

**BMP FORMATLI İKİ BOYUTLU RESİMLERİN DWG FORMATINA
DÖNÜŞTÜRÜLMESİ İÇİN BİLGİSAYAR PROGRAMI TASARIMI**

Ümit AKBAŞ

**YÜKSEK LİSANS TEZİ
MAKİNE EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

EYLÜL 2006

ANKARA

Ümit AKBAŞ tarafından hazırlanan BMP FORMATLI İKİ BOYUTLU RESİMLERİN DWG FORMATINA DÖNÜŞTÜRÜLMESİ İÇİN BİLGİSAYAR PROGRAMI TASARIMI adlı tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Mahmut GÜLESİN

Tez Yöneticisi

Bu çalışma jürimiz tarafından oy birliği ile Makine Eğitimi Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Başkan: Prof. Dr. Faruk MENDİ

Üye : Prof. Dr. Mahmut GÜLESİN

Üye : Doç. Dr. Abdullah ÇAVUŞOĞLU

Üye : Yard. Doç. Dr. Abdulkadir GÜLLÜ

Üye : Yard. Doç. Dr. İhsan KORKUT

Tarih : 21.09.2006

Bu tez Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

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

Ümit AKBAŞ

**BMP FORMATLI İKİ BOYUTLU RESİMLERİN DWG FORMATINA
DÖNÜŞTÜRÜLMESİ İÇİN BİLGİSAYAR PROGRAMI TASARIMI
(Yüksek Lisans Tezi)**

Ümit AKBAŞ

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Eylül 2006

ÖZET

Bu çalışmada, Bitmap (BMP) uzantılı teknik resim çizimleri VB and VBA (Visual Basic Application) programlama yöntemi kullanılarak AutoCAD ortamında DWG vektörel veri formatına dönüştürülmesi için bir program geliştirilmiştir. Bu amaçla bilgisayar ortamında bulunan Bitmap uzantılı resimler yapılan program tarafından taranarak koordinat noktaları tespit edilmekte ve text veri tabanı olarak kayıt edilmektedir. Öncelikle CAD veri formatına dönüştürülecek Bitmap uzantılı resim 24 bit ve 800 x 600 piksel siyah-beyaz olarak kayıt edilmektedir. BMP veri tabanındaki resim hazırlanan programda piksel piksel yatay ve dikey olarak taranarak resim üzerindeki nesnelerin x ve y koordinatları bulunmaktadır. Geliştirilen program AutoCAD programını çalıştırmakta listedeki koordinat değerleri ActiveX komutları kullanılarak DWG tipine dönüştürülmekte kayıt edilmektedir. Bu çalışmada Visual Basic 6.0 ve Visual Basic Application programlama dili kullanılmıştır.

Bilim Kodu	: 708.3.028
Anahtar Kelimeler	: Bitmap, Resim İşleme, DWG, Vektörel veri formatı, Teknik Resim, AutoCAD
Sayfa Adedi	: 118
Tez Yöneticisi	: Prof. Dr. Mahmut GÜLESİN

**DEVELOPING A COMPUTER PROGRAM FOR CONVERTING OF 2D
PICTURES WITH BMP FORMAT TO DWG FORMAT**

(M.Sc. Thesis)

Ümit AKBAŞ

**GAZI UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

September 2006

ABSTRACT

In this study, a computer program has been developed for converting of technical drawing with Bitmap format to DWG vectorial data format in AutoCAD using VB and VBA (Visual Basic Application) programming method. The picture files with Bitmap extension saved in the computer are scanned, coordinate points are determined and saved in text format. First of all, the picture with Bitmap file extension to be converted to CAD data format is saved 24 bit and 800x600 pixels as black and white. The Bitmap picture is scanned pixel by pixel horizontally and vertically and the x, y coordinates of the objects on the picture are determined. AutoCAD is run by the program developed and the coordinate values are converted to DWG format using ActiveX commands and saved in a file. In this work Visual Basic 6.0 and Visual Basic Application programming language were used.

Science Code : 708.3.028
**Key Words : Bitmap, Picture Processing, DWG, Vectorial data format,
Convert of Picture, AutoCAD**
Page Number : 118
Adviser : Prof. Dr. Mahmut GÜLESİN

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkıları ile beni yönlendiren danışmanım Sayın Prof. Dr. Mahmut GÜLESİN' e tezimin oluşturulmasında büyük katkısı geçen değerli kardeşim Sayın Mehmet AKBAŞ'a, tezimin oluşturulmasında yardımlarını esirgemeyen Sayın Yrd. Doç.Dr. Hüdayim BAŞAK'a ve değerli dostum Sayın Sinan İNCE'ye tez çalışmam süresince desteklerini esirgemeyen tüm dostlarıma ve aileme teşekkürlerimi bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ	x
ŞEKİLLERİN LİSTESİ	xi
SİMGELER VE KISALTMALAR	xv
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	3
3. GRAFİK FORMATLARI	8
3.1. Vektör Grafikler	8
3.2. Bitmap (Raster) Tabanlı Grafikler	12
3.3. Başlıca Bitmap Tabanlı Formatlar	15
3.3.1. Bmp veya dib (Bitmap)	15
3.3.2. Cpt (Corel photo paint)	16
3.3.3. Png (Portable network graphic)	16
3.3.4. Psd (Photoshop document)	16
3.3.5. Tga (Truevision targa)	17
3.3.6. Tif (Tagged image format)	17
3.3.7. Jpeg (Joint photographic experts group)	17
3.3.8. Gif (Compuserve graphics interchange format)	18

Sayfa

3.4. Dwg (AutoCAD drawing format) format yapısı.....	19
4. BMP (BITMAP) DOSYA YAPISI.....	20
4.1. Pikselin tanımı	20
4.2. Kırık, Kademeli Görüntü (Antialiasing) Oluşumu	22
4.3. Renk Derinliği (Color depth).....	22
4.4. Çözünürlük (Resolution)	23
5. RESİM İŞLEME.....	28
6. MICROSOFT VISUAL BASIC	30
6.1. Kullanılan Olay ve Nesneler.....	32
6.2. Kullanılan Visual Basic Döngüleri ve Şartları	35
7. VISUAL BASIC APPLICATION (VBA) VE ACTIVEX AUTOMATION	37
7.1. Visual Basic Application (VBA)	37
7.2. ActiveX Automation	38
7.2.1. Otomasyon nesneleri	39
7.2.2. Otomasyon ve AutoCAD	40
7.2.3. AutoCAD’de nesne modeli ve hiyerarşi.....	40
7.2.4. Visual Basic programından AutoCAD’e bağlanma	41
7.2.5. ActiveX Automation AutoCAD komutları	43
8. GELİŞTİRİLEN PROGRAM	44
8.1. Visual Basic ve ActiveX Komutlarını Kullanarak (Bmp) Dosya Okuma.....	44
8.2. Piksel Yorumlamaları.....	48
8.2.1. Noktanın yorumlanması.....	52

Sayfa

8.2.2. Doğru yorumlanması	57
8.2.3. Yay ve çemberlerin yorumlanması.....	70
8.3. Aynalama.....	82
8.4. Karmaşık Parçaların, Ölçülerin ve Yazıların Noktasal Olarak Çizilmesi	84
9. PROGRAMIN KULLANIMI	85
10. SONUÇ VE ÖNERİLER	92
10.1. Sonuçlar	92
10.2. Öneriler.....	93
KAYNAKLAR.....	94
EKLER.....	96
EK-1 AutoCAD çizim sözlüğü	97
EK-2 Dwg formatı kodları örneği	102
EK-3 Örnek Resimler.....	112
EK-4 Akış Diyagramı.....	117
ÖZGEÇMİŞ.....	118

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Çözünürlük eşlemeleri.....	26
Çizelge 7.1. Doğru çeşitleri.....	43
Çizelge 7.2. Eğri çeşitleri.....	43
Çizelge 8.1. Piksel ölçü değerleri	50

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. CAD programında çizilmiş vektör tabanlı resim	8
Şekil 3.2. Vektör, Bitmap resim arasındaki farklar	9
Şekil 3.3. Vektör tabanlı grafiklerin görüntü oranı	10
Şekil 3.4. Resim formatlarının çözünürlükleri	11
Şekil 3.5. Bitmap piksel yapısı	12
Şekil 3.6. Bitmap resmin büyütüldüğündeki görüntüsü.....	13
Şekil 3.7. AutoCAD programında çizilmiş Dwg resmi	19
Şekil 4.1. Piksel tanımlaması	20
Şekil 4.2. Piksel boyutları	21
Şekil 4.3. Piksel renkleri	22
Şekil 4.4. Piksel antialiasing özelliği	22
Şekil 4.5. 24 Bit renk özelliği.....	23
Şekil 4.6. Çözünürlük	24
Şekil 4.7. Ekran çözünürlüğü	26
Şekil 5.1. Raster görüntünün ekranda gösterilmesi	28
Şekil 6.1. Visual Basic ekranı	30
Şekil 6.2. Command butonunun tıklanması	30
Şekil 6.3. Seçilebilecek proje çeşitleri	31
Şekil 6.4. Program editörü	32
Şekil 7.1. AutoCAD programında çalışan (VBA) ekranı	37
Şekil 7.2. Alıcı-sunucu iletişimi	38

Şekil	Sayfa
Şekil 8.1. Paint programı arayüzü	44
Şekil 8.2. AutoCAD programı arayüzü.....	47
Şekil 8.3. Paint programında resim formatlarının kaydedilmesi.....	47
Şekil 8.4. “0 ve 1” pikseller.....	48
Şekil 8.5. Listeleme kutuları ve koordinat değerleri.....	49
Şekil 8.6. Düşey eksende tarama	51
Şekil 8.7. 800x600 ekran çözünürlüğü	51
Şekil 8.8. Noktanın yorumlanması	52
Şekil 8.9. Nokta taranması	53
Şekil 8.10. Nokta koordinatları	54
Şekil 8.11. Noktanın paint programında görünümü	55
Şekil 8.12. AutoCAD nokta sitil diyalog kutusu.....	55
Şekil 8.13. AutoCAD nokta uygulaması.....	56
Şekil 8.14. Doğru parçaları	57
Şekil 8.15. Yatay doğru taraması.....	58
Şekil 8.16. Yatay doğru için başlangıç ve bitiş doğruları koordinatları	58
Şekil 8.17. Yatay doğru uygulaması.....	60
Şekil 8.18. Yatay doğrunun paint programında gösterilmesi.....	61
Şekil 8.19. AutoCAD programında yatay doğru uygulaması	61
Şekil 8.20. Dikey doğru taraması	62
Şekil 8.21. Dikey doğru için başlangıç ve bitiş doğruları koordinatları	62
Şekil 8.22. Dikey doğru uygulaması.....	64

Şekil	Sayfa
Şekil 8.23. Dikey doğrunun paint programında gösterilmesi.....	65
Şekil 8.24. AutoCAD programında dikey doğru uygulması.....	65
Şekil 8.25. Açılı doğru taraması	66
Şekil 8.26. Açılı doğru için başlangıç ve bitiş koordinatları.....	66
Şekil 8.27. Açılı doğru uygulaması	68
Şekil 8.28. Açılı doğrunun paint programında gösterilmesi	69
Şekil 8.29. AutoCAD programında açılı doğru uygulaması	69
Şekil 8.30. Yay parçası	70
Şekil 8.31. Yay parçasında piksellerin durumu.....	70
Şekil 8.32. Yayın X ve Y koordinatları	71
Şekil 8.33. Yay parçası uygulaması.....	73
Şekil 8.34. Yay parçasının paint programında görüntülenmesi	75
Şekil 8.35. AutoCAD ortamında yayın çizilmesi.....	75
Şekil 8.36. Çember ve elemanları.....	76
Şekil 8.37. Piksellerle oluşturulan çember.....	76
Şeki 8.38. Çemberin X ve Y değerleri.....	77
Şekil 8.39. Çember uygulaması.....	78
Şekil 8.40. Çemberin paint programında görüntülenmesi	79
Şekil 8.41. AutoCAD ortamında çemberin çizilmesi	79
Şekil 8.42. Eksen ve kesik çizgi	80
Şekil 8.43. Eksen ve kesik çizgi uygulaması	80
Şekil 8.44. Parça resmi.....	81

Şekil	Sayfa
Şekil 8.45. Paint ve AutoCAD programında doğrulardan oluşmuş parça	81
Şekil 8.46. Paint ve AutoCAD programında prizmatik parça gösterilmesi	82
Şekil 8.47. Aynalama uygulaması	82
Şekil 8.48. Aynalama sonucu çizimin gösterilmesi.....	83
Şekil 8.49. Ölçülendirilmiş parça çizimi.....	84
Şekil 9.1. Program ana penceresi.....	85
Şekil 9.2. Dosya açma penceresi	86
Şekil 9.3. Dosya isminin görüntülenmesi	86
Şekil 9.4. Resim işleme menüsü.....	87
Şekil 9.5. Txt veri oluşturulması	87
Şekil 9.6. İşlem menüsü	88
Şekil 9.7. Başlangıç ve bitiş noktalarının listelenmesi	88
Şekil 9.8. AutoCAD programına resmin gönderilmesi.....	89
Şekil 9.9. Nokta koordinatı uygulaması.....	89
Şekil 9.10. Nokta koordinatları	90
Şekil 9.11. AutoCAD ortamına nokta koordinatların gönderilmesi.....	90
Şekil 9.12. Çıkış menüsü.....	91

SİMGELER VE KISALTMALAR

Bu tez çalışmasında kullanılan bazı simge ve kısaltmaların açıklamaları aşağıda verilmiştir.

Simgeler

Açıklama

PT1

Başlangıç koordinat Noktası

PT2

Bitiş Koordinat Noktası

Kısaltmalar

Açıklama

DWG

Drawing Format

DXF

Data Exchange Format

BMP

Bitmap

GIF

Graphic Interchange Format

JPEG veya JPG

Joint Photographic Experts Group

TIF veya TIFF

Tagged Image File Format

PSD

Adobe PhotoShop File Format

PNG

Portable (Public) Network Graphic

WMF

Windows Meta File

VB

Visual Basic

VBA

Visual Basic Application

CAD

Computer Aided Design

RGB

Red Green Blue (Kırmızı-Yeşil-Mavi) Renk Grubu

1. GİRİŞ

Günümüzde, artan sayıda kullanılan bilgisayar destekli çizim ve üretim programları üretime esneklik, düşük maliyet, istenilen tolerans ve kaliteli parça üretimi gibi konularda büyük kolaylıklar sağlamaktadır. Bilgisayar destekli çizim ve üretim programlarının asıl amacı daha kısa sürede tüketicinin beklediği kalitede üretim yapılmasını sağlamaktır. Bu amaç doğrultusunda birçok yazılım firması sektöre girmiştir. Bunlardan AutoCAD programı Autodesk firmasının sektöre kazandırdığı bir yazılımdır. Gelişen imalat yöntemleri doğrultusunda, hassas işlemlerin yapılabilmesi için ülkemiz orta ve büyük ölçekli üretim firmalarında AutoCAD ve benzeri yazılımlar yaygın olarak kullanılmaktadır. CAD/CAM sistemlerinde kullanılmakta olan veri yapıları DXF, DWG, IGES, STEP ve SET' dir.

Bilindiği gibi AutoCAD yazılımı DXF ve DWG (Drawing) formatı kullanarak yapılan çizimler kaydedilmekte ve çıktı alınmaktadır. DWG formatı AutoCAD programına ait bir kayıt formatıdır ve şu anda bu formatta yapılan çizimler bir kısım yazılımlar tarafından desteklenmektedir.

Visual Basic ayrı çalışan Visual Basic Application programı ise AutoCAD altında çalışan bir programdır. VBA' da yapılan bir yazılım sadece AutoCAD ortamına yüklenerek çalıştırılabilir. VBA kodları VB kodları ile aynıdır farklı olarak AutoCAD programına ait komutlarda içermektedir. ActiveX automation kullanımında bir başka programdan koordinat değerleri girilerek AutoCAD programı çalıştırılarak çizim yaptırılabilir.

BMP resim formatı Bitmap tabanlı bir resim formatıdır ve piksel (noktasal) özellik gösterir. Resim piksel piksel kaydedilir ve ekranda gösterilir.

DWG vektörel bir formattır. Bu format üzerinde istenilen değişiklik özellik kaybı olmadan yapılabilir. Bu çalışmada, BMP resimleri DWG formatına çeviren bir program tasarlanmıştır. Elde edilen resim DWG vektörel formatında AutoCAD programı ortamında açılabilir.

Önceden elle çizilmiş veya çıktısı elde olan ancak CAD ortamında kayıtlı dosyası olmayan teknik resimler tekrar çizilmek yerine tarayıcıda taranarak BMP (Bitmap) resim formatında kaydedilmektedir. Visual Basic (VB) ve Visual Basic Application (VBA) programlama dili kullanılarak BMP resim formatı DWG formatına dönüştürülmekte AutoCAD ortamında açılması ve üzerinde değişiklik yapılması sağlanmaktadır.

2. LİTERATÜR ARAŞTIRMASI

Bilgisayar ve bilgisayar teknolojisine sahip makinelerin teknik çizim ve modelleme alanında kullanılması gün geçtikçe daha da yaygınlaşmaktadır. Özellikle çizim ve modelleme programlarının sayısı artmakta bununla birlikte dosya veri çeşitliliğini de beraberinde getirmektedir.

Her kullanılan programın görsel modelleme, dosya veri boyutu, dosya destekleme çeşitliliği, çizim ve modelleme kabiliyeti, kullanıcı ara yüzü gibi birbirine göre avantajları ve dezavantajları vardır. Bu nedenle piyasada kullanılan programların sayısında çok farklılık olabilmektedir. Kullanılan programın çeşitliliği dosya veri çeşitliliğini ve veri dönüştürme zorluğunu beraberinde getirmektedir.

Çizim ve modelleme programlarının birçok veri dosyasını desteklemesi ve dosya veri dönüşümünü yapması kullanıcı sayısını artırmaktadır. Sadece, dosya verilerini birbirine dönüştüren programlar da bulunmaktadır. Böylece herhangi bir veri diğer verilere dönüştürülerek kullanılabilirliği artırılmaktadır. Veri dönüştürme işlemi ile zaman, para ve iş gücü kazancı yapılmaktadır.

Dosya veri dönüştürme programları; bilgisayar ortamında yapılmayan veya bilgisayar çizim ve modelleme programlarından önce yapılan çizim ve modellerin de sayısal verilere dönüştürülerek tekrar kullanmasını sağlamaktadır. Ayrıca teknik çizim ve modellemeler arşivleme amaçlı olarak da yaygın şekilde kullanılmaktadır.

Özellikle ofis işlemlerinde, çok kullanılan resim formatlarının birbirine yazı karakterlerinin sayısal verilere dönüştürülmesinde ve veri tabanının dönüştürülmesinde önemli başarılar sağlanmıştır. Son zamanlarda CAD çizimlerinin ve teknik resim çıktılarının CAD programlarına aktarılması, yeniden düzenlenmesi veya arşiv olarak saklanması çalışmaları da yapılmaktadır.

Bu işlemlerde BMP, JPG, GIF, TIFF vb. resim veri yapısına sahip çizimler CAD veri veri yapısına dönüştürülmektedir. Bu işlemde teknik resim çıktıları tarayıcı ile

taranarak normal resim formatı olarak kayıt edilebilmektedir. Normal resim formatına sahip bu çizimler veri formatı dönüştürücü (converter) programları ile CAD formatına dönüştürülmektedir.

Resim formatlarının CAD veri formatına dönüştürülmesi için araştırma ve eğitim amaçlı çalışmalar çok az sayıda bulunmakta olup daha çok profesyonel ticari amaçlı yazılım ve programlar bulunmaktadır. Bu çalışmalarda doğru ve eğri çizgilerin tanımlanması ve yorumlanmasında önemli başarılar sağlanmış ise de yay, daire, spiral çizgiler, ölçülendirmeler, toleranslar, yazı ve harf karakterlerinin tanıtılması ve yorumlanmasında önemli bir başarı sağlanamamıştır.

