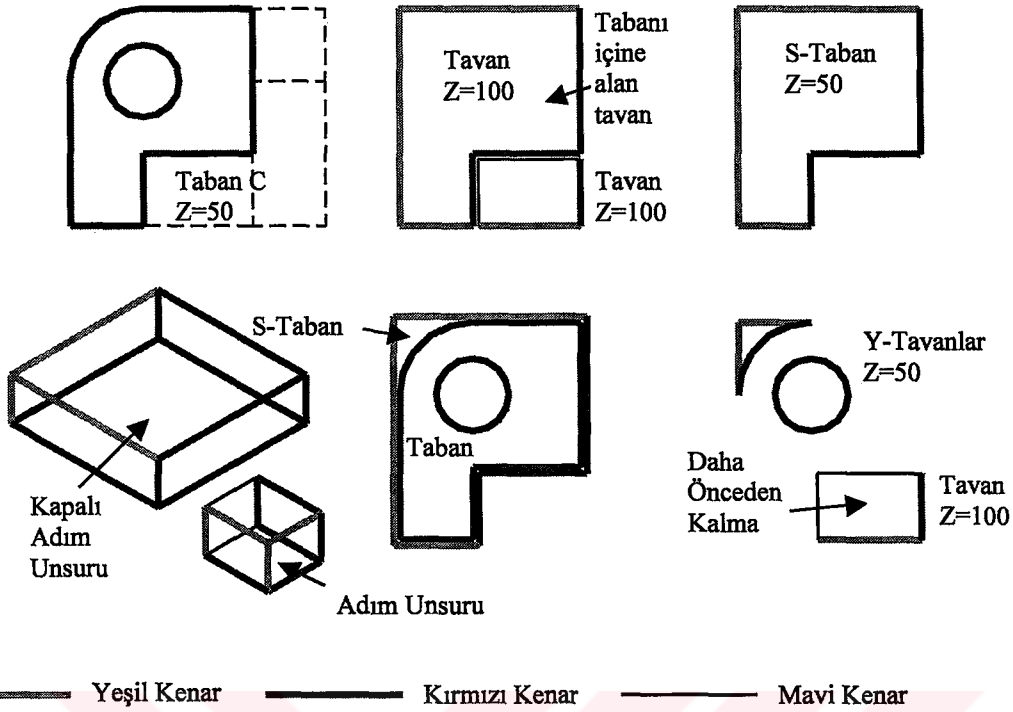
Şekil Ek 1.7. F-tabana ait tavan, S-taban, unsur ve oluşturulan yeni tavanların gösterimi.

4) **Taban B:** Bu taban 3-boyutlu olarak kavisli tabandır, fakat Z doğrultusunda bakıldığında dikdörtgen şeklinde görülmektedir. Daha önceden 3 adet tavan oluşturulduğundan bunlardan içerisine girdiği tavan olarak kullanılır. S-Tabanın şekli tabanla aynı olacağı için kavislidir, fakat kenarlarının rengi tavanınkiyle aynıdır. Oluşan unsur yuvarlak pah unsuru olmuştur. S-Tabanla tavan çakışık olduğundan yeni tavan oluşmamıştır. Daha sonraki aşamaları daha önceden oluşturulan mevcut iki tavan kalmıştır (Şekil Ek 1.8).



Şekil Ek 1.8. B-tabanına ait tavan, S-taban ve unsurun gösterimi.

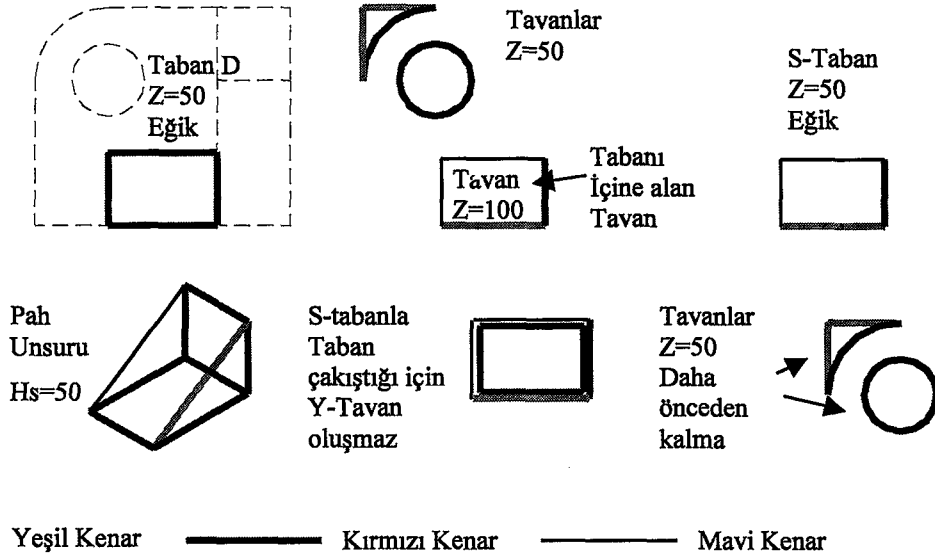
4) **Taban C:** Tabanın şekli düz olup içerisinde de ada şeklinde ikinci bir alan daha vardır. Daha önceden kalma 2 tane tavan olduğu için bunlardan içine girdiği S-taban olarak kullanılır. S-tabanla, tavan arasındaki fark olan süpürme yüksekliği (Hs=50) dir. Oluşan unsurun şekli unsur kütüphanesinde olmadığı için S-taban daha önceden bahsedilen “Karmaşık unsurları parçalama” metoduna göre parçalara ayrılır. Bunun sonucunda 1 tane “Kapalı Adım Unsuru”, 1 tane de “Adım Unsuru” ortaya çıkar. Adım unsurunda mavi yüzey olduğu için işleme sırası “kapalı adım unsurundan” sonra gelir. S-tabandan taban çıkarıldığında iki tane Y-tavan oluşur. Bunlardan 1 tanesi içteki adanın çıkarılmasıyla oluşmuştur. Daha önceden de 1 tane tavan olduğundan, toplamda 3 tane tavan bir sonraki aşamaya bırakılmıştır (Şekil Ek 1.9).