Çalıkoğlu ve Ceylan yaptıkları çalışmada Paint Brush programında çizilmiş veya sayfa tarayıcılarla okutularak kaydedilen eğrilerin koordinatlarını belirleyen bir program gerçekleştirmişlerdir. Yapılan program ile eğri bilgileri sayısallaştırılmıştır. Eğri bilgilerinin sayısallaştırılması bilgisayarlarda işlenmesini mümkün kılmaktadır. Gerçekleştirilen bu program bir alt program olarak çalışabilecek şekilde hazırlanmıştır. Programlama dili olarak Turbo Pascal programlama dili kullanılmıştır.

Program, Paint Brush programı ile "Sıkıştırılmamış Format" seçeneği ile saklanmış, Tif uzantılı dosyaları okuyabilmektedir. Program üç bölümden oluşmaktadır. Birinci bölümde program tarafından sıkıştırılmamış formatla saklanmış dosyalar okutularak bir diziye kaydedilir. İkinci bölümde dizideki veriler işlenerek noktaların koordinat bilgileri belirlenmektedir. Üçüncü bölümde ise, grafiklerin ekrana belirlenen koordinatlarda basılması veya noktaların koordinat bilgilerinin ekranda belirlenmesi sağlanır. Bu çalışma arazi topografik yapısını belirten haritaların koordinat bilgilerinin işlenmesinde kullanılabileceği düşünülmüş ve hazırlanmıştır [1].

Idesawa yaptığı çalışmada 2B'lu çizimlerden 3B'lu katı model elde etme konusunda ilk çalışmaları yapmıştır. Geliştirdiği algoritmada B-Rep (Boundary Representation) yaklaşımı kullanmıştır.

Bu yaklaşımın işlem basamakları:

- 1) 2B'lu noktaları 3B'lu noktalara dönüştürme,
- 2) 3B'lu noktalardan 3B'lu doğrular elde etme,
- 3) 3B'lu doğru parçalarından yüzeyler oluşturma,
- 4) Yüzeylerden 3B'lu nesneler oluşturmaktır [2].

Akbulut ve Kurt yaptıkları çalışmada uzaktan algılama yöntemiyle, fotogrametrik yöntemle elde edilmiş raster resimlerden ya da taranmış grafik paftalardan vektör bilgi elde edebilmek için iki boyutlu benzerlik dönüşümleri kullanmaktadır. Bunun için eşlenik noktalara ait konum bilgilerine ihtiyaç duyulur. Bu tür resim bilgilerinin geçerli konum bilgilerine dönüştürülebilmesi için gerekli olan kontrol noktaları bu resimler üzerinde bir grup piksel ile gösterilir. Bu piksel grubu içerisinde, kontrol noktasının merkezini en iyi temsil edenin seçilmesi dönüşüm sonuçlarının daha gerçekçi olmasını sağlayacaktır. Fakat böyle bir seçim piksel sınırlarının altına inmeyi sağlamayacaktır. Bu çalışmada, kontrol noktalarının resim üzerindeki doğruluklarını artırabilmek için, resim işleme teknikleri ve aritmetik ortalama bağıntılardan yararlanarak basit algoritmalar oluşturmaktadır [3].

Glynn ve Thorpe yaptıkları çalışmada piksel olarak okunan resmin çizgi haline nasıl dönüştürüldüğü anlatılmaktadır. Bu çalışmada Delphi dili kullanılmıştır [4].

Yaman, vd. yaptıkları çalışmada, Ankara Hızlı Raylı Sistemde, Kızılay-Ankaray istasyonunda bekleyen yolcuların, sistemde güvenlik amaçlı kullanılan kameralar vasıtasıyla algılanan gri görüntüleri, bilgisayar ortamına aktarılmıştır. Daha sonra, görüntü segmentasyon işlemleri ile nesneler arka plandan ayrılmış ve ayrılan nesnelere ait görüntüler, görüntü güçlendirme metotları ile belirginleştirilmiştir. Bir sonraki aşamada, netleştirilmiş görüntülerin gri-seviye histogramlarından nesnelere ait alansal bilgiler çıkarılmıştır. Hesaplanan yolcu yoğunluk oranı değerleri ile gözle sayılan yolcu sayıları arasındaki ilişkiler incelenerek Ankara Hızlı Raylı Ulaşım Sistemde tren sefer aralıklarının optimizasyon işlemlerine giriş verileri sağlanacak hale getirilmiştir. Elde edilen bu sayısal değerler, zaman serisi verileri olarak alınıp hafta içi yolcu yoğunluk modelleri yardımıyla modellenmesi yapılmıştır. Sonuçta bu

model kullanılarak söz konusu hizmet sisteminde dinamik çizelgeleme sürecine veri hazırlanması sağlanmıştır [5].

Akbulut ve Eser bu çalışmada Raster Bilgilerin Vektör Bilgilere Dönüşümünü yapmışlardır. Raster bilgiler piksel olarak taranmıştır. Veriler vektör verilere dönüştürülmüştür [6].

Silver yaptığı çalışmada dijital görüntünün işlenmesi için kişisel bilgisayarda pikselleri taramış dijital bellekte depolamıştır. Sonra işlemde geçirerek analiz etmektedir [7].

Matlab adlı analiz programında resim formatları taranıp istenilen değerler elde edilmektedir. Bu veriler daha sonra çeşitli yöntemlerle kullanılmaktadır [8].

Kılıçarslan çalışmasında Vp HybridCAD adlı bir yazılımı kullanmıştır. Kâğıt ortamındaki proje herhangi bir tarayıcı vasıtası ile bilgisayar ortamına aktarılır. Bu işlemde siyah-beyaz dökümanlar, tif formatında renkli dokümanlar ise jpg, bmp olarak kayıt edildikten sonra Vp HybridCAD programı dosyayı açıp yorumlamaktadır. Öncelikle görüntü üzerinde bulunan kirli bölgeleri, ister manuel ister otomatik olarak temizlenmektedir. Bu işlemi bölgesel olarak yapabileceği gibi bütün çizim alanında da uygulayabilir. Bu işlemin ardından projede gerekli olan bölgeler manuel olarak düzenlenebilir, ardından vektör veri haline dönüştürebilir. Bu işlem otomatik olarak vektörizasyon işlemine tabi tutulmak isteniyorsa öncelikle yapılması gereken proje üzerinde bulunan dairelerin, yazıların, yayların yaklaşık ölçüleri belirlenerek bir parametre listesi oluşturulmaktadır. Bu liste doğrultusunda otomatik olarak vektörizasyon işlemi gerçekleştirilmektedir. Vektörizasyon işlemi sonrasında proje katmanlara ayrılmış hale gelecektir. Bu işlem neticesinde vektör veri üzerinde blok oluşturma, sembol tanımlama işlemleri yapılabilmektedir [9].

Başak çalışmasında ActiveX Automation kullanarak AutoCAD ortamındaki komutlara nasıl ulaşılabileceğini açıklamıştır. ActiveX komutları ile Visual Basic

programında koordinatları verilen nesneleri AutoCAD programına göndererek çizim nesnesi oluşturmuştur [10].

Pal çalışmasında resimleri piksel olarak işlemiş ve işlenen kısımları hafızaya alıp resmin geriye kalan kısımları üzerinde çalışmıştır [11].

Raphael çalışmasında sanat resimlerini piksel olarak işlemiştir [12].

Yapılmış akademik çalışmalarda sadece konumuzla ilgili olarak tif formatındaki bir resmin pascal programı yardımı ile koordinatları çıkarılmış olup çok az akademik çalışma yapılmıştır. Visual Basic 6.0 programlama dili haricinde daha gelişmiş diller kullanılmıştır. Diğer çalışmalarda ise resim formatları arasında dönüşüm yapılmıştır.

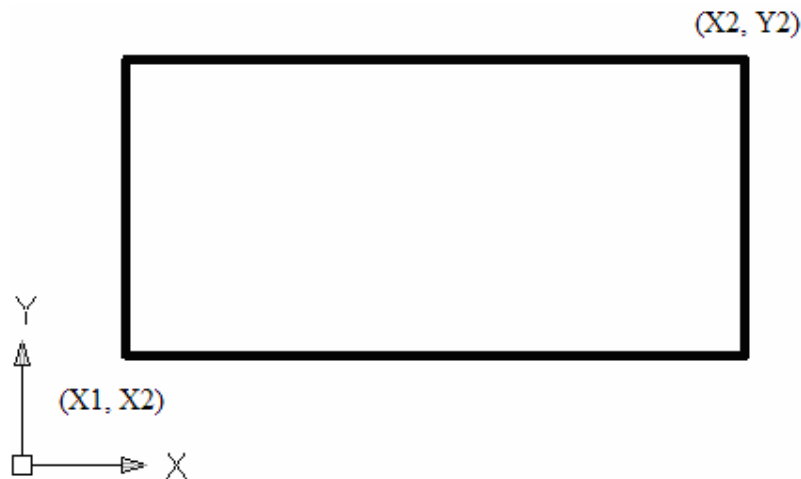
3. GRAFİK FORMATLARI

3.1. Vektör Grafikler

Görüntü formatları iki şekilde depolanabilir. Vektör tabanlı ve piksel tabanlı olarak kaydedilir. Vektör grafikler matematiksel olarak tanımlanan nesnelerden meydana gelir. Vektör tabanlı görüntülerin avantajı istenilen ölçeklerde büyütme – küçültme yapılabilmesidir. Raster grafiklerde ise görüntü piksel piksel oluşturulur ve kaydedilir.

Raster grafikler ise piksel veya noktaların oluşturduğu ızgaralardır. Bu grafikler bitmap olarak bilinmektedir. Piksel tabanlı görüntülerin dosya boyutu büyütme oranı arttıkça artmaktadır. Vektör grafikleri genelde rahatça piksel tabanlı grafiklere dönüştürülebilmektedir, tersi ise her zaman kolay değildir. Vektör grafikleri içerisinde pikseli grafik kullanmak mümkündür ancak tersi şu an mümkün değildir.

Vektör grafikler: Doğrular, eğriler, noktalar, vb. kullanılarak resim oluşturma biçimidir. Vektör olarak adlandırılmakla birlikte ayrıca pozisyon bilgileri ve renk bilgileri de içermektedirler. Şekil 3.1’ de vektörel bir resmin birinci köşe noktası (X_1, Y_1) olan ve ikinci köşe noktası (X_2, Y_2) olan bir dikdörtgeni gösterilmektedir.



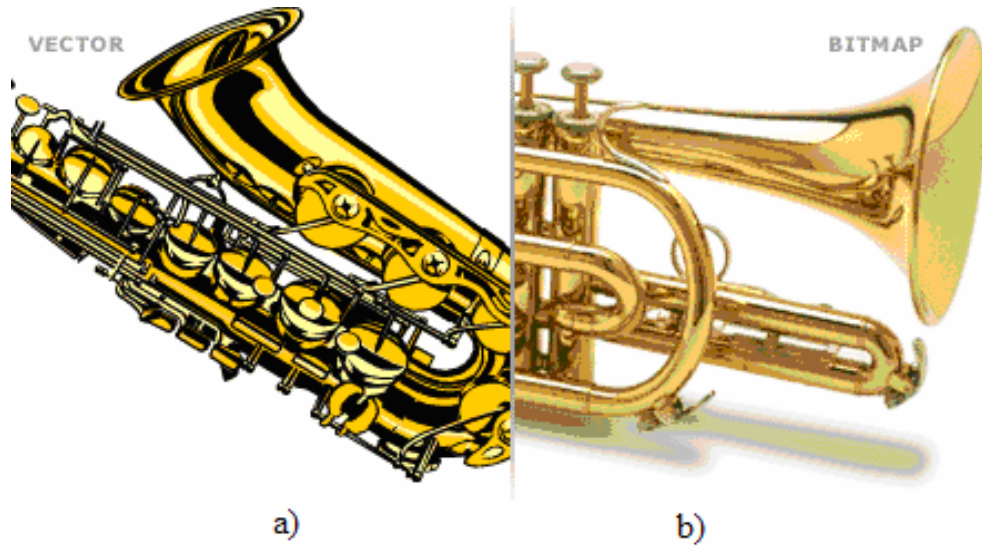
Şekil 3.1. CAD programında çizilmiş vektör tabanlı resim

Bu grafikde ilk önce (X1, Y1) noktasında oluşan köşe ile (X2, Y2) noktasında oluşan köşe noktaları birleştirilerek dikdörtgen çizilmiştir. Dosya yapısı kalabalık olmadığı için dosya boyutları küçüktür.

Vektör tabanlı resimlerde nesneler için matematiksel formüller kullanılır. Ayrıca çizgi tipi ve rengi gibi bilgiler de bulunur.

Vektör grafikler özel çizilmiş şekillerden oluştuğu için her şekil (çizgi, daire, yay, nokta, çokgen vb.) tek renge sahiptir (Şekil 3.2).

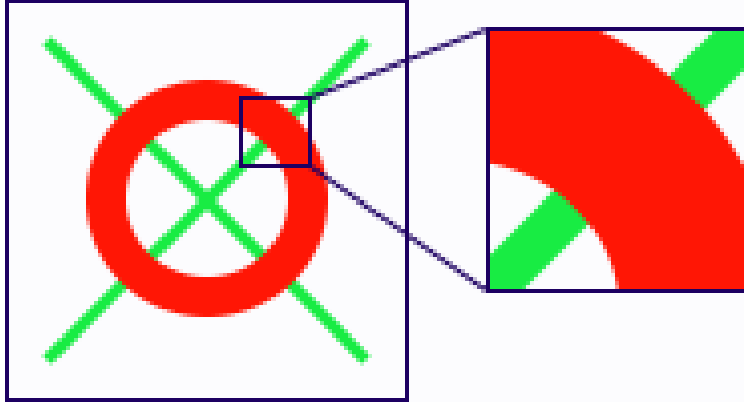
Büyültme veya küçültme işlemlerinde çizim yapıları matematiksel olduğu için bilgisayar tarafından yeniden çizilmektedir. Vektör resimlerin görünüş özellikleri farklıdır. Resmi oluşturan her nesnenin kendine ait rengi bulunmaktadır.



Şekil 3.2. Vektör, Bitmap resim arasındaki farklar

- a) Vektör tabanlı resim
- b) Bitmap tabanlı resim

Vektörel resimler eğilip, bükülebilen, boyanabilen çizgilerden meydana gelen, az yer kaplayan ve küçültme büyültmelerde hiç bir şekilde bozulma yapmayan formatlardır [21].



Şekil 3.3. Vektör tabanlı grafiklerin görüntü oranı

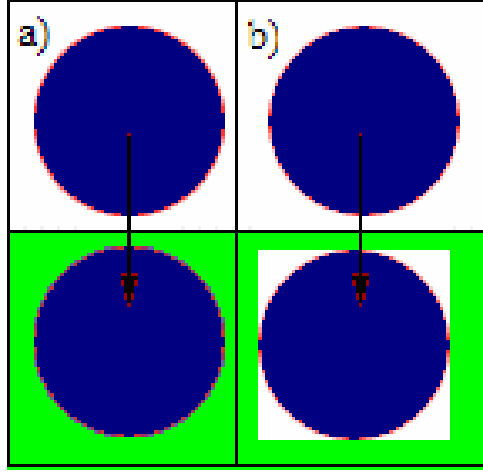
Örnekte görüldüğü gibi şeklin belli bir kesimi 10 kat büyütülerek bakıldığında bile resim bozulmaz (Şekil 3.3).

Çözünürlükten bağımsız olan, ölçeklenebilir vektör grafikler genelde CAD programları, çizgi film, karikatür benzeri resimlerde kullanılıyor olsa da birçok farklı amaca hitap etmektedirler. Bu tip grafikleri kaydedebileceğimiz başlıca formatlar şunlardır:

AI (Adobe Illustrator), CDR (CorelDRAW), CMX (Corel Exchange), CGM Computer Graphics Metafile, DRW (Micrografx Draw), DWG AutoCAD, WMF (Windows Metafile).

Çözünürlüğe göre resimler yeniden meydana getirildikleri için ekranda iyi gözükmenin dışında kâğıda çıktı alırken de başarılı sonuçlar sağlamaktadırlar (Şekil 3.4).

Bu tip grafikler metinleri de, özellik olarak barındırdığından resim içindeki yazılar da kaliteli olarak resimlenir. Ayrıca çizilen şekillerin arka planları olmak zorunda değildir. Vektör grafikler dörtgenlerle sınırlandırılmamıştır. Başka nesnelerin üzerine yerleştirebilen bu grafiklerin boş bölümlerinde diğer nesneler resimlenebilir.



Şekil 3.4. Resim formatlarının çözünürlükleri
a) Vektör tabanlı b) Bitmap tabanlı resimler

Bitmap biçimde kaydedilen bir dairenin dörtgen bir çerçevesi mutlaka olacaktır. Bu çerçevenin boş alanları arka plana uymadığında istenmeyen resimler ortaya çıkabilir. Ama vektör grafiklerde böyle bir sorun yaşanmaz.

Vektör grafikler yazılımlar sayesinde elde edilir. Bir yazılım kullanmadan tarayıcıdan aktarılan resimler vektör grafik olarak kaydedilemez. Öte yandan vektör grafikleri bitmap'e çevirmek oldukça kolaydır.

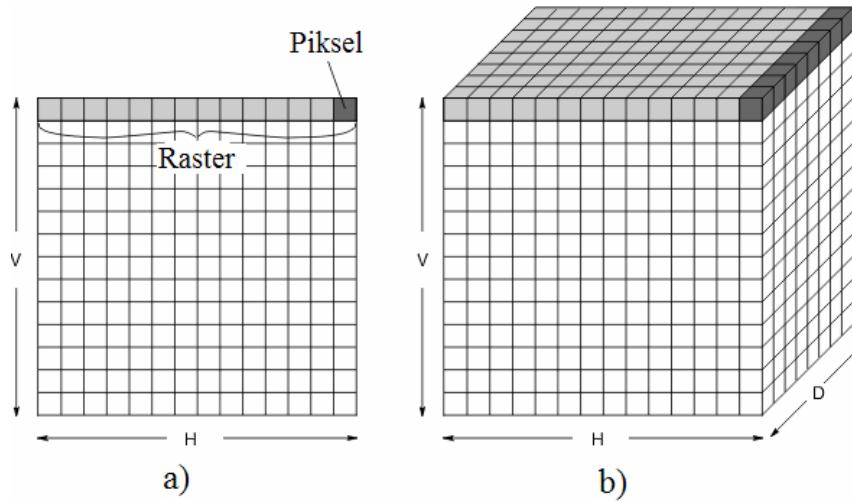
Office dokümanlarına eklenen clip art resimleri genelde wmf uzantılıdır. Metafile olarak adlandırılan wmf ve emf uzantılı dosyalar da bir nevi vektör grafik dosyalarıdır. Farklılıkları ise, vektör komutlarıyla beraber içlerinde bitmap Şekilresimleyebilirler. Örneğin vektör olarak çizilmiş bir pencerenin dört parçasından birinde yine formüllerle belirtilmiş özellikler yerine dışardan çağrılmış bir bitmap resim bulundurabilirler.

Svg henüz yaygınlaşmamış olan ama üzerinde yoğun çalışmalar yapılan bir vektör grafik formatı da xml teknolojisinden faydalanan Svg (Scalable vector graphic) formatıdır. Gif ve Jpg dosyalarına alternatif olması için geliştirilen başlıca formatlardan biri olan Svg dosyaları bir metin editöründe açılıp incelenebilir. Daha

çok Gif dosyalara alternatif olması beklenen Svg, fotoğraflık olmayan, sınırlı sayıda renk tonu barındıran resimlerde hem Gif hem de Jpg' den küçük boyutlarda kaydedilebilmektedir. Bu formatın en heyecan verici yanı ise, internet için hazırlanan bu yeni resim biçiminin html ve xml'in nesne kavramlarına uygun olması ve de başka grafik dosyaları ile komutlara değer verip alabilmesidir. Svg ile internet'ten sadece formülün indirilip, resimin oluşturulması kullanıcının bilgisayarına bırakılınca sayfalar daha hızlı gelecektir. Bu formatın çok daha derin renk özellikleri, mükemmel piksel konumlandırması, daha başarılı çıktılar sağlaması, sayısız yazı tipini barındırıp hatta içindeki metinlerin aranabilmesi söz konusu olmaktadır [21].