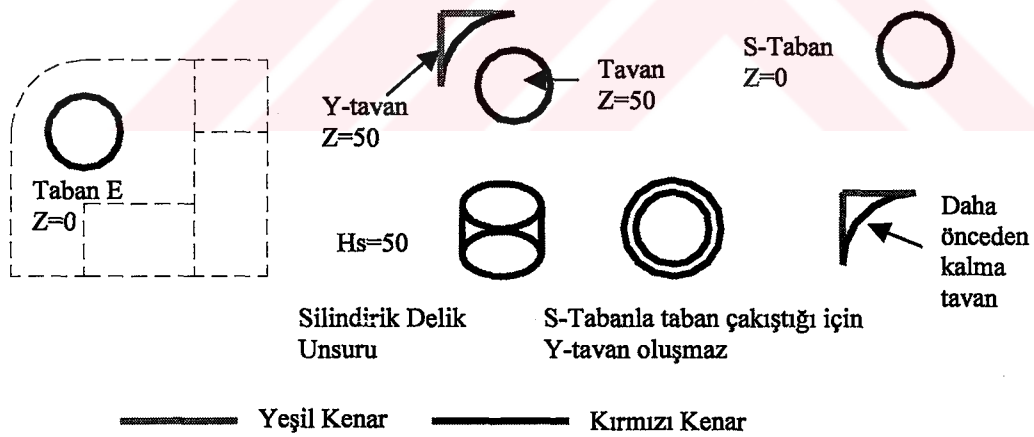
Şekil Ek 1.9. C-tabanına ait tavan, S-taban, unsur ve oluşturulan tavanların gösterimi.

5. Taban D: Bu taban uzayda eğik durmaktadır. Daha önceki aşamadan 3 tane tavan gelmiştir. S-taban, tabanla aynı olacağından eğik durmaktadır, fakat kenarlarının rengi tavanla aynıdır. Oluşan unsur “Düz Pah”dır. S-tabanla tavan birebir çakıştığı için Y-tavan bu aşamada oluşmamıştır. Daha önceden kalma 2 tane tavan bir sonraki aşamaya kalmıştır (Şekil Ek 1.10).

6. Taban E: Bu taban düz ve daire şeklindedir. Daire şeklindeki Y-tavanla çakıştığı için S-taban da daire şeklinde olur ve oluşan hacim “silindirik delik” unsurudur. S-tabanla taban çakıştığı için Y-tavan oluşmamıştır (Şekil Ek 1.11).

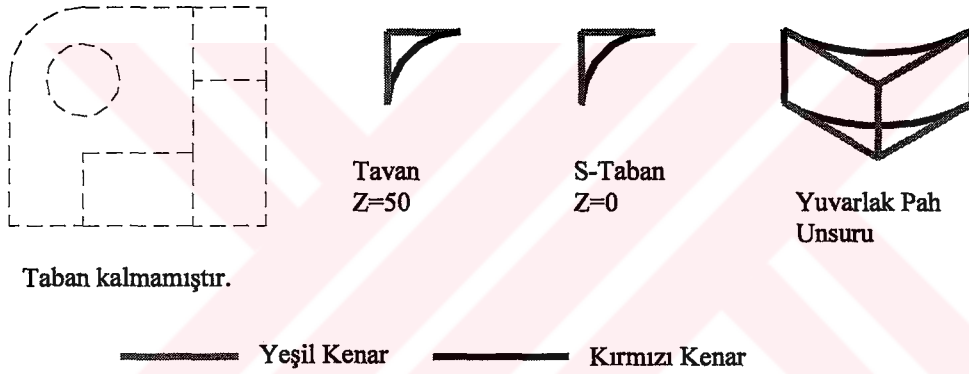


Şekil Ek 1.10. D-tabanına ait tavan, S-taban ve unsurun gösterimi.



Şekil Ek 1.11. E-tabanına ait tavan, S-taban ve unsurun gösterimi.

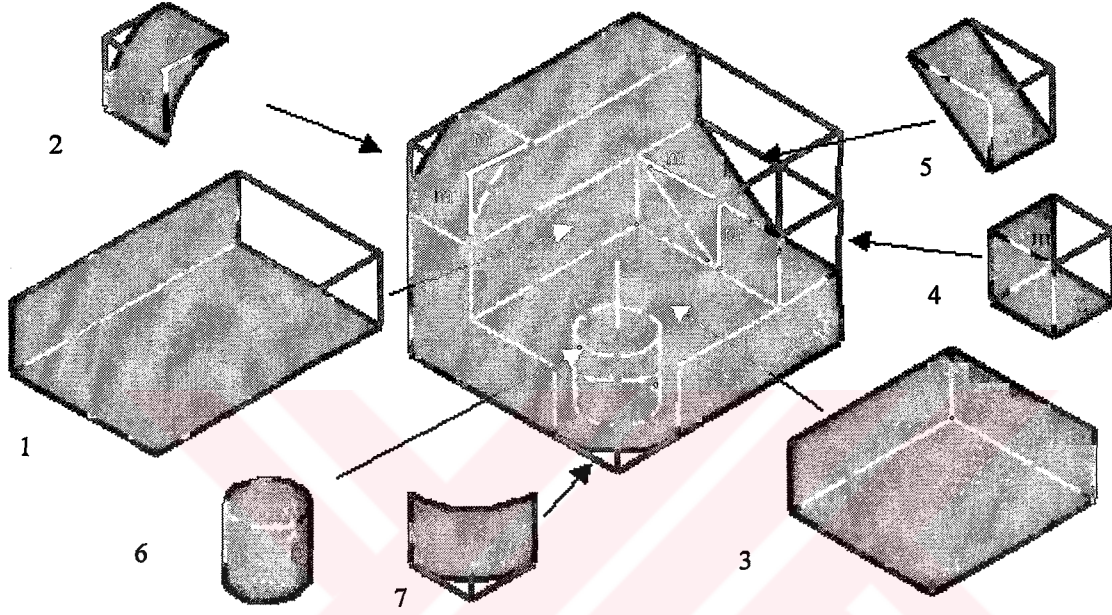
7. Taban Kalmadığında Y-Tavan Bulunma Durumu: Eğer hiç bir “tavan” kalmadıysa ve hala dosyada “tavan” bulunuyorsa oluşturulacak unsur var demektir. Bu durumda tavanlar “S-taban” olarak kullanılır ve S-tabanın yüksekliği ($Z=0$) alınır. Z değeri daha önceki işlemlerde tabandan alındığından burada ise doğrudan “0” kabul edilir. Tavanla, S-taban arasındaki “Hs” mesafesince unsur oluşturulur ve S-tabana bakarak tipi tespit edilir. S-tabandan çıkarılacak taban kalmadığı için artık yeni tavan oluşturulamaz (Şekil Ek 1.12).



Şekil Ek 1.12. Taban kalmadığında, mevcut tavandan oluşturulan unsurlar.

Bütün bu negatif unsurları ham malzemeden (sınır kutudan) çıkarılırsa geriye cismin 3B’lu hali kalır (Şekil Ek 1.13). Dikkat edilirse burada yatay duran delik algoritma tarafından bulunamamıştır. Bunun sebebi bu deliği tanımlayacak bir sürekli çizgili alanın temel görünüşte (ön görünüşte) olmamasıdır. Bu deliği bulabilmek için en son oluşturulan 3B’lu cismin tekrar geriye dönük olarak 2B’lu çizimi oluşturulur. Orijinal çizimle karşılaştırıldığında daha önceden kullanılmamış

çizgiler ortaya çıkar. Hangi görünüşte sürekli çizgili alan var ise, tekrar o görünüş temel görünüş kabul edilerek, o alana ait olan unsur oluşturulana kadar algoritma tamamen bir daha tekrar edilir ve alana ait unsur bulunduğu anda önceki 3-boyutlu hacimden, o unsur da çıkarılır. Bu şekilde geri kontrole devam edilir.

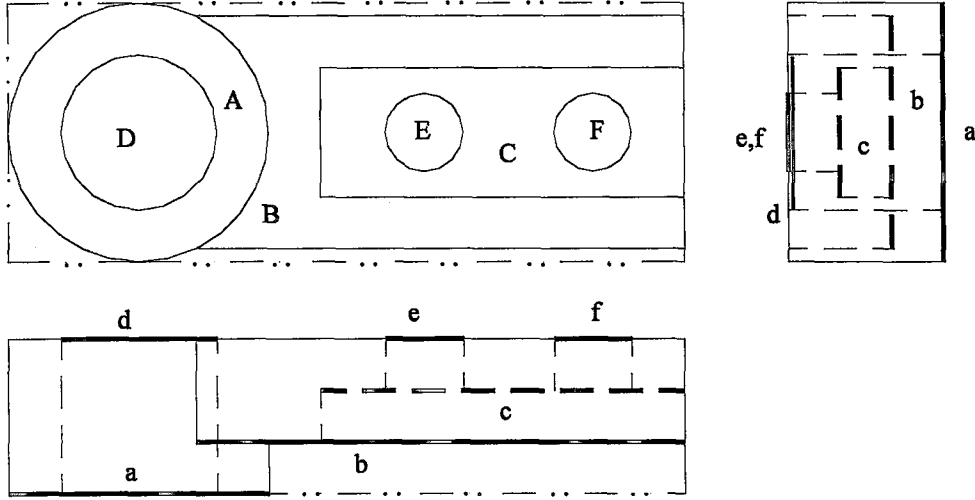


Şekil Ek 1.13. Sınır kutudan unsurların çıkarılması ve 3B'lu cismin oluşturulması (m, unsurlar üzerindeki mavi kenarları gösteriyor. Diğer çizgilerden açık olanları yeşil, koyu olanları kırmızı kenarlardır).