3.2. Bitmap (Raster) Tabanlı Grafikler

Raster resimler bir ızgara sistemi içerisindeki piksel adı verilen bölmelerin her birinin istenilen renk de boyanması ile elde edilmektedir. Bir bitmap resim ızgara yani kareler üzerine yerleştirilmiş her bölmenin (pikselin) renk bilgisinin, yerinin koordinat değerinin tarif edilmesi ile oluşturulmaktadır. Bir duvara yerleştirilen fayansların bir araya gelerek desen oluşturması gibi pikseller bir araya gelerek resmi meydana getirir (Şekil 3.5).

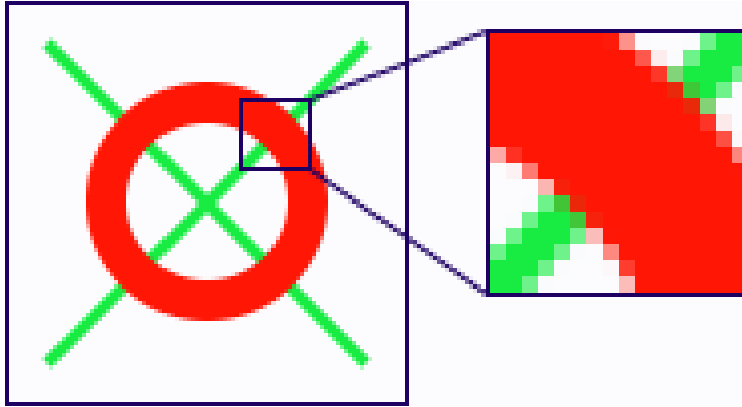


Şekil 3.5. Bitmap piksel yapısı

- a) İki boyutta
- b) Üç boyutta

Aynı monitörümüzün her piksel için ayrı resim bilgileri alıp bunları tek tek bize göstermesi gibi bitmap grafikler de resimdeki her pikselin özelliklerini kaydeder. En basit bitmap dosyası sadece iki renk yani siyah ve beyaz bilgilerini barındırır ve bu dosyalar her piksel için tek bir bit bilgi kullanır. Bu tip monochrome dosyalar bazen gri ölçekli resimlerle karıştırılır. Ama greyscale olarak adlandırılan bu tip dosyalar aslında boyut olarak çok daha fazla yer tutarlar, zira her piksel için oldukça fazla bilgi barındırırlar (Şekil 3.5).

Bitmap tabanlı grafiklerin sabit yapıları vardır. Çünkü yapıları ızgara içerisine yerleştirilmiş küçük parçalardan oluşmaktadır. Bir Bitmap resmin görüntü kalitesi değişebilir. Resmin boyutları büyültme veya küçültme işlemine tabi tutulduğunda resmi oluşturan piksellerin boyut ölçüleri değiştiğinden görüntü kalitesinde bozulmalar oluşmaktadır. Özellikle büyültme işlemlerinde açıkça fark edilmektedir (Şekil 3.6).



Şekil 3.6. Bitmap resmin büyütüldüğündeki görüntüsü

Küçük olarak görülen sıradan bir simge aslında her noktası detaylı olarak tanımlanmış bir bitmap dosyadır. Büyütünce görülen bu noktalar dışında arka plan rengi de her piksel için ayrı ayrı tanımlanmıştır. Temelde bitmap dosyalarda dört derece renk derinliğinden biri seçilir. Tek renk, 16 renk (4 bit/piksel), 256 renk (8 bit/piksel), 16 milyon renk (24 bit/piksel). Artık 256 renk ve 16 milyon renk resimler standart olmuştur.

256 renk resimler, örneğin gif dosyaları milyonlarca renk arasından, söz konusu resimde kullandıklarını bir araya toplayarak en fazla 256 renk barındıracak bir palet oluştururlar. 640' a 480 piksel boyutlarda 8 bit bir grafik dosyası yaklaşık 300 KB tutar ve yeterli kalitede resim sunar ama 35mm'lik bir fotoğrafın kalitesini tatmin edemektedir.

$640 \times 480 = 307,200$ Piksel,

$600 \times 800 = 480,000$ Piksel,

$1280 \times 960 = 1,228,800$ Piksel [22].

Siyah ve beyaz tonlardan oluşan bir resimle gri tonlardan faydalanan bir Şekil arasında sadece kalite değil, önemli miktarda boyut farklılıkları da olmaktadır. 24 bitlik resimler ise her bir piksel için üç ana rengin tonlarını kaydetmekte, mavi, yeşil ve kırmızı için 256 'şar ayrı ton sunan bu yolla $256 \times 256 \times 256$ yani 16,7 milyon renge ulaşılmaktadır. Çoğu 24 bitlik Şekilinde 32 bit olarak kaydedilmekte, ekstra bir bit de o noktanın alpha değerini yani efekt özelliğini barındırmaktadır. 640×480 boyutlarında bir resim yaklaşık 920 KB yer tutmaktadır.

Bitmap grafikler çözünürlükten bağımsız olmadıkları için farklı boyutlarda aynı kalitede Şekil veremeyebilirler. Resmi düzenlerken eğer küçültülürse bazı piksellerin bilgileri kaybolabilir. Eğer büyütülürse yeni oluşturulan pikseller çevresindekilere bakılarak tahmin edilmeye çalışılacaktır. Bu nedenle resimler yazıcıdan basılırken beklenmeyen boyut ve kalite değişiklikleriyle karşılaşılabilir. Bitmap resimler, piksellerin diziliminden oluşmaktadır.

Bir bitmap dosya yapısı: Her bitmap resmi bir bitmap başlığı, bitmap bilgi bölümü, renk tablosu ve bitmap bit karakterlerinin oluşturduğu Byte değerinin düzen bilgisini içermektedir. Bir bitmap dosyası belirli bir dosya yapısına sahiptir;

Bitmap dosya başlığı: Bitmap dosyası başlık kısmı dosyanın tipi, dosya boyutu ve Bitmap dosyasının düzenlenmesi hakkında bilgi içermektedir. Başlık kısmı bir Bitmap dosyasının yapısını tarif etmektedir.

Bitmap başlığı: Bilgi kısmı Bitmap dosyasının ölçüsel boyut özelliklerini, sıkıştırma tipini, Bitmap renk formatı bilgilerini içermektedir.

Renk tablosu: Renk tablosu Bitmap’ deki her pikselin renk bilgisini içermektedir. Bitmap renk tablosu her Bitmap için 24 bit rengi vermez. Çünkü her piksel gerçek Bitmap bilgi alanındaki Kırmızı-Yeşil-Mavi (Red Green Blue-RGB) nin 24 bit değerini içermektedir. Renklerin veya resmin kayıt edildiği bit değeri değişiklik gösterebilir. Buda renklerin görüntü netliği veya sayısını etkilemektedir. Bitmap de renklerin diziliş sırası çok önemlidir. Bitmap formatındaki birçok resim gerçek görüntülerini sürücülerini yükledikleri zaman gösterilebilmektedir.

Dosya Byte değeri: Bitmap dosyalarının boyutları ebat ölçüleri, renk değeri, bit derinliği ve sıkıştırma oranlarına göre değişmektedir. Kısaca dosya boyutunun değeri ebat ölçülerinin ve bit değerinin çarpımı ile elde edilmektedir [22].

Bu resim tanımlamada her bir hücrenin bir değeri vardır. Başlıca bitmap tabanlı dosya formatları şunlardır:

BMP, GIF, JPEG, JPG, PNG, PICT (Macintosh), PCX, TGA, TIFF, PSD.

3.3. Başlıca Bitmap Tabanlı Formatlar

3.3.1. Bmp veya dib (Bitmap)

Windows Bitmap formatı, Microsoft'un DOS ortamına grafikleri taşırken kullandığı, Windows ve DOS ortamına özel başlıca araçlardan biridir. Renk paleti sunan BMP'nin saydamlık özelliği yoktur.

Özel formatlardan Macintosh'un Paint ve Pict'i ile X-windows'un XBM formatlarına da kısaca değinmek gerekmektedir. Apple ticari olarak ilk başarılı grafik formatı sayılabilecek paint ile MacPaint programında siyah beyaz resimler oluşturulmasını sağladı.

3.3.2. Cpt (Corel photo paint)

Corel Photo-Paint'in kayıt formatlarından biri olan CPT şekilleri piksellerle betimleyerek resmi oluşturur. Coreldraw CPT dosyalarını başka dosyalara ekleyebilir ya da alabilir. Bu dosyaların renkli ya da gri ölçekli olması mümkündür. Corel photo-paint İle CPT olarak kaydedilen dosyaların mask, lens, floating objects özellikleri korunur.

3.3.3. Png (Portable network graphic)

Portable Network Graphic formatı Gif'e alternatif olarak doğdu ve çok daha fazla özellik sunmaktadır. Piksel başına 48 bit'e varan kalitede Şekilile gerçek renk sunan, gri ölçekli resimleri de piksel başına 16 bit bilgiyle kaydedebilen Png, Gif'ten daha farklı saydamlık özellikleri sunmaktadır. Genel saydamlık maskeleriyle tam alpha kanalı, gamma parlaklık özellikleri ve hızlı bir 2D interlace sunulur. Dosyadaki bozuklukları daha iyi ortaya koyar ve progressive resimleri daha hızlı yüklemektedir.

Png hem gerçek renk derinliği için 24 bit hem de Gif gibi sınırlı bir renk paletini kullanarak 8 bit formatlarda kullanılabilmekte, fazladan bir alpha kanalıyla orijinal dosyanın katman özelliklerini de saklayabilmektedir [21].

3.3.4. Psd (Photoshop document)

Grafik düzenleme programlarının en ünlüsü Photoshop'ın dosyaları kaydederken kullandığı kendi formatı da sadece bu nedenle dahi başlıca formatlar arasında yer almaktadır. Psd daha fazla yer kaplar, gerçek renk derinliğini başarıyla uygular, her zaman katman bilgisi barındırır. Sıkıştırma kullanmaz ve saydamlık özelliği yoktur.

Photoshop'da çalıştığınız projeler varsa en iyisi çalışmanızı Psd olarak tutmanız gerekmektedir. Her seferinde farklı bir formatta kaydedip bunu açmaktansa Photoshop'ın dilini en iyi anladığı Psd'yi kullanıp düzeltmeleri yapmakta fayda vardır.

3.3.5. Tga (Truevision targa)

Truevision targa sıkıştırma ile kaliteden vazgeçip küçük boyutta dosyalar elde edilebilen bir formattır. Gerçek renk derinliği ve katman bilgisi barındıran Tga saydamlık seçeneği sunmaz. Tif formatı ile çok benzer özelliklere sahip olan Tga arka planda kalmış formatlardan biri olarak tanımlanabilir.

3.3.6. Tif (Tagged image format)

Tagged image format birkaç firma tarafından iddialı olarak geliştirilen, genişletilebilen özellikler sunan bir çalışmadır. Saydamlık özelliği olmayan Tif gerçek renk derinliğinde katmanları destekler. Sıkıştırma ile iyi sonuçlara ulaşabilirsiniz.

24 bitlik resimler ise her bir piksel için üç ana rengin tonlarını kaydederler. Mavi, yeşil ve kırmızı için 256'şar ayrı ton sunan bu yolla 256x256x256 yani 16,7 milyon renge ulaşılır. Çoğu 24 bitlik Şekilasında 32 bit olarak kaydedilmekte, ekstra bir bit de o noktanın alpha değerini yani efekt özelliğini barındırmaktadır. 640x480 boyutlarında bir Şekilyaklaşık 920 KB yer tutar [21].

3.3.7. Jpeg (Joint photographic experts group)

Jpeg formatı 24-bit renk derinliği sunmaktadır. Bu da bir web sayfası için yeterli olacak kalitede fotoğraf ve benzeri resimlerin kullanılmasını sağlar. Jpg'nin en büyük artlarından biri, kalite oranının tamamen grafikçiye bırakılmasıdır.

Kaydederken sıkıştırma oranını, böylece de kaliteden ne kadar feragat edileceğini kullanıcı belirlemektedir. 32 bit renk derinliği ve 64000'e 64000 piksel boyutuna kadar çıkabilen bu format grafik formatlarının en başarılılarından birisidir. Ayrıca progressive olarak kaydedilen Jpg grafiklerin yüklenen bölümleri hemen resimlenir, böylece kullanıcı nasıl bir resimle karşılaşacağını tahmin edebilir. Tabii bu özellikteki dosyalar, az da olsa daha büyük boyutta olurlar. Sanılanın aksine, dosya

sıkıştırılırken resimde yer alan her piksel parlaklık özelliklerine göre kaydedilir. Renkler ve farklılıkları ise detaylı olarak değil, yuvarlanarak kaydedilir. Yani elimizde tam tamına, tüm detayları kodlanmış bir dosya olmaz. Bu nedenle her seferinde JPG olarak kaydedilmiş dosyayı açan program bilgiyi derler, dosya her ortamda fark edilmeyecek kadar oranlarda da olsa, değişiklik gösterilebilir.

Her seferinde hesaplanıp yeniden oluşturulan resimin kalitesini sıkıştırılma oranı belirler. Günümüz resim işleme programlarının bir kısmında sıkıştırma oranı sorulmaktadır.

Kodu açılan resimdeki renkler, yayılan bloklar olarak ekrana gelirken farklı tonlar da üstüste binebilir. Bu nedenle belirgin çizgilerle birbirinden ayrılan resimleri JPEG ile kaydedebilmek zordur ve fazla yer tutar. Özetlemek gerekirse JPEG'ler birçok rengin bulunduğu ve birbirine derece derece geçtiği, belirgin sınırlar içermeyen fotoğraflar için ideal kayıt formatıdır.

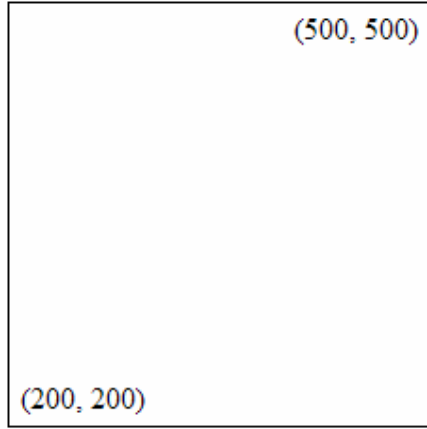
3.3.8. Gif (Compuserve graphics interchange format)

Gif olarak saklanan dosyalar fotoğraflarda ve bol renkli resimlerde Jpg'e göre daha kötü sonuçlar vermektedir. Önce benzer renklerdeki pikseller ayrı birimlere ayrılıp kaydedilir, ardından da dosya bir renge endekslenir. Bu nedenle Gif dosyaları en fazla 256 renk derinliğine sahiptir. Bazı durumlarda, Gif olarak kaydetme aşamasından önce resimdeki renk derinliğinin 256'ya düşürülmesi gerekebilir.

Sıkıştırma özelliği olmayan Gif ile kalite kaybı olmaz ancak grafiği 256 rengin üzerindeyse bu renge düşürülünce birçok özelliği kaybolabilir. Ama bu kısıtlı renk derinliği dosya boyutunu oldukça küçültebilir. Çizgi film benzeri basit çizgisel resimler, şirket logoları, imzalar Gif' in başarılı olduğu resim tiplerine sadece birkaç örnektir. Ayrıca Gif dosyası birkaç Şekilbarındırır böylece hareketli bir animasyon sunabilir. Renklerin birbirine karışmadığı kalitede şeffaflık da Gif ile mümkündür. Resimdeki bir renk şeffaf olarak atanabilir [21].

3.4. Dwg (AutoCAD drawing format) Format Yapısı

Autodesk firmasının AutoCAD yazılımı için geliştirmiş olduğu ikili sayı sistemine göre çalışan çizim formatıdır. 24 bit ve 3d desteği vardır. Değişik CAD programlarında da çalışmaktadır. C dili kullanılarak geliştirilmiştir. Dwg formatı üzerinde çizim, ölçülendirme, yazı, renk vb istenilen her özellik değiştirilebilir. Format yapısı eklerde verilmiştir.



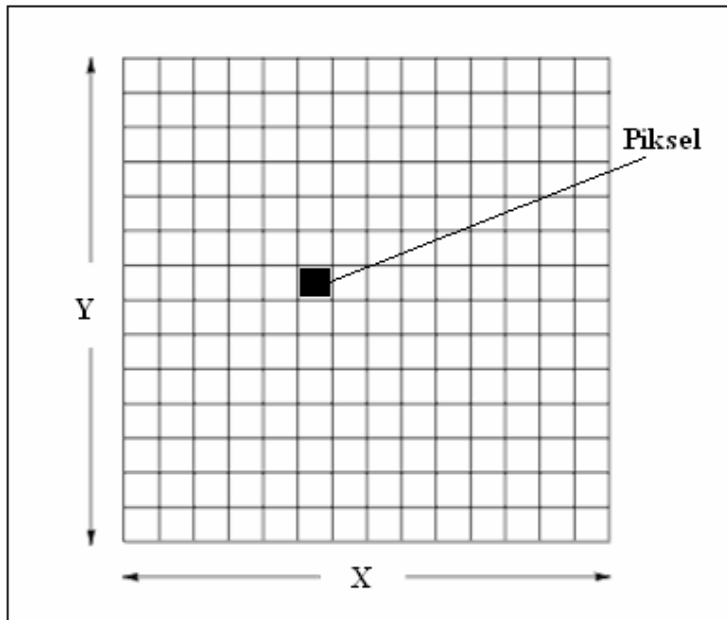
Şekil 3.7. AutoCAD programında çizilmiş Dwg resmi

AutoCAD ortamında çizilen resimler Dwg formatında kaydedilmektedir (Şekil 3.7). Dwg format yapısında Dxf format yapısında olduğu gibi değişkenler mevcuttur. Dwg format yapısı değişkenleri örneği eklerde belirtilmektedir. Tasarlanan programda ActiveX Automation kullanılmaktadır [16].

4. BMP (BITMAP) DOSYA YAPISI

4.1. Pikselin Tanımı

Bitmap dosyalarında resmin en küçük karesine piksel denir. Dijital resimler yan yana gelen pikseller topluluğundan oluşmaktadır. Dijital resim, imgenin eninde ve boyunda bulunan piksel sayısı ile tanımlanır (Şekil 4.1). Pikselin kendi başına en ve boy değeri yoktur [22].



Şekil 4.1. Piksel tanımlaması

Tek bir piksel 1x1 mm, 1x1 cm ve 3x2 mm olabilir. Aksi belirtilmedikçe piksellerin en ve boy oranı eşittir. Çözünürlük ise boyut tanımlamalarında ek olarak gereken bir kavramdır. Kendi başlarına boyut sahibi olmayan piksellere çoğu zaman bir boyut değeri tanımlamak gereklidir. Bu şekilde piksellerin boyutu belirlendiğinde uzunluk biriminde kaç piksel bulunacağı da belirlenmiş olur.

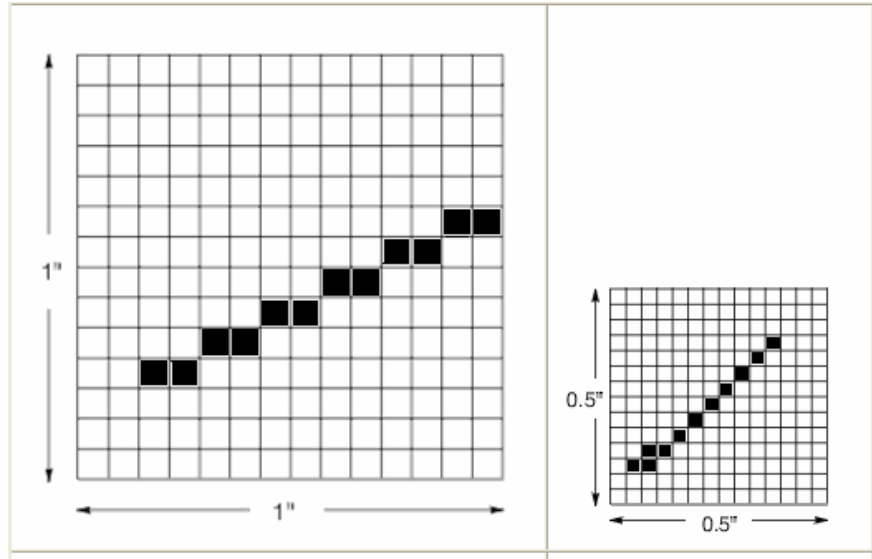
Örneğin bir pikselin boyutu 1 mm olarak tanımlanmışsa her santimetrede 10 piksel bulunacaktır.