EK 2. YATAKLAMA PARÇASI

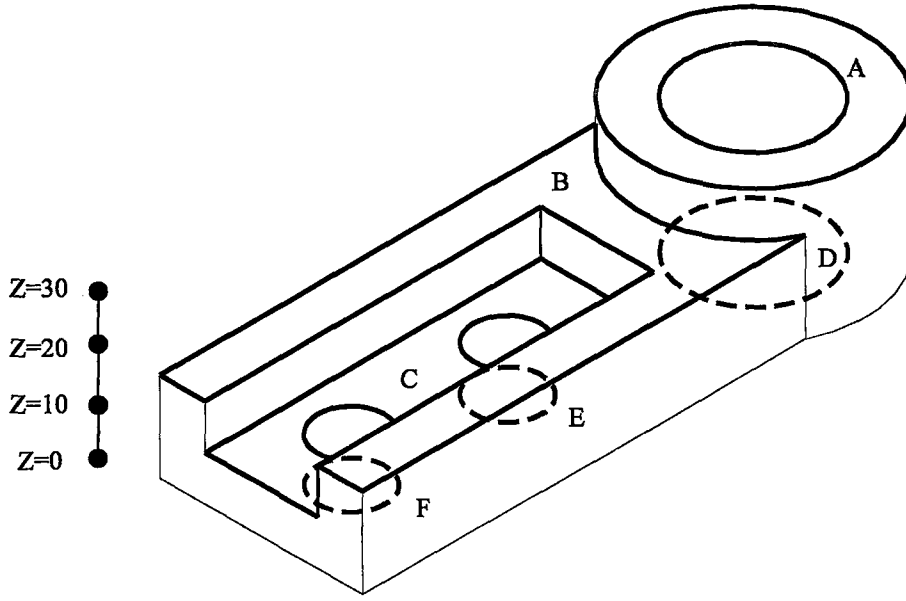
Ele alınan bu parça (Şekil Ek 2.1); üzerindeki delik, kapalı kanal, prizma ve silindir şekillerinin birleşimiyle, makine sektöründe karakteristik olarak kullanılan parçalara bir örnek teşkil etmektedir. Parçanın tablaya bağlandığı alt yüzeyinde herhangi bir işlenecek unsur bulunmamaktadır. Tamamı BSD dik işleme merkezinde bir bağlamada çıkarılabilecektir. Bu sebeple algoritmada temel görünüş bir defa kullanılmaktadır. Parça üzerindeki unsurların çıkarımı ve parçanın 3B'lu modelinin oluşturulması maddeler halinde aşağıda açıklanmaktadır.

1. İlk olarak parçanın üç temel görünüşü, program içerisindeki çizim modülünde çizilir. Çizimdeki bütün çizgiler kesişim noktalarından, daire ve yaylar ise çeyrek noktalardan parçalara ayrılır. Daha sonra bütün çizgilerin görünüşlere taksimi yapılır. Taksimden sonra her görünüşün sınır çerçevesi oluşturularak, çerçeveye ait çizgiler kesişim yerlerinden parçalara ayrılarak görünüşlere eklenir (Şekil Ek 2.1).
2. Her görünüş içerisindeki elemanların, düğüm-koordinat, kenar-düğüm, düğüm-kenar, köşe-kenar-koordinat ve alan-kenar dosyaları oluşturulup kaydedilir.
3. Daha sonra bütün görünüşlerdeki çizgi, çizgi grupları ve sürekli çizgili alanların eksenler üzerine düşen projeksiyonları tespit edilir. Ardından temel görünüşteki sürekli çizgili alanların projeksiyonları ile diğer görünüşlerdeki çizgi veya alanların projeksiyonları eşleştirilerek, sürekli çizgili alanların diğer görünüşlerde karşılığı olan çizgiler veya alanlar tespit edilir. Daha sonra bu çizgilerin ortak çözümü bulunur (Şekil Ek 2.1).



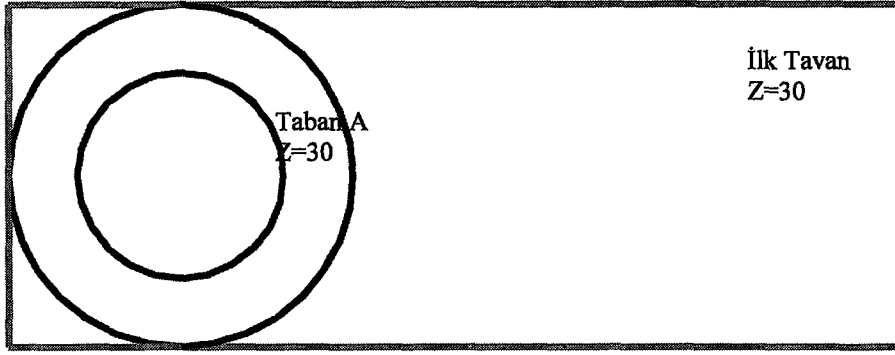
Şekil Ek 2.1. Örnek parçanın üç görünüşü ve temel görünüşteki alanların ortak çözümleri.

4. Alanların ortak çözümünden 3B'lu uzaydaki şekli ortaya çıkarılır. Bu alanların içerisinde eğik ve kavisli yüzeyler olmadığı için doğrudan taban alanı olarak kullanılır. Taban alanların Z koordinatına göre daha önceden açıklandığı şekilde yükseklik sırası belirlenir. İleride bu sıralamaya göre unsurlar oluşturulur. Tespit edilen bütün tabanların yükseklik sırası A, B, C, D, E, F şeklindedir (Şekil Ek 2.2).
5. Bütün tabanlar tespit edildikten sonra unsurların oluşturulmasına geçilir. Bunun için öncelikle “ilk tavan” tespit edilir. Her hacim oluşturulduktan sonra tavan tabanın seviyesine indirilerek “süpürme taban” (S-Taban) olarak kullanılır. S-Tabanla tavan arasındaki mesafeye ise “süpürme yüksekliği” (H_s) denir ve bu yükseklik sıfırdan farklı ise unsur oluşturulur ($H_s = H_{\text{tavan}} - H_{\text{S-taban}}$). Oluşturulan unsurlar sırasıyla aşağıda verilmiştir.

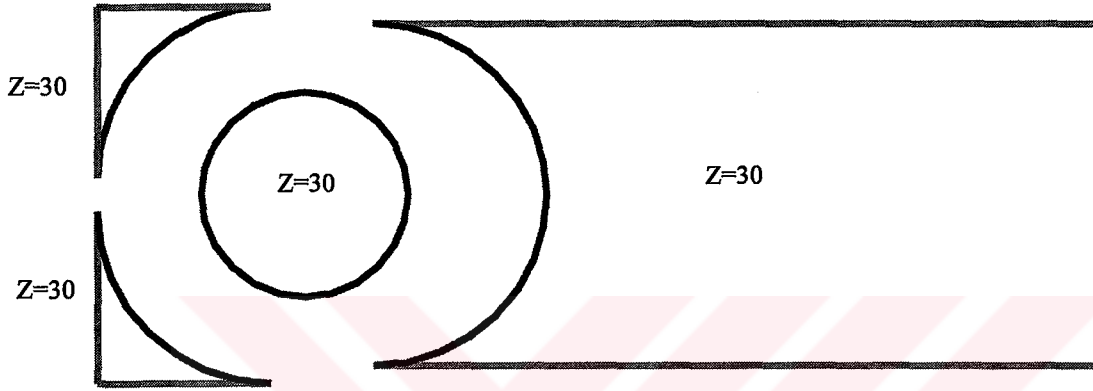


Şekil Ek 2.2. Örnek modeldeki taban yüzeylerin gösterimi.

- 1) **Taban-A:** Bu taban iç içe iki dairenin arasında kalan alandır. İlk kullanılan taban olduğu için ona ait olan tavan ise ilk tavadır ve sınır kutunun en üst yüzeyidir. S-Taban ile Tavan arasındaki yükseklik ($H_s=0$) olduğu için unsur oluşmamıştır (Şekil Ek 2.3).
- 2) **Taban-B:** Düz şekilde duran bu tabanın yüksekliği $Z=20$ 'dir. Şekil Ek 2.4'de görüldüğü gibi $Z=30$ olan bir tavanın içinde bulunmaktadır. Tavanla S-Taban aynı şekle sahiptir, fakat S-tabanın yüksekliği $Z=20$ 'dir. S-tabanın şekli karmaşık olduğundan, unsur tiplerini belirlemek için bu şeklin parçalanması gerekmektedir. Bu amaçla kavisli çizgilerin dik kenarlarından mavi çizgiler oluşturulur. Daha sonra bu mavi çizgilerden alanlar parçalandığında 3 tane unsur çıkar. Bunları işleme sırasına göre dizilirse 1. Unsurun mavi kenarı kırmızı, diğer komşu iki unsurun mavi kenarları yeşile döndürür (Şekil Ek 2.4).