Kapladığı alan ne olursa olsun resim birimi olan her piksel sadece tek bir renk değeri içerebilir. Dijital resim işleme programları imge üzerinde işlem yaparken inch, cm mm gibi ölçü birimlerini kullanmazlar. Tüm kesme, yapıştırma, yer değiştirme işlemlerinin tamamı piksellere uygulanmaktadır [22].

Vektörel resim dosyalarından farklı olarak noktasal resim dosyaları için çözünürlük ve içerdiği piksel sayısı çok önemlidir. Dökümandaki işlemlerin hepsi piksellerin rengi veya konumunu değiştirerek etki yapar. İşlemlerin ne kadar süreceği ise üzerinde işlem yapılan piksel sayısına ve işlemin karmaşıklığına bağlıdır. Çözünürlük hesaplarında uzunluk birimi olarak inch veya santimetre kullanılır.

Bir Inch = 25,4 mm veya 2.54 cm, basit hesaplarda 2.5 cm değerini almaktadır.

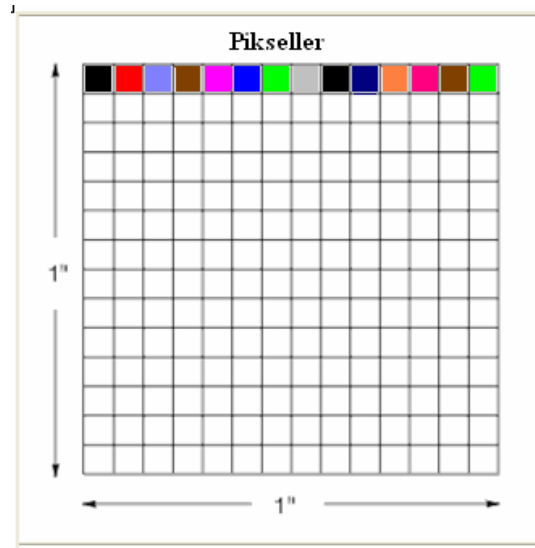
Piksellerin boyutu büyüdükçe resmin kalitesi düşecektir. Piksel boyutu küçüldükçe ise kalite yükselecektir (Şekil 4.2).



Şekil 4.2. Piksel boyutları

Çizim kalitesi arttıkça resmin dosya boyutları artacaktır. Her pikselin şekli kare biçimindedir ve her pikselin sadece bir rengi vardır. Yani bir pikselin bir kısmı açık

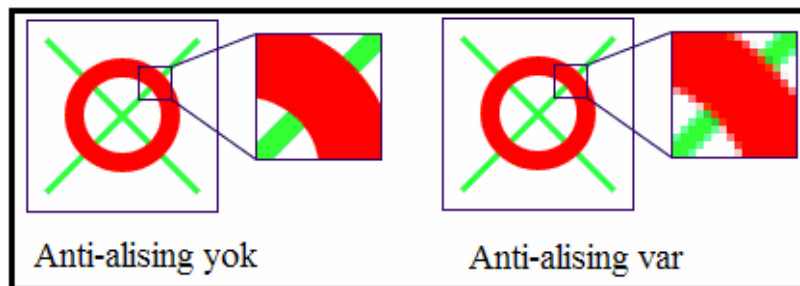
bir kısmı koyu, ya da yeşil, koyu yeşil olmaz. Bu tür renk geçişleri farklı renkte piksellerin yan yana gelmesinden oluşmaktadır (Şekil 4.3).



Şekil 4.3. Piksel renkleri

4.2. Kırık, Kademeli Görüntü (Antialiasing) Oluşumu

Pikseller kare biçimli en küçük elemandır. Resimler piksellerin düzenli ve belirli bir sıraya göre toplamından meydana gelmektedir. Bitmap resimler pikseller yan yana getirip görüntüyü oluşturmaktadır. Yuvarlak, kavisli ve çapraz gibi dönmesi gereken yerlerde kademeli bir görüntü ortaya çıkmaktadır. Bu kademeli görüntüye anti-aliasing denilmektedir [22].



Şekil 4.4. Piksel antialiasing özelliği

4.3. Renk Derinliği (Color depth)

Bit işlem resmindeki her bir piksel renk bilgisini içerir. Belirli bir kısım için bu bilgiler genellikle aynıdır. Bit işlem için çok miktarda renk bilgisi vardır. Fakat bunların uygulamalarda kullanılan belirli standartları vardır. Bunlardan sadece 24 bit anlatılacaktır.

1-bit: Monochrome (2 renk)

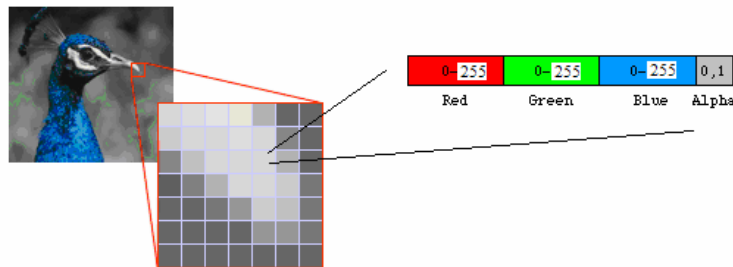
4-bit: Greyscale or colour (16 renk)

8-bit: Greyscale or colour (256 renk)

16-bit: High Colour (65,536 renk)

24-bit: True Colour (16,777,216 renk)

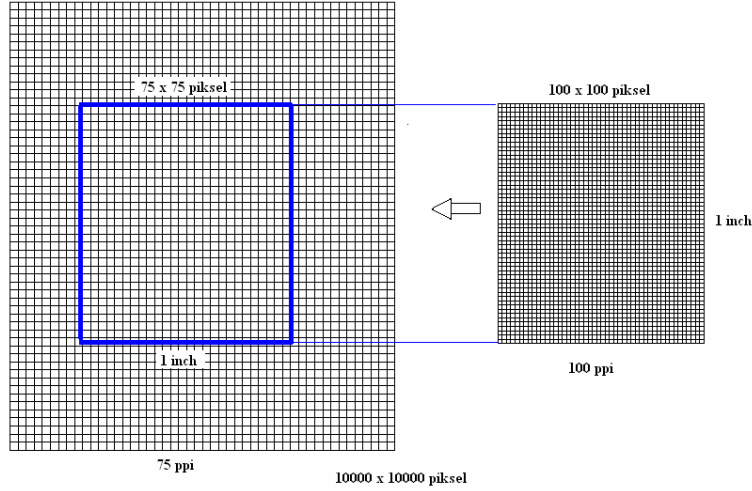
24 Bit RGB (Kırmızı, Yeşil, Mavi): Bu formatta Kırmızı, Yeşil ve Mavi renklerinin her biri 8 bitlik değerden oluşur. Her bir rengin 0 değeri o renkten hiç mevcut olmadığını ve 255 değeri de maksimum oranında bulunduğunu gösterir (Şekil 4.5). Her bir rengin 255 durumu olduğundan toplam 16,777,216 durum mevcuttur [22].



Şekil 4.5. 24 Bit renk özelliği

4.4. Çözünürlük (Resolution)

Çözünürlük, vektörel veya bitmap resimler üzerindeki belirli bir kare ölçüde üzerine düşen nokta veya piksel sayısını verir. Görüntüyü alan ve gönderen her makine (yazıcı, tarayıcı, monitör) kendine ait bir netliğe sahiptir (Şekil 4.6).



Şekil 4.6. Çözünürlük

PPC, PPI: Pixel Per (Cm) Inch: Genellikle DPI ile karışır ancak ikisi farklı birimlerdir. Piksel/(cm) inch, bir (santimetreye) inch'e düşen piksel sayısıdır. Bu, resmin öznitelikleri ile ilgili bir kavramdır.

Diğer ölçümler gibi, yazıcı veya tarayıcılar ile doğrudan ilgili değildir. Resimde birim olarak piksel/inch veya piksel/cm kullanılması yaygındır.

10 cm uzunluğunda ve 20 cm enindeki bir resim, 100 ppc çözünürlüğe sahip ise, boyunda $10\text{cm} \times 100\text{ppc} = 1.000$ piksel, eninde ise $20\text{cm} \times 100\text{ppc} = 2.000$ piksel vardır.

Resimdeki toplam piksel sayısı $\text{en} \times \text{boy} = 2.000 \text{ piksel} \times 1000 \text{ piksel} = 2.000.000$ pikseldir.

Günümüz dijital resimlerinin hepsinin temelinde kare şeklinde, resmin parçalanamaz birimi olan piksel yatmaktadır. Bu pikseller yan yana geldikçe ana resim oluşmaktadır. Resim boyutu bu nedenle bilgisayar için hep piksel olarak hesaplanmaktadır. Eninde 2, boyunda 3 piksel olan bir resimde toplam $2 \times 3 = 6$ piksel vardır. En ve boy oranları arttıkça piksel sayısı ve dosya boyutu kenarların çarpımı kadar artmaktadır [22].

Piksellerin kendi başlarına bağımsız enleri ve boyları yoktur. Örneğin 2x3 piksel boyutundaki bir resim 2x3 cm olarak basılabiliyorken, 2x3 metre veya 2x3 mm olarak da basılabilir.

Örneğin 1 cm başına 2 piksel düşüyorsa resim çözünürlüğü 2 piksel/santimetredir. Bu değer ppc veya "pixel per centimeter" olarak ifade edilmektedir.

Lpi (Line Per Inch): Ofset baskıda tramlarla resim basılır. Tramlar, birbirine özel açılar ile dizilmiş noktacıklardır ve sonuçta aralarında belli bir miktar boşluk vardır. Paralel çizgiler halinde sıralandıklarını düşünebilir. Tramlar yardımı ile farklı renkteki mürekkepler kâğıda aktarılır. Bu çizgilerin yoğunluğu ve aralığı baskı kalitesini belirler. En çok kullanılan tram aralıkları, 175, 150, 133, 120, 110, 100, 85 ve 65 LPI'dir.

3. hamura baskı yani 40–50 lik tram = 110 – 130 LPI

1. hamura baskı yani 60 lık tram = 150 LPI

Kuşe kâğıda baskı 70 lik tram = 175 LPI

Kuşe kâğıda baskı 80 lik tram = 200 LPI

Dpi (Dot Per Inch) : Çıktı cihazlarında 1 inç'te (2.54 cm x 2.54 cm) noktalanın (yani basılan) ya da pozlanan piksel sayısıdır.

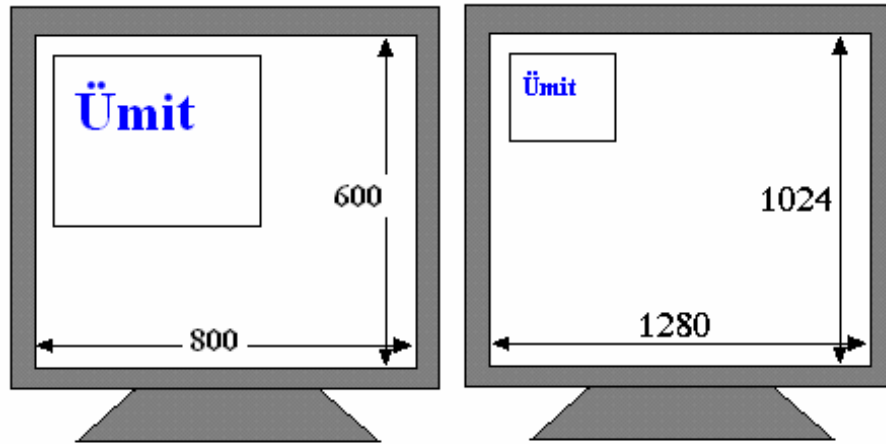
Renkli/siyah-beyaz baskı yapan yazıcılar çıktıyı küçük noktalar ile yaparlar. Her inç'e noktalanın piksel sayısı yazıcının özelliğine göre değişir. Çünkü her yazıcının kullanabileceği en yüksek nokta yoğunluğu farklıdır. 300 DPI, 600 DPI, 1200 DPI vb. CMYK adı verilen Cyan, Magenta, Yellow ve Black ile kâğıdın üzerinde küçük mürekkep noktacıkları basar, bunlar da bilinen renkleri oluşturur. Film çıkış cihazları da çözünürlüğü DPI'e, yani bir Inch başına konan nokta sayısına göre hesaplar. İnç başına düşen nokta sayısı çok azalırsa kalite kaybı olur.

DPI = 16 x LPI değerinde olması resimdeki renklerin, yani oradaki gri dağılımının sağlıklı basılmasını sağlar [22].

Çizelge 4.1. Çözünürlük eşlemeleri

Çözünürlük eşlemeleri				
LPI	PPI	PPC	DPI	DPC
150	300	240	2400	950
85	150	50	1200	480
40	80	30	600	240
20	40	15	300	120

Kullanılan bilgisayar monitörün çözünürlüğü ne kadar fazla ise ekrandaki şekiller o oranda küçülür. Çözünürlüğün fazla olması piksel sayısının fazla olması anlamına gelmektedir. Dolayısı ile iyi bir görüntü demektir (Şekil 4.7).



Şekil 4.7. Ekran çözünürlüğü

Tarayıcılarda mantık ise taranacak nesne, satır satır ışığa duyarlı araçlar tarafından taranarak sayısallaştırılır. Tarayıcının daha iyi yapılabilmesi için nesne, bir ışık kaynağı ile aydınlatılır. Taranması istenen görüntü, üzerinden ışık kaynağı geçtikten sonra bir mercek aracılığıyla, fotoelektrik hücrelerden oluşmuş bir görüntü algılayıcı üzerine düşürülür. Yani, ışık değerleri ölçülür, ölçüm değerine göre bir voltaj değeri üretilir. Değişen voltajlarda elektrik impuls üreten bu algılayıcı, daha ışıklı ve daha açık tonlardaki desenleri yüksek voltajla, koyu desenleri ise düşük voltajla gösterir.

Analog voltaj sinyali, bir tür modem gibi işleyen analog-sayısal dönüştürücü yongası ile sayısallaştırılarak bilgisayarın belleğine aktarılır. Sinyaller, görüntü dosyası formatında disk ortamına kaydedilir. Daha sonra bu dosya üzerinde görüntü programları ile işlem yapılabilmektedir.

Monitörler ve yazıcılar gibi, tarayıcılarda da görüntüler küçük noktalarından oluşur. Birim alana düşen nokta sayısı ne kadar yüksekse, elde edilen görüntü o kadar kaliteli olacaktır. Profesyonel olarak kullanılacak bir tarayıcının çözünürlüğü normalden (en az 2400 dpi) yüksek olmalıdır.

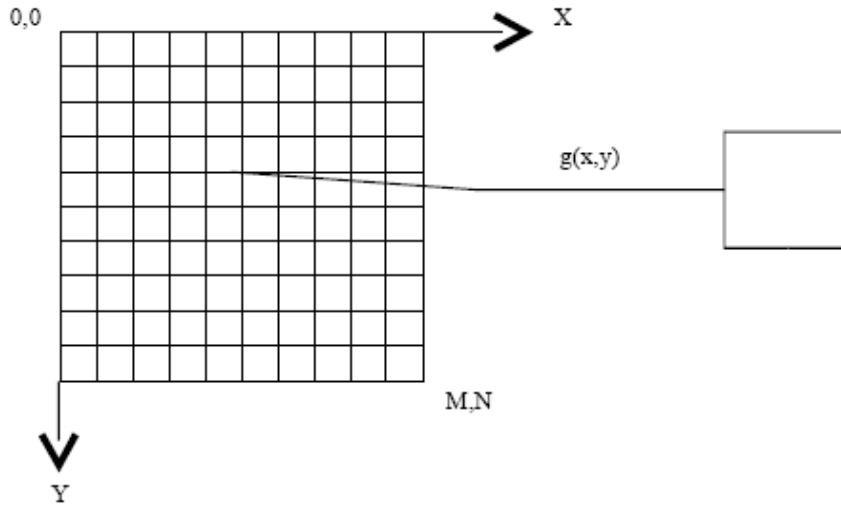
Normalde ise sadece ekranda görüneceği için daha düşük çözünürlükte kullanılabilir. Tarayıcılar, sadece çözünürlüklerine göre değil, algılayabildikleri renk sayısına göre de farklılık göstermektedirler. Renkli görüntüler bilgisayarda çok daha fazla yer tuttuğu için, genellikle ve sıkıştırma programları kullanılmaktadır. Ya da çeşitli resim formatlarında saklanmaktadırlar [22].

5. RESİM İŞLEME

Herhangi bir resim üzerinde piksel değerleri ile ilgili işlemler yapmaya ve bu işlemler sonucu oluşan bir takım sonuçlara bağlı olarak resim üzerinde bazı değişiklikler yapmaya kısaca resim işleme diyebiliriz.

Birçok uygulama içinde resim işleme yöntemleri kullanılmaktadır. Uygulamalar bilgisayarın kaynaklarını zorlamakta her zaman daha fazla hıza ve belleğe ihtiyaç duymaktadır. İşlemlerin özellikle gerçek zamanlı gerçekleştirilmek durumunda olması günümüzdeki en büyük sorunlardandır. Çalışmada resim işleme uygulamalarının nasıl yapılacağı ve geliştirilmesi üzerinde durulmaktadır.

Uygulamalarda genellikle filtreleme, gri renk skalasına dönüştürme gibi çalışmalar yapılmaktadır. Resim işleme uygulamaları öncelikle işlenecek resmin elde edilmesi ile başlar (Şekil 5.1).



Şekil 5.1. Raster görüntünün ekranda gösterilmesi

Uzaktan algılama yöntemi ile elde edilmiş görüntülerden, fotogrametrik yöntemle elde edilmiş taranmış (raster) fotoğraflardan ya da taranmış grafik paftalardan vektör bilgi elde edilmektedir. Böyle taranmış görüntülerden elde edilen vektör bilgiler

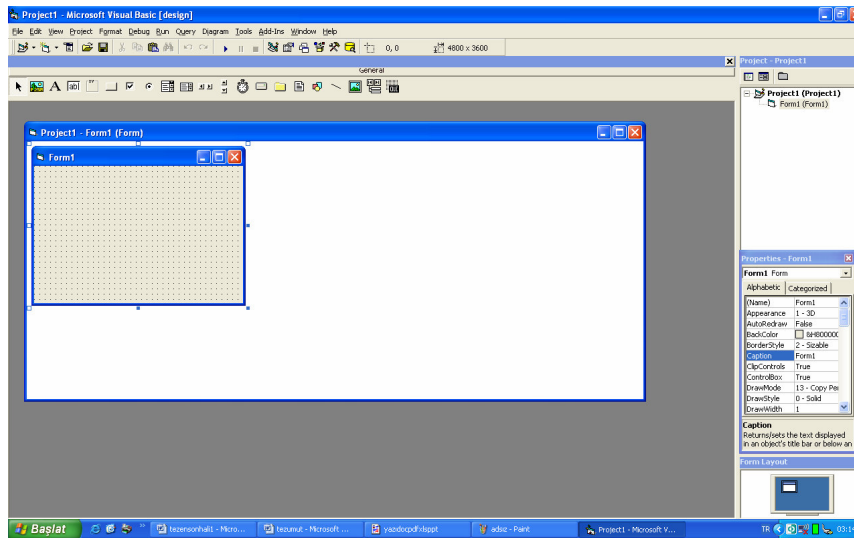
doğrudan kullanılamazlar. Taranmış görüntülerden elde edilen bilgilerin geçerli konum bilgilerine dönüştürülmesi gerekmektedir. Bu dönüşüm işlemi taranmış görüntü üzerinde yer alan ve gerçek konum bilgileri bilinen kontrol noktaları yardımı ile gerçekleştirilir. Taranmış görüntülerden kazanılan bilgilerin doğruluğu, öncelikle bu kontrol noktalarının görüntü üzerindeki konumlarının mümkün olduğunca iyi belirlenmesine bağlıdır.

Çünkü kontrol noktalarının görüntü üzerindeki konumlarında yapılacak bir hata, sistematik olarak diğer bütün noktalara yansırken, ayrıntı noktalarındaki hatalar rasgele özellik göstereceklerdir. Kontrol noktalarının (ve ayrıntı noktalarının) doğruluğu elde edilen görüntünün çözünürlüğü ile doğrudan ilgilidir. Çözünürlüğü etkileyen en önemli etkenler; görüntünün elde edildiği yöntemin fiziksel yapısı, görüntünün taşıdığı bilgi (siyah beyaz) ve görüntünün büyüklüğüdür (taşıdığı piksel sayısı).

Görüntü nokta koordinatlarını belirlemek amacıyla bir uygulama gerçekleştirilmiştir. Geliştirilen yazılım koordinat noktalarının görüntü üzerindeki konumunu, renk değerini kapsayacak bir renk aralığı içinde araştırarak gerçekleştirmektedir. Aynı program görüntü koordinatlarını Visual Basic ve Visual Basic Application programlama dili kullanarak Dwg ve Dxf koordinatlarına dönüştürerek görsel olarak sergileyebilmektedir [1-2-3-5-7-11-12-13-14, 20].

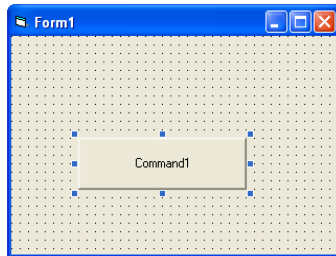
6. MICROSOFT VISUAL BASIC

Görsel olarak program yazmaya yarayan tüm dünyada kullanılan bir programlama dilidir. Visual basic ile görsel programlama yapılabilir. Önce tasarım ekranına Command, Text, Label gibi nesneler yerleştirilir. Daha sonra nesneler ile ilgili program satırları yazılarak program oluşturulur. Resim işleme programımızda Şekil 6.1’de olduğu gibi Vb6 programlama dili ekranı kullanılmaktadır [15].



Şekil 6.1. Visual Basic ekranı

Ekrana yerleştirilen nesnelere kullanıcının seçimine bağlı olarak olaylar tanımlanır. Örneğin Command butonu nesnesinin yapacağı olay tıklanmasıdır. Butona tıklandığı zaman olay gerçekleşir (Şekil6.2).



Şekil 6.2. Command butonunun tıklanması

Private Sub Command1_Click()

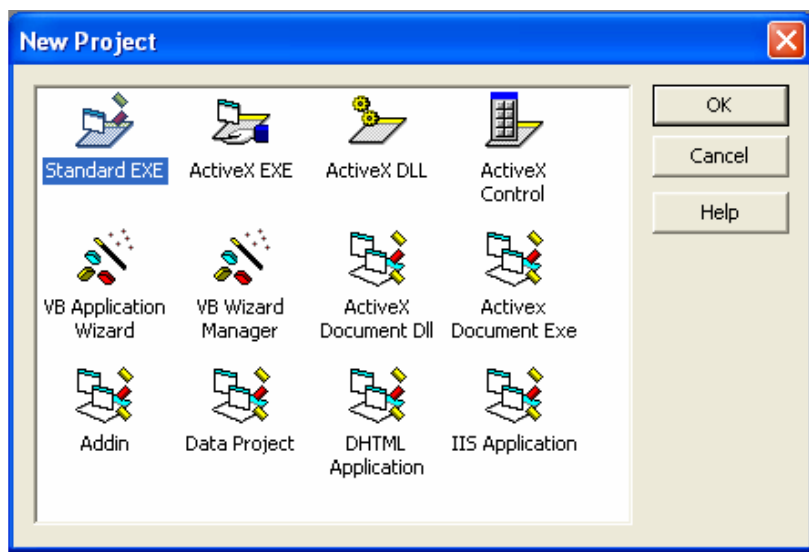
Print "Butona tıklandı"

End Sub

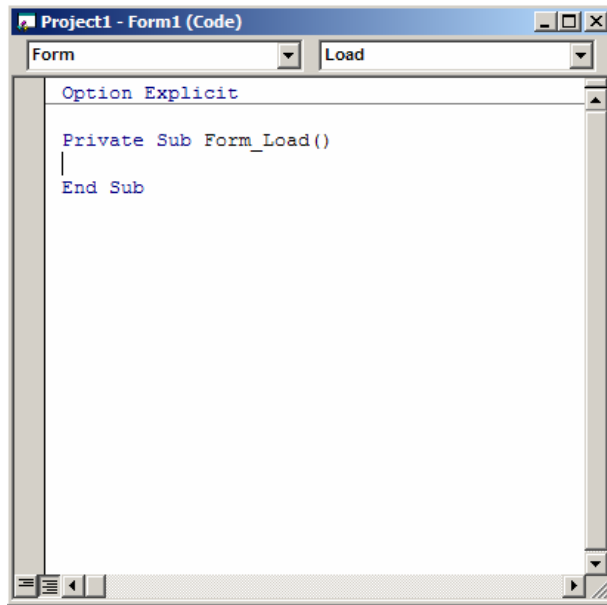
Bu üç satırlık kod, Command1 isimli butonun click (tıkla) olayı sonucunda çalışacaktır. Kullanıcı programa tıkladığı zaman olay gerçekleşir. Ekranı “Butona tıklandı” yazdırılır. Nesne ile olay arasındaki bağlantıyı ilk satırda geçen Command1_Click () ifadesi belirtir.

Program başlatıldığı zaman geliştirilecek proje çeşidine göre seçim yapılmalıdır. Visual Basic çalıştırıldığında ekrana bir diyalog penceresi gelir. Bu diyalog penceresinde 3 adet sekme bulunur. New sekmesinde oluşturulmak istenilen yeni proje için alternatifler bulunmaktadır.

Genellikle StandardEXE seçeneği seçilerek yeni bir projeye başlanır (Şekil 6.3). Eğer istenirse diğer seçeneklerde kullanılarak ActiveX denetimleri, Dll dosyaları, DHTML sayfalar oluşturulabilir. Existing sekmesi ile daha önceden oluşturulmuş projeler sürücü ve klasör seçimi yapılarak açılabilir.



Şekil 6.3. Seçilebilecek proje çeşitleri



Şekil 6.4. Program editörü

İyi bir programcının, kullandığı derleyicinin editörünü çok iyi tanıması gerekir. Visual Basic’de bu editöre Code Editör adı verilir. Code Editör’ü bir form üzerine veya nesneye çift tıklanarak veya sağ click yaptıktan sonra View Code seçeneği kullanarak açabilir.

Code Editor, Project Explorer penceresinden bir öge seçimi yapıldıktan sonra View Code butonuna tıklanarak da açılabilir [15].

6.1. Kullanılan Olay ve Nesneler

Form nesneleri Visual Basic uygulamalarının temelidir. Geliştirilen tüm uygulamalarda en az bir form olacaktır. Formlar üzerine nesneler yerleştirilir ve editöre geçilerek program yazılır.



CommandButton: Kullanıcının tıklayarak bir işlemi seçmesi için komut butonu kullanılır. Komut düğmelerinin en önemli iki özelliği Name ve Caption’dur. Name özelliği nesneyi projede kullanılan diğer kontrollerden ayırt edebilmemizi

sağlar. Caption özelliği ekranda görülecek etiket belirler. Caption özelliği verirken & karakterini kullanarak istenilen harfin altı çizgili gösterilir.

Komut düğmelerinde en çok kullanılan özelliklerden biride Default ve Cancel özellikleridir. Default özelliğine True değeri verildiğinde form üzerinde Enter tuşuna basılması ile komut düğmesinin click olayı meydana gelir. Cancel özelliğine True değeri verilmesi ile ise form üzerinde herhangi bir denetim üzerinde ESC tuşuna basıldığında komut düğmesinin click olayı meydana gelir.

Komut düğmelerinde en çok kullanılan olay ise Click olayıdır. Kullanıcı komut düğmesi üzerine tıkladığı anda gerçekleşir.



TextBox: Genellikle kullanıcıdan veri toplamak amacı ile kullanılır. Metin kutularının caption özelliği bulunmaz bunun yerine text özellikleri vardır. Metin kutusu içerisine yazılan değeri görmemizi sağlar.



Label: Etiketlerde metin kutularına benzerdir, aralarında fark ise kullanıcı etiketlere değer giremez. Çok fazla özelliği yoktur. Caption özellikleri kullanılarak içlerine yazılar yazılabilir. Daha çok metin kutuları ile birlikte kullanılır.



PictureBox: Resim kutuları genellikle grafikleri (örneğin bmp, jpg, gif) resimlemek için kullanılır. Eğer sadece bir resim göstermek isteniyorsa resim kutusu daha iyi bir seçimdir. Grafiğin form üzerinde taşınması isteniyorsa resim kutuları daha esnektir. Resim kutuları içerisine çizim yapılabilir.



Frame: Frameler çerçeve olarak da geçerler ve formların biraz daha profesyonel görülmelerini sağlarlar. Genellikle seçim kutuları ve onay kutuları ile birlikte kullanılırlar. Seçim kutuları kullanıcıya birkaç seçenek sunar. Kullanıcıda bu seçeneklerden sadece bir tanesini seçebilir. Onay kutuları da tıpkı seçim kutularına benzer fakat aynı anda birden fazla onay kutusu işaretlenebilir.



Listbox: Liste kutuları kullanıcıya bir liste sunmanın ideal bir yoludur. Kullanıcıya liste kusuntunda bulunan verileri inceleyebilir ve istediklerini seçebilirler. Liste kutularının bazı özellik ve metotları şunlardır.

ListCount: Listede bulunan eleman sayısını verir. Eğer ListCount 0 ise liste kutusu boştur. Listcount – 1 ifadesi ise listedeki son elemanı vermektedir.



Common Dialog (Diyalog Pencereleeri) : Microsoft Common Dialog ile standart diyalog pencereleri çağrılır. Aç, Kaydet, Renk, Help, Yazdır ve Font diyalog kutularının çağrılmasını sağlar.

CommonDialog1.Action = 1 Aç

CommonDialog1.Action = 2 Yeni Adla Kaydet

CommonDialog1.Action = 3 Renk

CommonDialog1.Action = 4 Font

CommonDialog1.Action = 5 Yazdır

CommonDialog1.Action = 6 Help

FileName: Dosya aç kutusunda açılması istenen dosya tipleri listelenir.

FileName = “*.txt”

Filter: Dosya aç kutusunda açılmasını istediğiniz dosyalar.

Filter= “Text dosyaları (*.txt)|*.txt|Bitmap(*.bmp)|*.bmp|Tüm

resimler(.jpg;*.wmf;*.ico)| *.jpg;*.wmf;*.ico|*

Tüm dosyalar(.*)|*.*|”*

6.2. Kullanılan Visual Basic Döngüleri ve Şartları

Döngü Deyimleri: Tekrarlanamsı istenen işlemler döngü içine alınarak istendiği kadar tekrarlanabilir.

For-Next Döngüsü: Tekrarlanacak ifadeler For ve Next komutları arasına yazılır.

Tekrar sayısı sayaçla kontrol edilir.

Aşağıdaki formatta yazılır:

For Sayaç=Başlangıç değeri To Bitiş değeri (Step) adım adım eklenebilir.

Komutlar.....

Next

Örnek: 1'den 10'ye kadar olan sayıları yazdıran program

Dim a

For a=1 To 10

Msgbox "Sayınız", a , "dır."

Next

If then else deyimi: Özel veya bir blok ifadeyi koşula bağlı olarak çalıştırmak için kullanılan karar yapısıdır. Her if endif komutu ile sonlandırmalıdır. Eğer tek bir satırda şart var ise endif yazılmaz.

If Şart Then

Komutlar.....

Else

Komutlar.....

End If

Örnek: Şifre programı

Dim sifre

sifre = Text1.Text

If sifre = "123" Then

Text2.Text = "Şifre doğru"

Else 'Aksi takdirde

Text2.Text = "Şifre yanlış"

End If

While-Wend Döngüleri: Döngünün kaç defa çalışacağı önceden bilinmiyorsa kullanılırlar. Belirtilen şart gerçekleşene kadar belirtilen komut blokları çalıştırılır [15].

While Şart

Komutlar.....

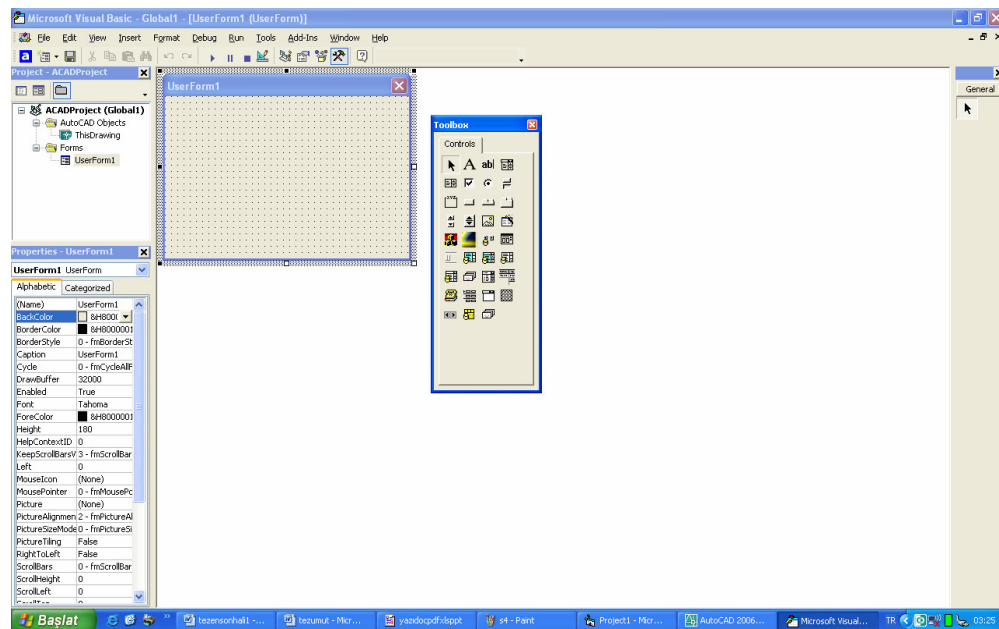
Wend

7. VISUAL BASIC APPLICATION (VBA) VE ACTIVEX AUTOMATION

7.1. Visual Basic Application (VBA)

VBA (Visual Basic For Application) Yıllar boyunca Microsoft uygulamaları ile kullanılacak programlama dili olmaktadır. AutoCAD gibi farklı programlara ulaşılarak kendi ortamında isteğe bağlı yeni eklemeler yapmak için kullanılır. Programa yeni menüler eklenerek programda olmaya özellikler eklenebilir [10].

İnsanların uygulamaları özgünleştirebilmeleri ve kendi ortamlarına uygun beceriler ekleyebilmeleri için, birkaç uygulama için ortak bir dil ve programlama ortamı oluşturmak için kullanılmaktadır. VBA'nın asıl gücü, onu destekleyen uygulamaların nesnelerinden kaynaklanır (Şekil 7.1).



Şekil 7.1. AutoCAD programında çalışan (VBA) ekranı

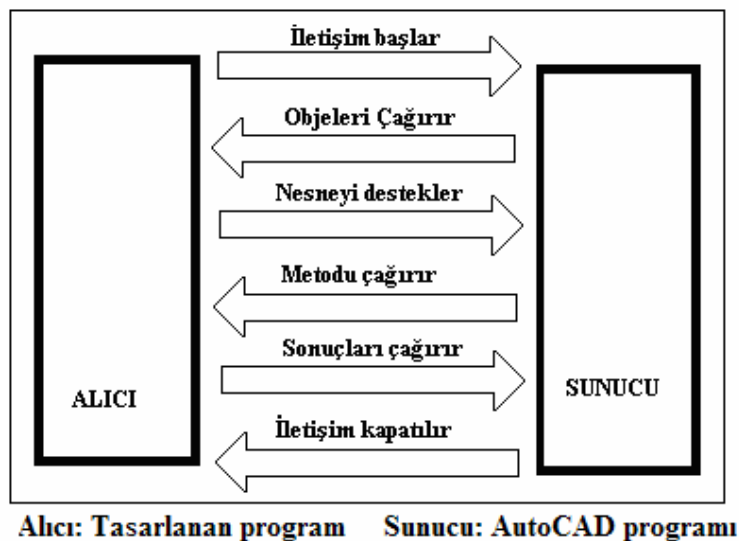
VB ve VBA birbirine benzediği için olaylar ve nesneler aynıdır. Özellikler menüsü, araçlar menüsü, editör kısmı VB ile aynıdır.

Farklı bağımsız çalışamamasıdır AutoCAD altında çalışır ve AutoCAD programı için kodları bulunmasıdır.

7.2. ActiveX Automation

AutoCAD menülerdeki araç çubuklarındaki ve Autolisp programlarındaki komutlarla kontrol edilir. AutoCAD'in kontrolü C++ programlama dili üzerinden iç hat kümesi tarafından sağlanır. Bu kümelere ObjectARX denilir ve C++ programlama dili üzerinden kullanılması için tasarlanmıştır. AutoCAD uygulama alanı diğer bütün bilgisayar uygulamaları gibi belirli bir alanı kapsamaktadır. AutoCAD'in mevcut özellikleri ile grafik programlarının bazı özelliklerinin bir araya gelmesi ile daha verimli çizim uygulamaları üretilebilir. AutoCAD sınırlı olduğu alanlardaki problemleri çözmek için ActiveX Automation'u ortaya çıkarmıştır [10].

ActiveX Automation iki uygulama arasında diyalog içersede bu eşler arasında iki yönlü değildir. Alıcı diyalogu başlatan bir uygulamadır sunucuda alıcıya cevap veren bir uygulamadır. ActiveX Automation kodu, sunucu tarafından kod kontrolleri uygulandığı zaman, alıcı içinde harekete geçer (Şekil 7.2).



Şekil 7.2. Alıcı-sunucu iletişimi

Şekilde görüldüğü gibi her ActiveX automation operasyonunun içinde çeşitli adımlar vardır. Önce ActiveX automation operasyonunun başlamasına karar verilir. Bu uygulama otomatikman alıcı olur ve uygulama çağrılır. Server hangi komutların meydana çıkacağına karar verir. Alıcı uygulaması komutları istediği zamanda uygular. Komutlara uygun şekilde cevap veren sunucuya ActiveX Automation tarafından komutlar geçilir. Alıcı sunucuya ne kadar komut göndermek isterse gönderir.

7.2.1. Otomasyon nesneleri

Nesneler sunucu (server) uygulamasının bir bölümünün temsilidir. Nesne, bağımsız bir uygulama, bir uygulama tarafından yönetilen dokümanın bir kısmı veya araç çubuğu bir uygulamanın arabirimi olabilir. Nesneleri ayıran özellikler aşağıdaki gibidir [10].

1. Nesnenin tipi ve sınıfı
2. Nesnenin özellikleri
3. Nesnenin metodları

Bunlar nesneyi tarif eden özellikleridir. Örneğin; AutoCAD tarafından ihtiyacı karşılanan nesnelerden bir tanesi, AutoCAD çizimlerinin herhangi bir yerindeki bir çizgiyi göstermektedir. AutoCAD içerisindeyken çizgi çizilebilir. Çizgi nesnesinin bazı özellikleri;

1. Renk
2. Katman
3. Başlangıç noktası
4. Bitiş noktası
5. Kalınlık

Doğruların metodları, AutoCAD kullanıcı arabirimlerinde yapılabilecek işleri içermektedir. Doğru nesnesinin bazı metodları;

1. Kopyalama
2. Silme
3. Aynalama
4. Döndürme
5. Taşıma

7.2.2. Otomasyon ve AutoCAD

Uygulamalar, ActiveX Automation yoluyla yeteneklerini ortaya çıkarmak için boyut içinde, geniş olarak değişmektedirler. Microsoft Excel gibi uygulamalar, alıcı uygulamalara hem kendi arabirimlerini hemde verilerini kullanmaları için izin verir. Visual basic uygulaması, kendi arabiriminden bazı parametreleri, ActiveX Automation alıcılarının etkisine bırakır. Çizimlerini görünümünü kontrol eden faktörler ile çizim alanının büyüklüğü ve yerleşim gibi konularda ActiveX Automation üzerinden çizim içinde kullanılan bütün nesneler elde edilebilir. ActiveX Automation ile AutoCAD 2000 ve sonrası sürümleri için menülerin ve araç çubuklarının daha iyi kullanılmasını sağlayan özellikler eklenmiştir [10].