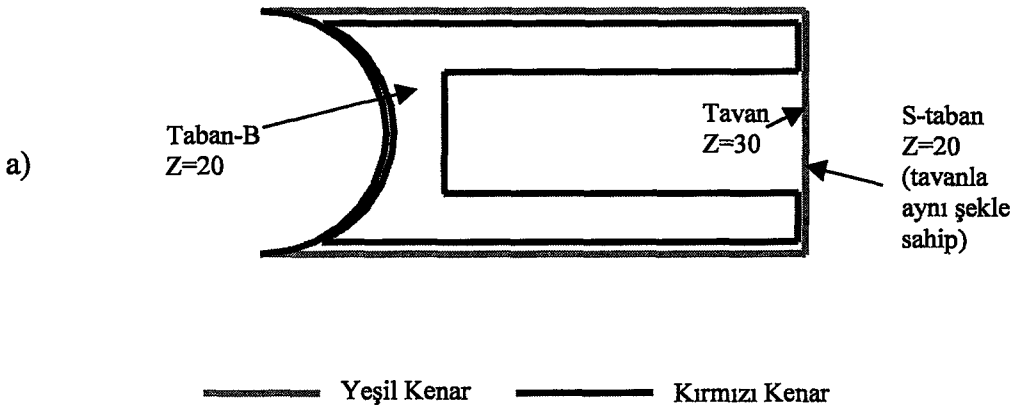
$H_s=0$ olduğu için unsur oluşmaz



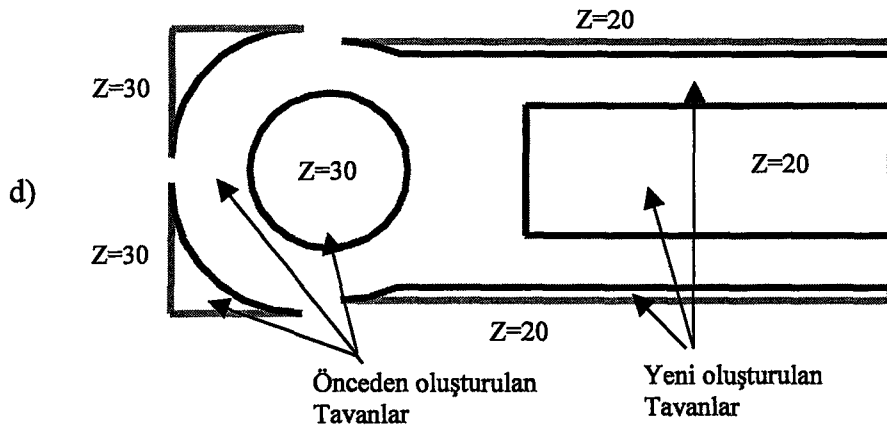
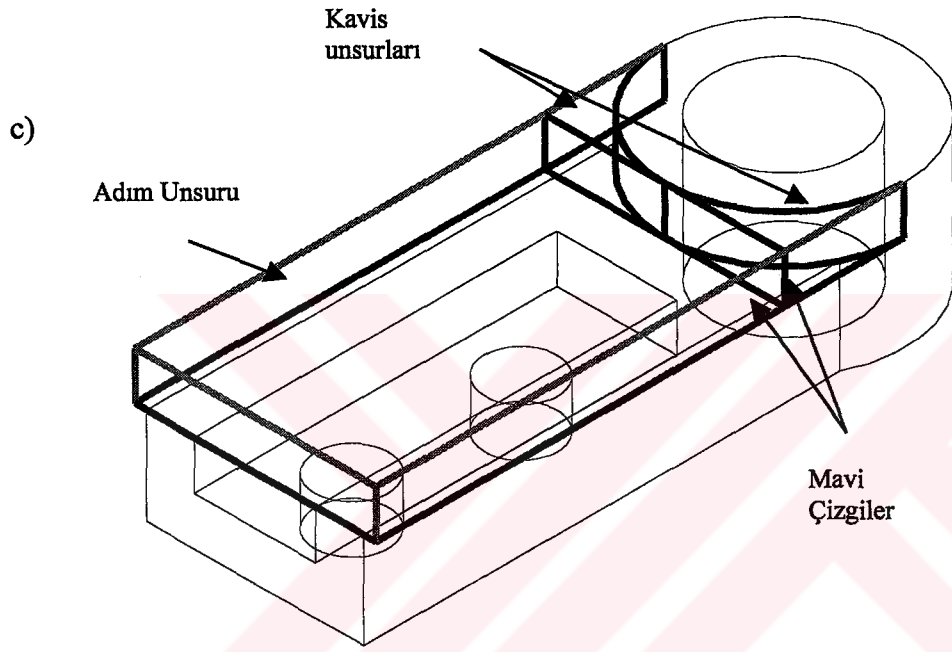
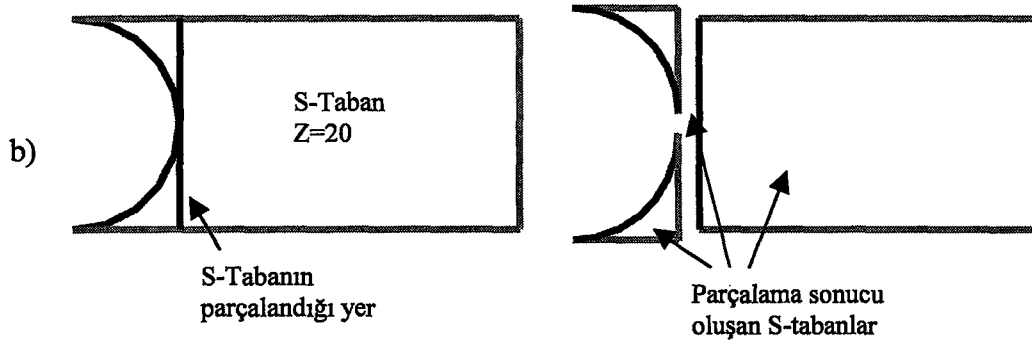
Yeni Tavanlar

———— Yeşil Kenar ——— Kırmızı Kenar

Şekil Ek 2.3. Taban-A, İlk Tavan ve oluşan Yeni Tavanlar.