7.2.3. AutoCAD’de nesne modeli ve hiyerarşi

Nesne AutoCAD ActiveX çalışma noktasının ana yapısıdır. AutoCAD ‘in ActiveX ortak ara yüzünde çok değişik türde nesneler bulunur.

Bu nesnelerden bazıları:

1. Doğru, yay, yazı ve ölçü gibi grafik nesneleri,
2. Doğru tipi, sili ve ayarları,
3. Katman ve blok,
4. Görünüş.

Visual Basic Application’un (VBA) içerisinde AutoCAD’e geçmek kolaydır. VBA açılmış AutoCAD altından çalışır. Nesneye ya doğrudan ya da kullanıcı tanımlama

değişkeniyle ulaşılabilir. Nesnelere doğrudan ulaşmak için nesne çağrılan hiyerarşiye eklenir.

Örneğin aşağıdaki ifade model boşluğuna bir doğru ekler.

Dim baslama_noktasi (0 to 2) as double

Dim bitis_noktasi (0 to 2) as double

Dim lineobj as AcadLine

baslama_noktasi (0) = 0 (X eksenini ifade eder.)

baslama_noktasi (1) = 0 (Y eksenini ifade eder.)

baslama_noktasi (2) = 0 (Z eksenini ifade eder.)

bitis_noktasi (0) = 30 (X eksenini ifade eder.)

bitis_noktasi (1) = 20 (Y eksenini ifade eder.)

bitis_noktasi (2) = 0 (Z eksenini ifade eder.)

Set lineobj=ThisDrawing. ModelSpace. AddLine (baslama_noktasi, bitis_noktasi)

Set deyimi lineobj komutuna başlama ve bitiş noktaları arasında model ekranına çizim yaptırır.

7.2.4. Visual Basic programından AutoCAD'e bağlanma

Visual Basic'den (VB) AutoCAD'i çalıştırmak için aşağıdaki değişkenleri tanımlamak gerekir. Bu değişkenler sadece AutoCAD programını açar [10].

Çizim için ilgili komutun ve koordinatların girilmesi gerekmektedir.

Dim acadapp As AcadApplication

Set acadapp = New AcadApplication

Dim acaddoc As AcadDocument

```

Set acaddoc = acadapp. Documents. Add
acaddoc. Activate
acadapp. Visible = True
Örnek

```

VB'den başlama ve bitiş noktaları arasına doğru çizdirmek için bir Command butonun altına aşağıdaki kodlar yazılır.

Bu butona tıklanınca AutoCAD çalışıp çizim yapılacaktır.

```

Private Sub Command1_Click()

On Error Resume Next
Dim acadapp As AcadApplication
Set acadapp = New AcadApplication
Dim acaddoc As AcadDocument
Set acaddoc = acadapp. Documents. Add
acaddoc. Activate
acadapp. Visible = True

Dim lineobj As AcadLine
Dim sp(0 To 2) As Double
Dim ep(0 To 2) As Double

sp(0) = 0
sp(1) = 100
sp(2) = 0

ep(0) = 200
ep(1) = 200
ep(2) = 0

```

Set lineobj = acaddoc. ModelSpace. AddLine(sp, ep)

End Sub

7.2.5. ActiveX Automation AutoCAD komutları

ActiveX komutlarından bazıları aşağıda verilmiştir [10].

Line: İki nokta arasında doğru çizmeyi sağlar. Geçerli olan sürekli doğru tipidir. Doğru komutu aşağıdaki Çizelge 7.1’ de verilen tiplerden birisi ile kullanılabilir.

Çizelge 7.1. Doğru çeşitleri

Adline	İki nokta arasında doğru çizer.
AddlightweightPolyline	Köşe noktalarından geçen iki boyutlu polyline çizer.
AddMline	Çoklu doğrular çizer.
AddPolyline	2 veya 3 boyutlu polyline çizer.

Doğru çizmek için Başlama noktası ve Bitiş noktası gereklidir.

Çember, yay, spline vb. eğrileri çizmek için Çizelge 7.2’ deki komutlar kullanılır.

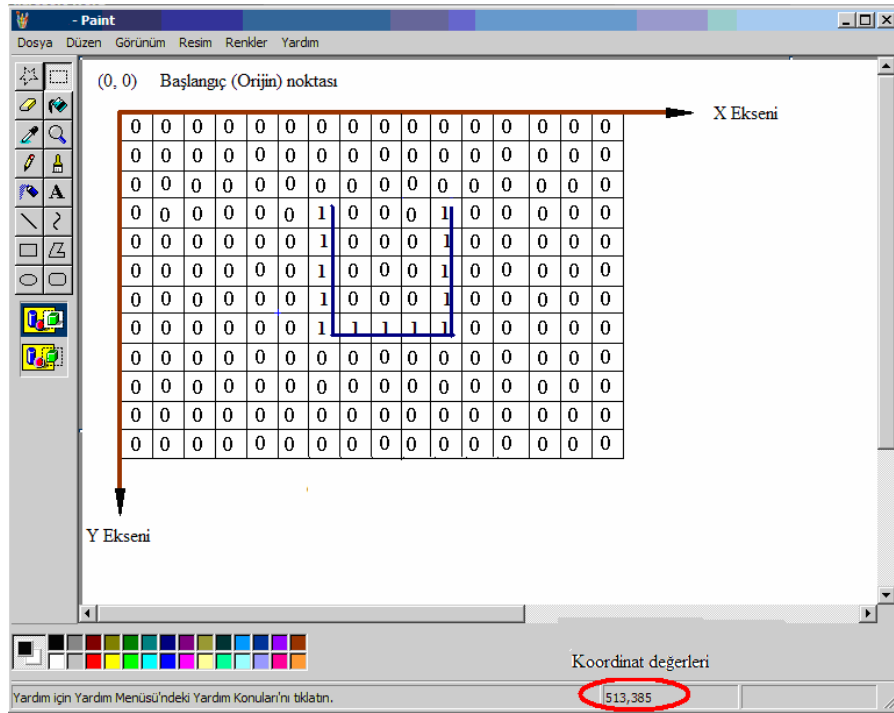
Çizelge 7.2. Eğri çeşitleri

AddArc	Merkezi, yarıçapı ve başlangıç ve bitiş açıları verilen bir yay meydana getirir.
AddCircle	Merkezi ve yarıçapı verilen bir çember çizer.
AddEllipse	Merkezi, birinci eksenin başlangıç noktası ve yarıçapı oranı verilen bir elips çizer.
AddSpline	Verilen kontrol noktalarına göre spline eğrisi çizer.

8. GELİŞTİRİLEN PROGRAM

8.1. Visual Basic ve ActiveX Komutlarını Kullanarak (Bmp) Dosya Okuma

Yorumlanacak Şekil önce paint programında açılır. Şekil okuma 1 ve 0 sistemine göre çalışmaktadır. 1'ler dolu 0'lar boş noktayı temsil etmektedir. Bir piksel, 0 ya da 1'den sadece birinin değeri alabildiği rakamdır. Resimler 24 bit derinliğinde okunmaktadır (Şekil 8.1).



Şekil 8.1. Paint programı arayüzü

Taranan resim 800x600 piksel ölçülerinde ekrana alınmaktadır. Paint programı (0, 0) koordinatlarından başladığı için en büyük piksel değeri 799x599 olarak kabul edilir.

Ekrana alınan resmi yorumlamak için tüm piksellerin taranması gerekmektedir. Bu işlem için For-Next döngüsü kullanılmaktadır.

Resim yatayda 800'e kadar dikeyde 600'e kadar piksel piksel taranarak yorumlanır.

Public xxx ' Dikey tarama

Public yyy ' Yatay tarama

Taranan resimin boyutları otomatik olarak alınmaktadır. Formülleri aşağıdaki gibidir.

s = "Genişlik=" & BT. bmWidth & vbCrLf

xxx = BT. bmWidth - 1 'Genişlik otomatik olarak atanır

s = s & "Yükseklik=" & BT. bmHeight & vbCrLf

yyy = BT. bmHeight - 1 'Yükseklik otomatik olarak atanır

Taranacak resmin ölçüleri 800x600 piksel ölçülerinden büyükse resim işlenmeyecektir. Aşağıdaki program satırları ile boyut kontrolü yapılır.

xxx = xxx - 1

yyy = yyy - 1

If xxx > 599 Or yyy > 799 Then MsgBox "Resim ölçüleri (800x600) pikselden büyük", , "Piksel Büyüklüğü"

If xxx > 599 Or yyy > 799 Then Exit Sub

Visual basic programının 800x600 piksel ölçülerinde verimi oldukça düşmektedir. Tüm piksellerin yorumlanması çok zaman almaktadır. Bu yüzden örneklemeler 200x200 piksel üzerinden yapılmıştır.

Taranacak resimler 800x600 piksel arasında olacağından;

For X = 0 To xxx

For Y = 0 To yyy

Şeklinde döngüler tanımlandı. Döngüye göre X yatayda xxx kadar, Y dikeyde yyy kadar döngü içinde kalacaktır. Yalnız sınır 799x599 olacaktır. Çünkü tarama (0, 0)

koordinatlarından başlamaktadır. Resim kutusundaki resim (PictureBox) taranarak dolu pikseller 1 ile boş pikseller 0 ile tanımlanır.

Picture1.BackColor = Picture1.Point(X, Y)

back = Picture1.BackColor

Sadece çizim için kullanılacak olan dolu piksellere koordinat değeri atanır. Bulunan pikseller X koordinatları olarak list1 ve list3, Y koordinatları ise list2 ve list4 listeleme kutularına yazılır.

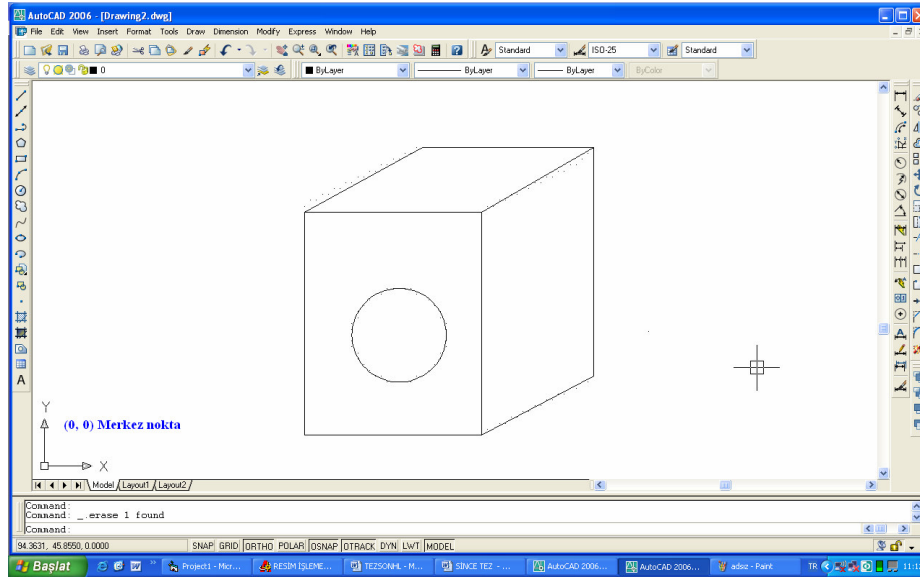
Bmp resimlerin koordinat değerleri ve yönleri AutoCAD programı gibi değildir. AutoCAD programında koordinat başlangıç noktası (orijin) sol alt nokta olduğu halde Bmp resmin paint programında sol üst noktadır (Şekil8.2).

Paint programı piksel olarak çalışmakta AutoCAD programı mm veya inch ölçüsü üzerinde çizim yapmaktadır. Piksel ile mm arasındaki dönüşümler için matematiksel katsayılar kullanılmıştır.

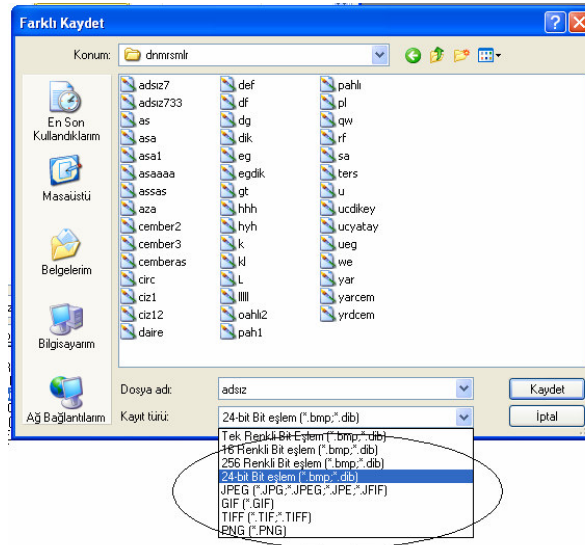
Bitmap dosyasının koordinat merkezi sol üst köşe, AutoCAD programının ise sol alt köşedir. Bitmap dosyasında Y eksenini aşağı doğru AutoCAD'de ise yukarı doğrudur. Taranan bitmap resim paint programına göre olduğu için resim ters çıkmaktadır. Aynalama kodları yazılarak Y ekseninde yeniden konumlama yapılmıştır.

Paint programında koordinat değerleri sol üstten başlar. AutoCAD programına göre Y eksenini ters yöndedir [16].

Çok teferruatlı bir resim işleme program olmamasına rağmen paint programında şekil düzeltmeleri veya değişik formattaki resimleri birbirine dönüştürme işlemleri yapılabilmektedir.



Şekil 8.2. AutoCAD programı arayüzü



Şekil 8.3. Paint programında resim formatlarının kaydedilmesi

Bmp olarak ortama alınan bir resim, jpeg, gif, tif veya png formatında kaydedilebilir.

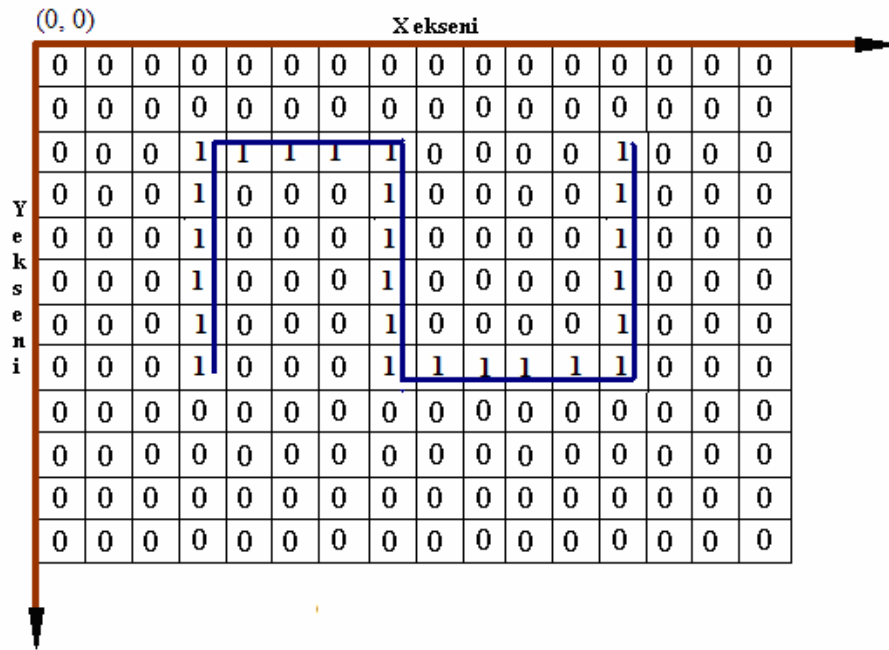
Geliştirilen program sadece bmp resim formatı işlediği için jpeg, gif, tiff ve png olarak gelen resimleri paint programında bmp formatına dönüştürmek gerekmektedir (Şekil8.3).

Dwg format yapısı ikili (binary) sayı sisteminde ve karmaşık olduğu için Visual Basic programında ActiveX Automation komutları kullanılmıştır. Resim yorumlandıktan sonra AutoCAD komutlarına ulaşılarak nesneler AutoCAD ekranına çizdirilmektedir.

8.2. Piksel Yorumlamaları

Resim veri tabanında 0 ve 1 olarak piksel bilgileri bulunur. Normal halde 0 siyah 1 beyaz noktadır. Programda kolay kullanım için 1 siyah 0 beyaz olarak seçilmiştir (Şekil 8.4).

Şekil 8.4’ deki gibi ekrandaki piksel boş ise değeri 0, dolu ise değeri 1 olacaktır.



Şekil 8.4. “0 ve 1” pikseller

Programda açılan resim piksel piksel taranarak dolu olan pikseller belirlenmekte ve koodinatlari hafızaya alınmaktadır.

Picture1.BackColor = Picture1.Point(X, Y)

back = Picture1.BackColor

If back = 0 Then

c = 1

Else

c = 0

End If

Resim yorumlandıktan sonra 1'lerin oluşturduğu başlangıç ve bitiş değerleri listeleme kutularına (listbox) atılır (Şekil 8.5).

b = d & " " & "X=" & X & "," & "Y=" & Y

Form1.List1.AddItem X1

Form1.List2.AddItem Y1

Form1.List3.AddItem X2

Form1.list4.AddItem Y2

KOORDİNATLAR Piksel koordinat noktaları

SIRA	X1	Y1	X2	Y2
1	8,12	7,773851	26,85512	7,773851
2	14,1	11,66077	21,90812	11,66077
3	12,0	17,66784	14,13427	17,66784
4	21,9	17,66784	23,32155	17,66784
5	4,59	28,62190	12,01413	28,62190
6	23,3	28,62190	30,03533	28,62190
7	30,0	10,95406	30,03533	28,62190
8	4,59	11,30742	4,593639	28,62190
9	14,1	11,66077	14,13427	17,66784
10	21,9	11,66077	21,90812	17,66784
11	12,0	17,66784	12,01413	28,62190
12	8,12	17,66784	23,32155	28,62190
13	23,3	7,773851	30,03533	10,95406
14	30,0	7,773851	4,593639	11,30742

Listeleme kutuları

- 1(x1,y1)23,22(x2,y2)76,22
- 2(x1,y1)40,33(x2,y2)62,33
- 3(x1,y1)34,50(x2,y2)40,50
- 4(x1,y1)62,50(x2,y2)66,50
- 5(x1,y1)13,81(x2,y2)34,81
- 6(x1,y1)66,81(x2,y2)85,81
- 7(x1,y1)85,31(x2,y2)85,81

Şekil 8.5. Listeleme kutuları ve koordinat değerleri

1 piksel yaklaşık 0.353 mm' dir (Çizelge 8.1).

Çizelge 8.1. Piksel ölçü değerleri

PİKSEL VE DİĞER BOYUT ÖLÇÜLERİ	
10 Piksel	3,53 mm
10 Piksel	0,139 inch
10 Piksel	0,35 cm
10 Piksel	10 points
1 Piksel	0,353 mm

Çizimlerin ölçülü çıkması için;

1 mm için piksel katsayısı,

$P=2,83286118980169971671388101983$ 'dır.

Yaklaşık 2.833 piksel olarak kullanılır. Buna göre çıkan koordinat değerlerini 2.833 katsayısına bölmek gerekmektedir.

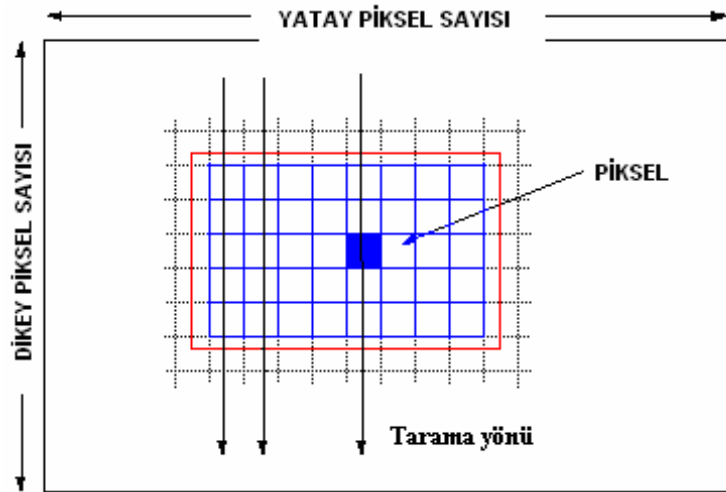
$X1 / (2.83), Y1 / (2.83), X2 / (2.83), Y2 / (2.83)$

Daha sonra listeleme kutularına atanan koordinatlar Active X komutları ile AutoCAD ortamına alınmaktadır. Listelerdeki koordinatlar “listeyi sil” düğmesi ile her defasında silinmektedir. X değeri bir listeleme kutusuna Y değeri de diğer listeleme kutusuna atanır.

Listeleme kutularındaki koordinat değerleri listeleme kutu isimleri ile programda işlenmektedir. Bmp resim tarama yönü olarak Y eksen (sütun) seçilmiştir. Yani tarama dikey olur. Tarama bir sütun bitince diğer sütunla devam eder. Bu işlem resim boyutu kadar piksel piksel devam eder.

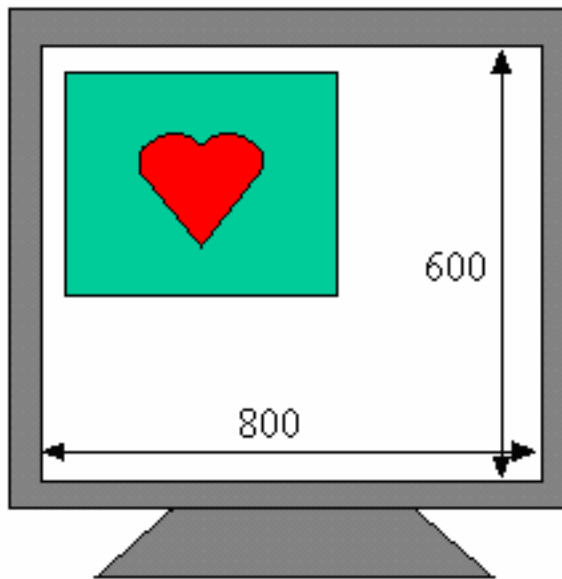
Resim sadece siyah ve beyaz olarak okutulmaktadır. Diğer renkler göz ardı edilmiştir. 1'ler dolu nokta, 0'lar ise beyaz nokta olarak okutulmaktadır (Şekil 8.6).

Tarama sonucunda elde edilen koordinat değerlerine göre nesneler yorumlanarak tipi (nokta, doğru, çember, vb.) belirlenir.



Şekil 8.6. Düşey eksende tarama

800x600 piksel çözünürlükteki resimler program içerisindeki picturebox (Resim kutusu) içine alındıktan sonra yorumlama yapılmaktadır (Şekil 8.7).



Şekil 8.7. 800x600 ekran çözünürlüğü

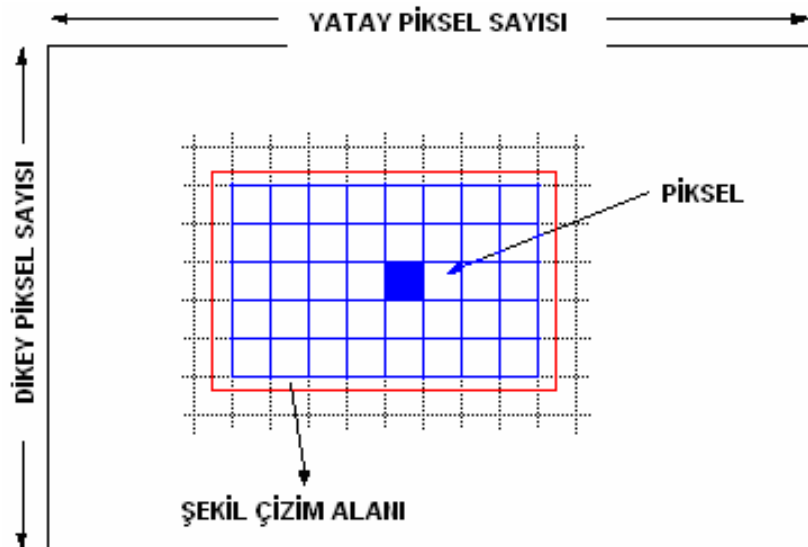
Visual basic (VB) programı picturebox elemanı ile bitmap, icon, metafile, jpeg ve gif gibi dosyaları resimlemek için kullanılmaktadır. Ayrıca çeşitli metotlar kullanılarak picturebox içerisinde çizimler yapılabilmektedir.

Aşağıdaki program satırı ile AutoCAD programında yeni dosya açılmaktadır.

```
Dim acadapp As AcadApplication 'AutoCAD programı
Set acadapp = New AcadApplication 'Yeni sayfa
Dim acaddoc As AcadDocument 'Çizim sayfası ekle
Set acaddoc = acadapp.Documents.Add 'AutoCAD aktive et
acaddoc.Activate 'Sayfayı aktive et
acadapp.Visible = True 'Görünürlük açık
```

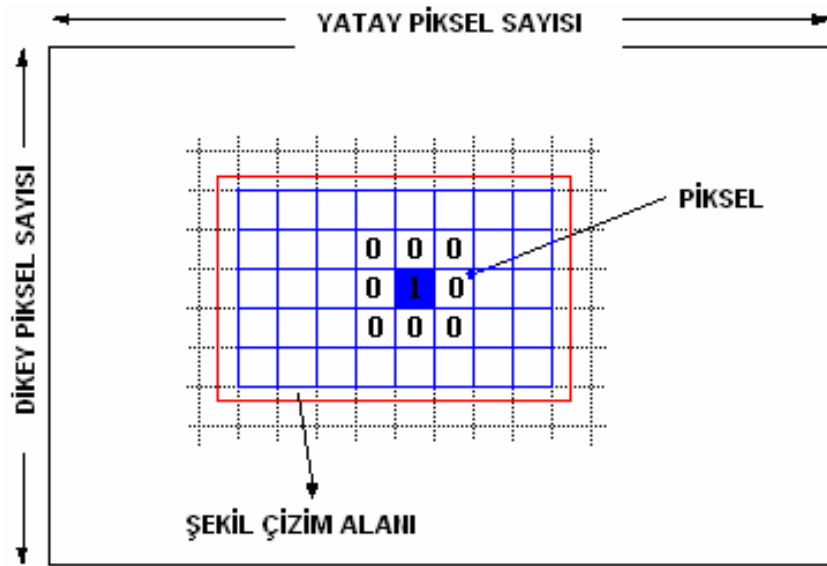
8.2.1. Noktanın yorumlanması

Bmp Şekilpiksel piksel (0 ve 1 sistemine göre) taranır. Programda tarama yapılırken bulunan tek noktanın altı, üstü, sağı ve solu kontrol edilir. Herhangi bir nesne yok ise bu nesne nokta olarak tespit edilerek koordinatı saklanır (Şekil 8.8).



Şekil 8.8. Noktanın yorumlanması

Programda taranan noktanın koordinatları list8.list' e (X), list9.list'e (Y) olarak atanır. Buradaki rakamsal koordinatlar AutoCAD ortamında ActiveX komutu olan “addpoint” komutu kullanılarak koordinata nokta konulur. 1 olarak tespit elden pikselin çevresindeki piksellerin değeri 0 ise bu nesne nokta olarak kabul edilir (Şekil 8.9).



Şekil 8.9. Nokta taranması

Değeri 1 olan pikselin X ve Y ekseninde çevresindeki diğer piksellerin değerine bakılır.

$(X(i-1), (Y(i-1)), (x(i), Y(i-1)), (X(i+1), Y(i+1)),$
 $(X(i-1), Y(i), (X(i+1), Y(i), (X(i-1), Y(i+1)),$
 $(X(i), Y(i+1)), (X(i+1), Y(i+1))$

Nokta piksel değerleri sıfır ise bu nesne nokta olarak belirlenir. Bulunan pikselin X değeri list2.list ve Y değeri ise list3.list listeleme kutularında belirir (Şekil 8.10).

Bu değerlere AddPoint komutu kullanılarak nokta atanır.

KOORDİNATLAR				
SIRA	X1	Y1	X2	Y2
1	8,12	7,773851	26,85512	7,773851
2	14,1	11,66077	21,90812	11,66077
3	12,0	17,66784	14,13427	17,66784
4	21,9	17,66784	23,32155	17,66784
5	4,59	28,62190	12,01413	28,62190
6	23,3	28,62190	30,03533	28,62190
7	30,0	10,95406	30,03533	28,62190
8	4,59	11,30742	4,593639	28,62190
9	14,1	11,66077	14,13427	17,66784
10	21,9	11,66077	21,90812	17,66784
11	12,0	17,66784	12,01413	28,62190
12	23,3	17,66784	23,32155	28,62190
13	30,0	7,773851	30,03533	10,95406
14	4,59	7,773851	4,593639	11,30742

1(x1,y1)23,22(x2,y2)76,22
2(x1,y1)40,33(x2,y2)62,33
3(x1,y1)34,50(x2,y2)40,50
4(x1,y1)62,50(x2,y2)66,50
5(x1,y1)13,81(x2,y2)34,81
6(x1,y1)66,81(x2,y2)85,81
7(x1,y1)85,31(x2,y2)85,81

Şekil 8.10. Nokta koordinatları

AcadPoint AutoCAD değişkeni olarak atanır.

Dim nokta_nesnesi As AcadPoint

Koordinatlar olarak bolge değişkeni atanır.

Dim bolge (0 To 2) As Double

Koordinat değerleri (x, y, z) olduğu için (0 to 2) olarak gösterilir. Bunun anlamı üç adet koordinat değeri vardır.

(0 To 2)

Burada;

0: X koordinatını,

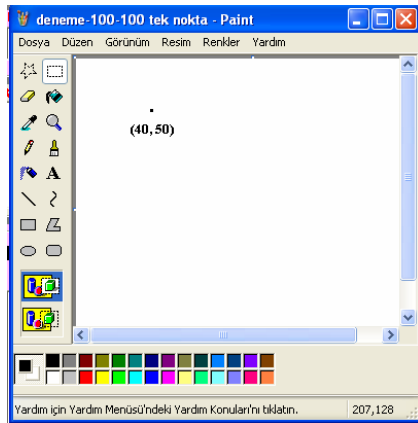
1: Y koordinatını

2: Z koordinatını ifade eder. Z koordinatı kullanılmayacağı için değeri “0” atanır.

Kullanılan değişkeni atamak için Set komutu kullanılmalıdır.

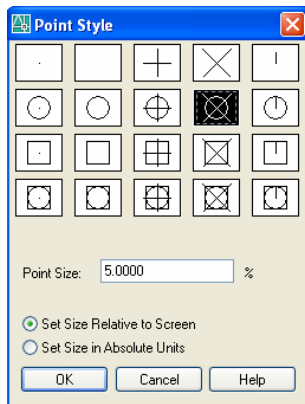
Set nokta_nesnesi = acaddoc. modelSpace. AddPoint(bolge)

Tanımlanan noktanın paint programındaki görünümü Şekil 8.11’ de verilmiştir.

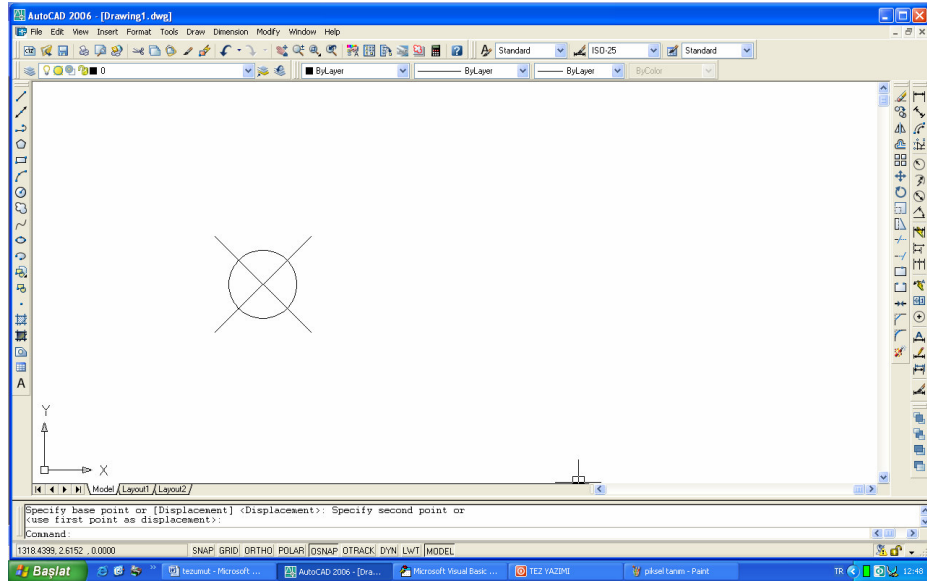


Şekil 8.11. Noktanın paint programında görünümü

AutoCAD ortamında açılan sayfada nokta nesnesinin gözükmesi için nokta nesnesi modu değiştirilebilir. Ölçüsü “ %” olarak girilebilir (Şekil 8.12). Burada nokta nesnesi kullanılan çizim (drawing) sayfasında çalışmaktadır. İlgili koordinatlara Addpoint komutu ile nokta atanır. AutoCAD çalışır ve ekranda nokta belirmektedir (Şekil 8.13).



Şekil 8.12. AutoCAD nokta stil diyalog kutusu



Şekil 8.13. AutoCAD nokta uygulaması

AutoCAD ortamında noktanın Şekil 8.13'deki gibi görüntülenmesi için aşağıdaki program kullanılır.

Sub nokta_olustur()

Dim nokta_nesnesi As AcadPoint 'Nesne nokta nesnesidir.

Dim bolge(0 To 2) As Double 'X, Y ve Z koordinatlarını ifade eder.

bolge(0) = 15 burada koordinat elle girilmiştir.

bolge(1) = 45 burada koordinat elle girilmiştir..

bolge(2) = 0

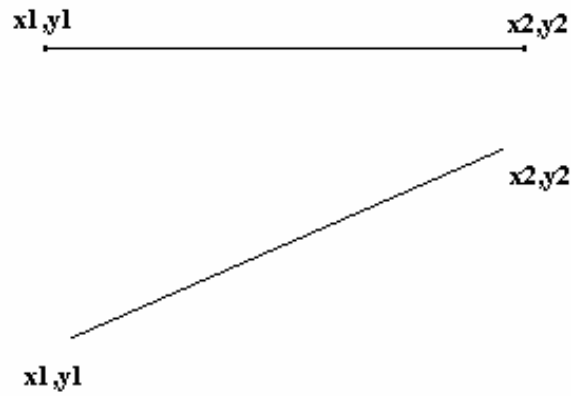
Set nokta_nesnesi = acadoc. ModelSpace. AddPoint(bolge) 'bolge değişkenine Set komutu ile nokta atması yapılır.

ZoomAll

End Sub

8.2.2. Doğru yorumlanması

Resimdeki pikseller yorumlanarak yatay, dikey ve açılı doğrular belirlenebilmektedir. Ekranda doğrular varsa bunların başlangıç ve bitiş nokta koordinatlarının tespit edilmesi gerekmektedir.



Şekil 8.14. Doğru parçaları

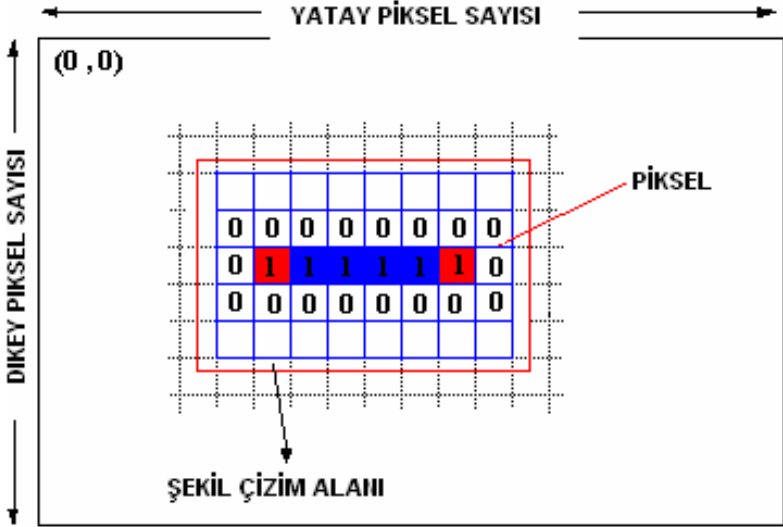
Yatay Doğru: Tarama yapılırken 1 değeri olan ilk pikselden sonra bitişik $X(i+1)$ pikselleri 1 değerinde ise ve Y koordinatında bir değişme yoksa bu nesne yatay doğru olarak kabul edilir (Şekil 8.15).

Pikselin yanındaki pikselin değeri 1 ise While-Wend döngüsü kullanılarak değeri 0 olan piksele ulaşıncaya kadar taramaya devam edilir.

Değeri 0 olan piksel bulununca X değerleri list1, list3 ve Y değerleri list2, list4 listeleme kutularına listelenir (Şekil 8.16).

Yatay doğru nesnesinin solunda, sağında, üstünde ve altındaki piksellerin değeri 0 olmalıdır.

Piksel değerleri X ekseninde oluşmaktadır.



Şekil 8.15. Yatay doğru taraması

SIRA	X1	Y1	X2	Y2
1	2,47345	19,43462	32,15547	19,43462

1(x1,y1)7,55(x2,y2)91,55

Şekil 8.16. Yatay doğru için başlangıç ve bitiş doğruları koordinatları

List3.list listeleme kutusundaki Y koordinat değerleri sabit olur (Şekil 8.16).

$Y(i)=Y(i+1)$ karşılığı

$List2.List(i) = List4.List(i + 1)$ olacaktır.

List2.list listeleme kutusundaki X koordinatlarında artış olacaktır.

$$X(i)=a, X(i+1)=a+1, X(i+2)=a+2 \dots X(i+n)=a+n$$
$$List1.List(i) = X(i), List3.List(i+1)=X(i+1)....Listn.list(i+n)=X(i+n)$$

Şart komut satırı da aşağıdaki gibi olacaktır.

If List2.List(i) = List4.List(i + 1) And List2.List(i) <= List3.List(i + 1) Then

X koordinatındaki artış, Y koordinatındaki sabitlik şartı sağlanınca *Yatay doğru* tespiti yapılmış olacaktır. Doğru çizmek için noktalara değişken ataması yapılır. Başlama ve bitiş noktası olarak PT1 ve PT2 değişkenlerin ataması yapılır. Başlangıç noktası (0 to 2) arasında tanımlanmaktadır.

Dim PT1(0 To 2) As Double

0:X koordinatı

1:Y koordinatı

2:Z koordinatı

Bitiş noktası (0 to 2) arasında tanımlandı.

Dim PT2(0 To 2) As Double

0:X koordinatı

1:Y koordinatı

2:Z koordinatı

Başlangıç noktası olarak List1 ve list2 'nin ve bitiş noktası List3 ve list4'ün ilk değerleri alınır.

PT1(0) = List1.List(i) 'Başlangıç X koordinatı

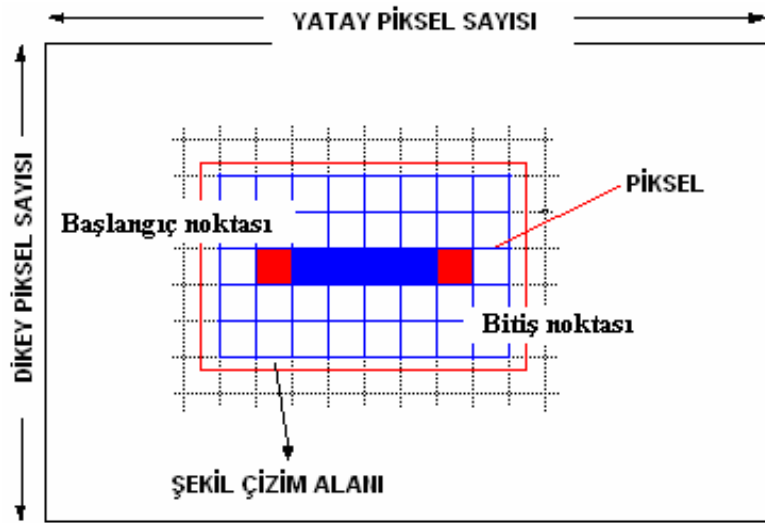
PT1(1) = List2.List(i+n) 'Başlangıç Y koordinatı

PT1(2) = 0 ' Z koordinat değerleri "0" olacaktır.