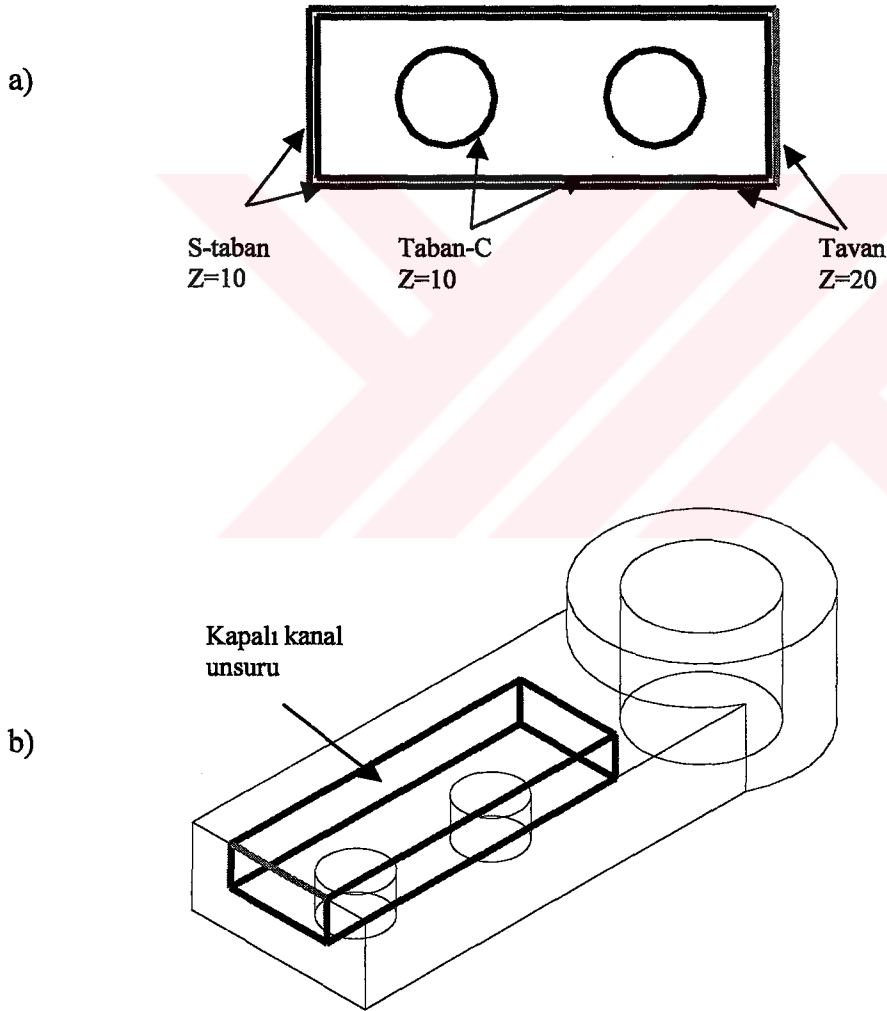


Şekil Ek 2.4. Algoritmanın Taban-B'üzerinde uygulanan aşamaları.

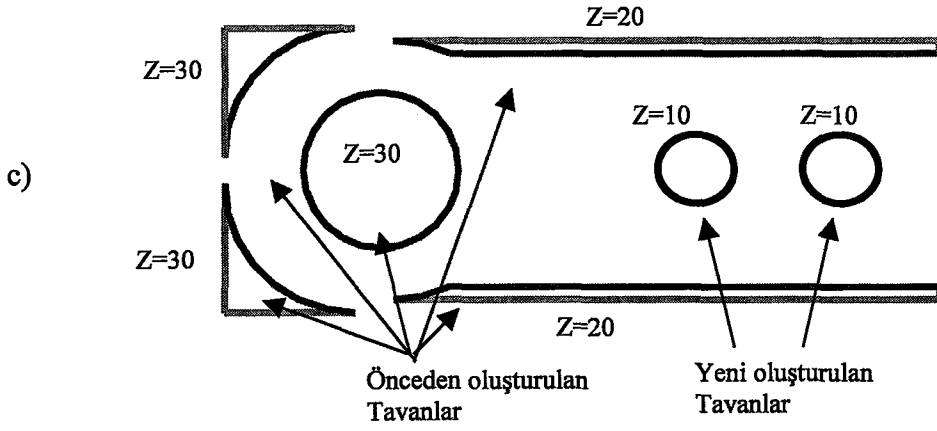


Şekil Ek 2.4 (devam). Algoritmanın Taban-B' üzerinde uygulanan aşamaları.

2) **Taban-C**: İçerisinde 2 tane iç alan bulunan bu taban, sınırları içerisine girmiş olduğu tavanla aynı ölçüdedir. Tabanın yüksekliği $Z_{\text{taban}}=10$ ve tavanın yüksekliği ise $Z_{\text{tavan}}=20$ 'dir. Tavanla aynı kenara, tabanla ise aynı yüksekliğe sahip olan S-tabanın yüksekliği de $Z_{\text{s-taban}}=10$ olur. Süpürme yüksekliği ($H_s=Z_{\text{tavan}}-Z_{\text{s-taban}}=10$) olur ve bu nedenle de unsur oluşur. Unsur tipi ise S-tavanın veya tavanın kenarların renk ve şekli incelenerek kapalı kanal olarak bulunur (Şekil Ek 2.5).

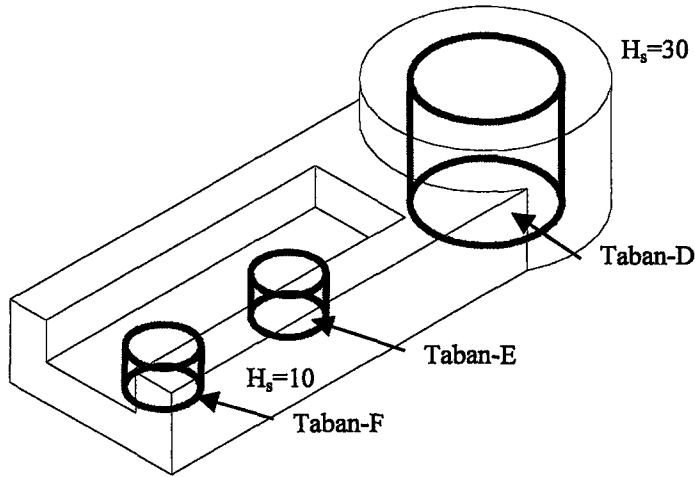


Şekil Ek 2.5. Algoritmanın Taban-C için uygulanan aşamaları.



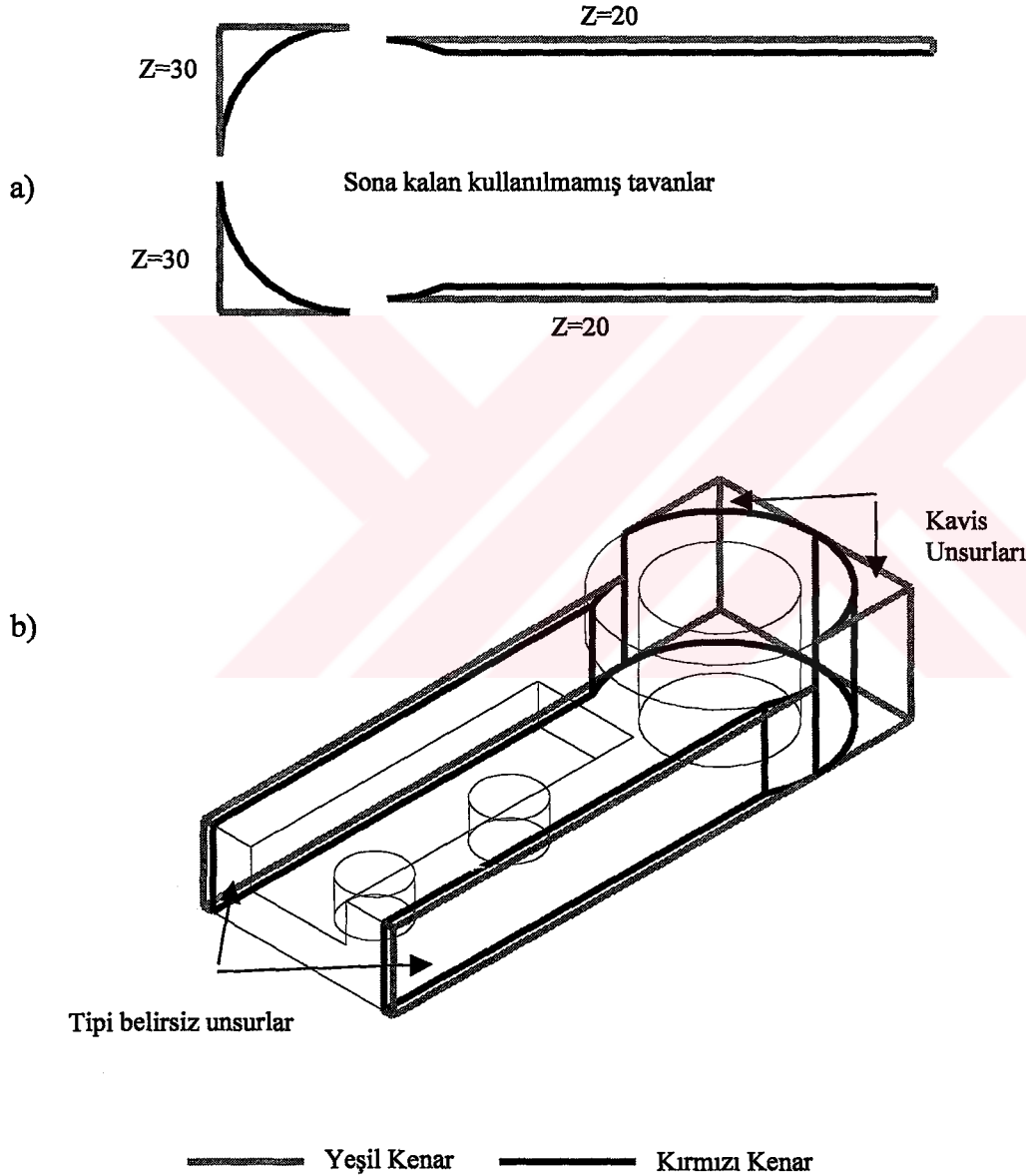
Şekil Ek 2.5 (devam). Algoritmanın Taban-C için uygulanan aşamaları.

4) **Taban-D,E,F:** Bu tabanlar $Z_{\text{taban}}=0$ olan yükseklikte daire şeklinde tabanlardır. Kendilerine ait olan tavanlarda aynı şekle sahiptir. D tabanına ait tavan $Z_{\text{tavan}}=30$ yüksekliğinde, E ve F tabanlarına ait tavanlar ise $Z_{\text{tavan}}=10$ yüksekliğindedir. Bu yükseklikler arasında unsurlar oluşturulursa üç tane silindir şeklinde delik unsuru meydana gelir (Şekil Ek 2.6).



Şekil Ek 2.6. D,E,F tabanlarından oluşturulan unsurların gösterimi (Çizgiler kırmızıdır).

5) **Taban Kalmadığında Oluşan Unsurlar:** Taban kalmadığı zaman hala, kullanılmamış tavan varsa bu tavanlar $Z=0$ yüksekliğinde S-taban olarak alınıp bu yükseklikten kendi yüksekliklerine süpürülerek unsurlar oluşturulur. Burada tabanı olmayan 4 tane tavan kalmıştır. Bunların ikisinden kavis unsuru oluşur. Diğer ikisinden ise belirsiz unsur oluşur (Şekil Ek 2.7). Bu belirsiz unsurlarda parçalanırsa pah ve kapalı adım unsurları oluşturulabilir.



Şekil Ek 2.7. Taban kalmadığında geriye kalan tavanlardan oluşturulan unsurların gösterimi.