PT2(0) = List3.List(i) 'Bitiş X koordinatı

PT2(1) = List4.List(i + n) 'Bitiş X koordinatı

PT1(2) = 0 'Z koordinat değerleri "0" olacaktır.



Şekil 8.17. Yatay doğru uygulaması

Bulunan ilk piksel koordinatı ile son piksel koordinatı arasına AutoCAD ortamında “addline” komutu ile doğru çizilir (Şekil 8.17).

Tarama sonucu tespit edilen doğrunun başlangıç ve bitiş noktasının koordinatları kaydedilir.

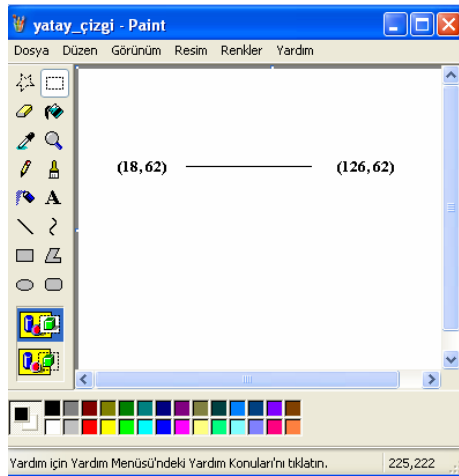
Değişken olarak nokta AutoCAD Visual Basic Application (VBA) komutu olarak atanır.

Dim objline As AcadLine

Listeleme kutularındaki koordinatlar arasına çizgi çizmek için AcadLine değişkeninin tanımlanması gerekir. Kullanılan değişkeni atamak için Set komutu kullanılmalıdır. PT1 ve PT2 noktaları arasına Set komutu kullanarak doğru çizimi yapılır.

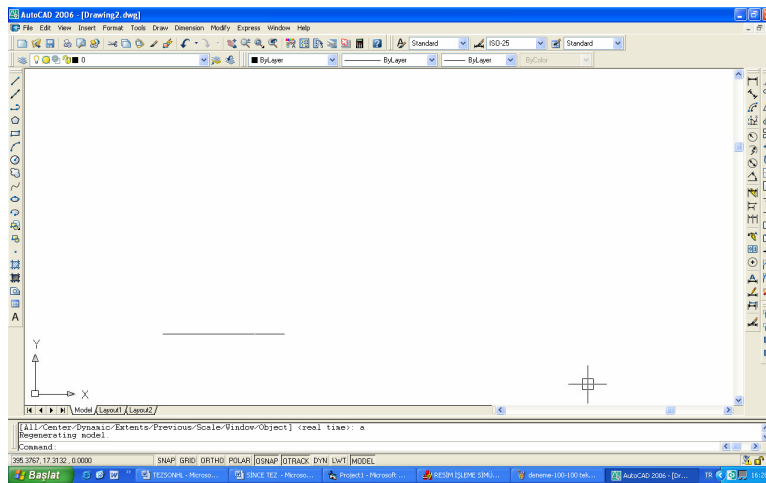
Set objline=acaddoc. ModelSpace. AddLine(PT1,PT2)

Yatay doğrunun paint programında görünümü (Şekil 8.18)’de ki gibidir.



Şekil 8.18. Yatay doğrunun paint programında gösterilmesi

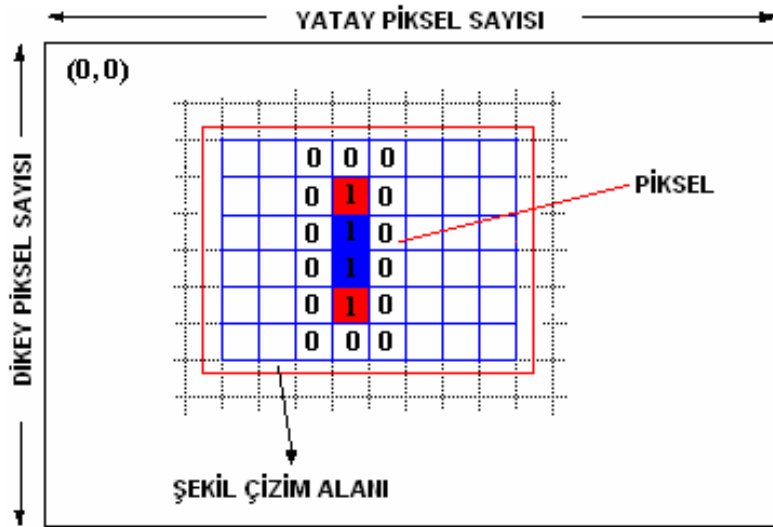
İlgili koordinatlar arası AddLine komutu ile birleştirilir. AutoCAD çalışır ve ekranda yatay çizgi belirir (Şekil 8.19).



Şekil 8.19. AutoCAD programında yatay doğru uygulaması

Dikey doğru: Ekran taraması yapılırken değeri 1 olan ilk pikselin X eksenindeki yanında bulunan bir sonraki pikselin değeri sıfırsa ve birbirine komşu aynı koordinattaki artarak giden diğer Y eksenindeki piksellerin değeri 1 ise bu nesne dikey doğru olarak tanımlanır (Şekil 8.20). Bulunan ilk noktanın yanına bakılır başka bir nokta varsa while-wend döngüsü kullanılarak sütun taranır yatayda değişme

yoktur. Tarama sonrasında Y değerleri list2, list4 ve X değerleri list1, list3 listeleme kutularına listelenir (Şekil 8.21).



Şekil 8.20. Dikey doğru taraması

KOORDİNATLAR				
SIRA	X1	Y1	X2	Y2
1	8,4805E	5,65371E	8,48056E	27,9151E
1(x1,y1)24,16(x2,y2)24,79				

Şekil 8.21. Dikey doğru için başlangıç ve bitiş doğruları koordinatları

Liste2.list listeleme kutusunda X değerlerinde değişiklik olmayacaktır (Şekil 8.21)

$$X(i)=X(i+1) , X(i)=X(i+2),...X(i)=X(i+n)$$

$List1.List(i) = List3.List(i + 1), List1.list(i)=List3.list(i+2),...List3.list.list(i+n)$ olacaktır.

List3.list listeleme kutusundaki Y koordinatlarında artış olacaktır (Şekil 8.21).

$$Y(i)=b, Y(i+1)=b+1, Y(i+2)=b+2, \dots, Y(i+n)=b+n$$

Şart satırı aşağıdaki gibi olacaktır.

If List1.List(i) = List3.List(i + 1) And List2.List(i) <= List4.List(i + 1) Then

Y koordinatındaki artış, X koordinatındaki sabitlik şartı sağlanınca *Dikey doğru* çizmek için noktalara değişken ataması yapılır. Başlama ve bitiş noktası olarak PT değişkeni olarak ataması yapıldı. Başlangıç noktası (0 to 2) arasında tanımlanır.

Dim PT1(0 To 2) As Double

0:X koordinatı

1:Y koordinatı

2:Z koordinatı

Bitiş noktası (0 to 2) arasında tanımlandı.

Dim PT2(0 To 2) As Double

Başlangıç ve bitiş nokta koordinatlarına değer ataması listeleme kutularından yapılır. Başlangıç noktalarına (PT1), list1 ve list2 listeleme kutusuna ataması yapılır. Bitiş noktalarına (PT2), list2 ve list4 listeleme kutusuna ataması yapılır.

PT1(0) = List1.List(i) 'Başlama noktası X koordinatı

PT1(1) = List2.List(i+ n) 'Başlama noktası Y koordinatı

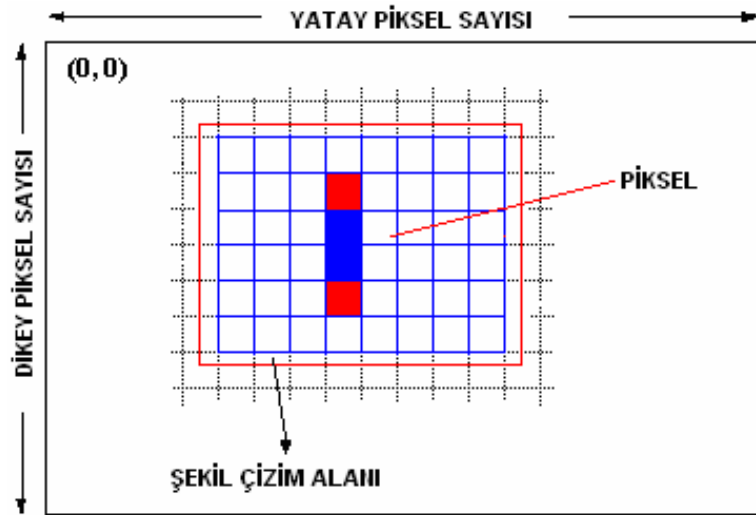
PT1(2) = 0 ' Z koordinat değeri "0" olacaktır.

PT2(0) = List3.List(i) 'Bitiş noktası x koordinatı

PT2(1) = List4.List(i + n) 'Bitiş noktası Y koordinatı

PT1(2) = 0 ' Z koordinat değeri "0" olacaktır.

Y ekseninde piksel değeri 0 olan noktaya ulaşılmıca işlem durdurulur. Doğrunun bitiş noktası belirlenir (Şekil 8.22).



Şekil 8.22. Dikey doğru uygulaması

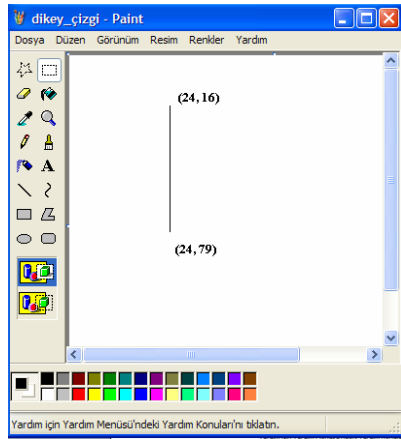
Tarama sonucu belirlenen doğrunun başlangıç ve bitiş nokta koordinatları saklanır. Koordinatlar AutoCAD ortamında “addline” komutu kullanılarak çizgiye dönüştürülür. Başlama ve bitiş noktaları arasında “AddLine” komutu ile dikey doğru çekilir. Değişken olarak nokta AutoCAD Visual Basic Application komutu olarak atanır.

Dim objline As AcadLine

Listeleme kutularındaki koordinatlar arasında çizgi çizmek için AcadLine değişkeninin tanımlanması gerekir. Kullanılan değişkeni atamak için Set komutu kullanılmalıdır. PT1 ve PT2 noktaları arasında Set komutu kullanarak çizgi çizimi yapılır.

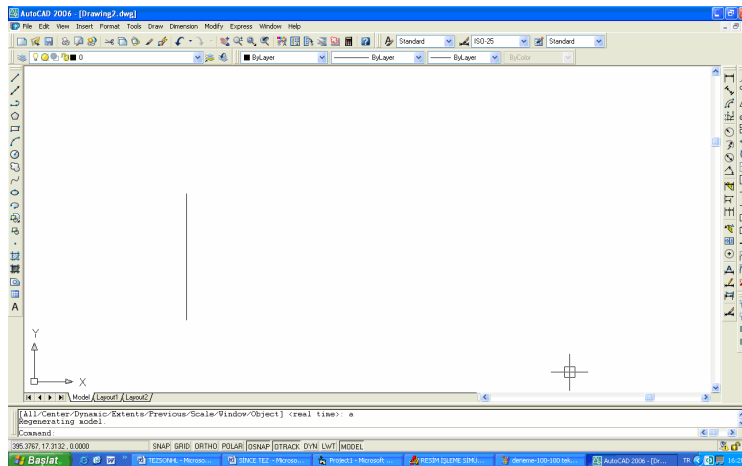
Set objline=acaddoc. ModelSpace. AddLine(PT1,PT2)

Yatay doğrunun paint programında görünümü (Şekil 8.23)’de ki gibidir.



Şekil 8.23. Dikey doğrunun paint programında gösterilmesi

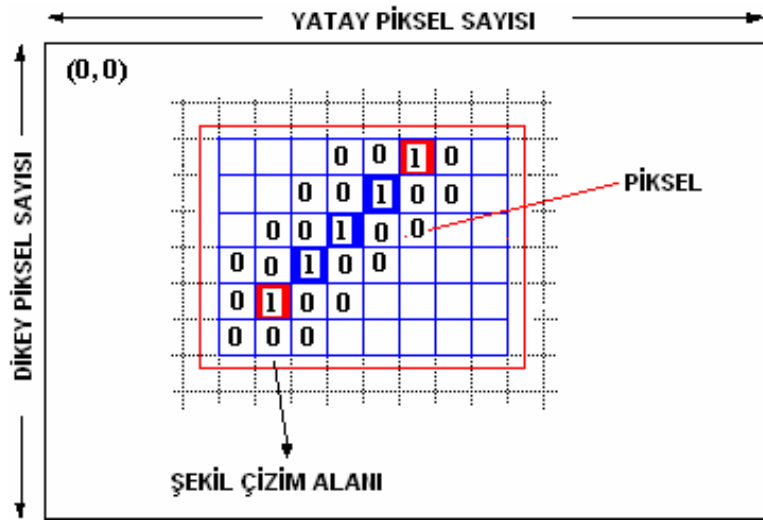
İlgili koordinatlar arası AddLine komutu ile birleştirilir. AutoCAD çalışır ve ekranda dikey çizgi belirmektedir (Şekil 8.24).



Şekil 8.24. AutoCAD programında dikey doğru uygulaması

Açılı doğru: Tarama yapıldığında 1 değerli ilk pikselin çarpaz komşu pikselleri 1 ise bu nesne açılı doğru olarak tanımlanır. Değeri 1 olan pikselin yatay ve dikey komşu piksellerin değeri 0 olur (Şekil 8.25). Piksel taraması While-Wend döngüsü kullanılarak yapılır. Son piksele ulaşınca işlem bitirilir.

Tarama sonrasında Y değerleri list2, list4 ve X değerleri list1, list3 listeleme kutularına listelenir (Şekil 8.26).



Şekil 8.25. Açılı doğru taraması

KOORDİNATLAR				
SIRA	X1	Y1	X2	Y2
1	27,2084	6,007067	5,300353	28,26855
2	26,8551	6,360424	5,300353	28,26855
1[x1,y1]77,17[x2,y2]15,80 2[x1,y1]76,18[x2,y2]15,80				

Şekil 8.26. Açılı doğru için başlangıç ve bitiş koordinatları

Liste1.list ve list2.list listeleme kutusunda X ve Y değerlerinde değişim olacaktır (Şekil 8.26)

$$X(1) <>= X(2) <>= X(3) <>= \dots X(n)$$

$$Y(1) <>= Y(2) <>= Y(3) <>= \dots Y(n)$$

$$List1.List(i) <>= List3.List(i + 1) <>= List3.list(i+n)$$

$$List2.List(i) <>= List4.List(i + 1) <>= List4.list(i+n)$$

Sonuçta şart aşağıdaki gibi olacaktır.

ElseIf List2.List(i) <= List2.List(i + 1) And List3.List(i) <= List3.List(i + 1) Then

X ve Y koordinatındaki değişim şartı sağlanınca açılı *doğru* meydana gelecektir. Koordinatlara doğru çizmek için başlangıç ve bitiş noktalarına değişken ataması yapılır.

Başlama ve bitiş noktası olarak PT değişkeni olarak ataması yapıldı. Başlangıç noktası (0 to 2) arasında tanımlandı.

Dim PT1(0 To 2) As Double

0:X koordinatı

1:Y koordinatı

2:Z koordinatı

Bitiş noktası (0 to 2) arasında tanımlandı. Yani 0 ile 2 arası (X, Y ve Z) koordinat olarak atanması yapıldı.

Dim PT2(0 To 2) As Double

Başlangıç ve bitiş nokta koordinatlarına değer ataması listeleme kutularından yapılır. Başlangıç noktası (PT1), list1 ve list3 listeleme kutusundaki değer ilk değeri olmaktadır. Bitiş noktası ise (PT2), list2 ve list4 listeleme kutusundaki değerlerdir.

PT1(0) = List1.List(i) 'Başlangıç X koordinatı

PT1(1) = List2.List(i) 'Başlangıç Y koordinatı

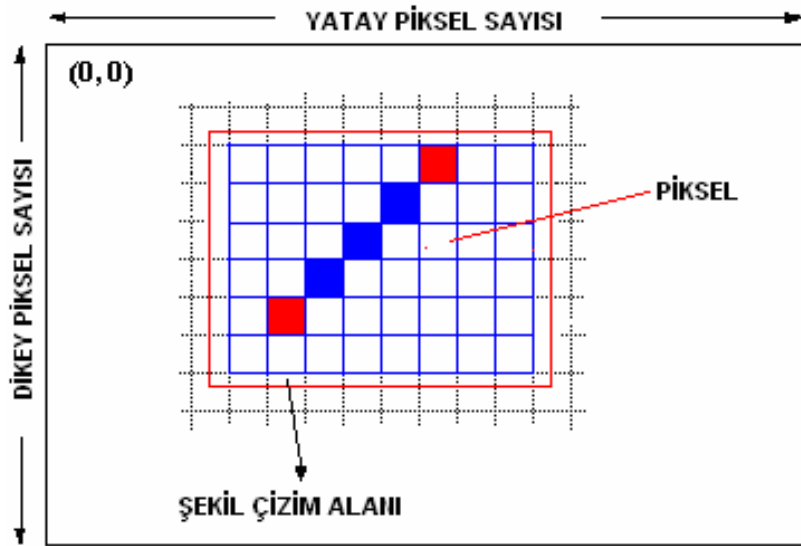
PT1(2) = 0 'Başlangıç Z koordinatı

PT2(0) = List3.List(i + n) 'Bitiş X koordinatı

PT2(1) = List4.List(i + n) 'Bitiş Y koordinatı

PT1(2) = 0 'Bitiş Z koordinatı

X artacak X+1 şeklinde ve Y artacak Y+1 şeklinde oluşacaktır. Sonunda ise bir nokta çıkılarak son nokta bulunacaktır (Şekil 8.27).



Şekil 8.27. Açılı doğru uygulaması

Tarama sonucunda 1 değeri alan ilk piksel ile son pikselin koordinatları başlama ve bitiş noktaları olarak kaydedilmektedir (Şekil 8.27).

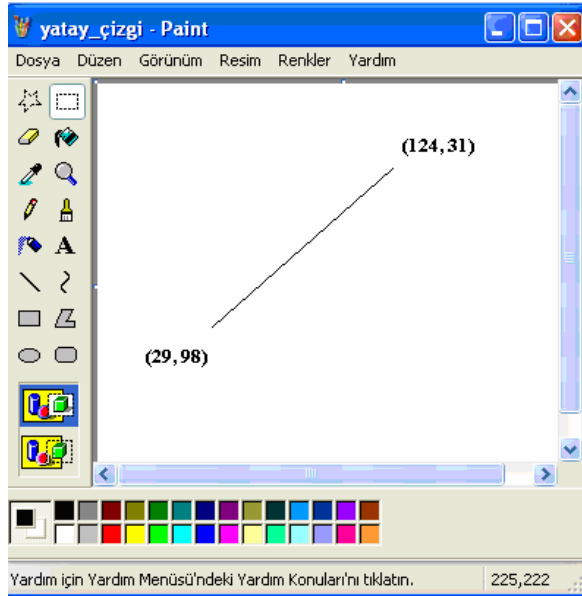
Programda taranan noktanın koordinatları başlama ve bitiş koordinatları olarak atanır. Buradaki rakamsal koordinatlar AutoCAD ortamında “addline” komutu kullanılarak doğruya dönüştürülmektedir. Değişken olarak nokta AutoCAD Visual Basic Application (VBA)komutu olarak atanır.

Dim objline As AcadLine

Listeleme kutularındaki koordinatlar arasına doğru çizmek için AcadLine değişkeninin tanımlanması gerekir. Kullanılan değişkeni atamak için Set komutu kullanılmalıdır. PT1 ve PT2 noktaları arasına Set komutu kullanarak doğru çizimi yapılır.

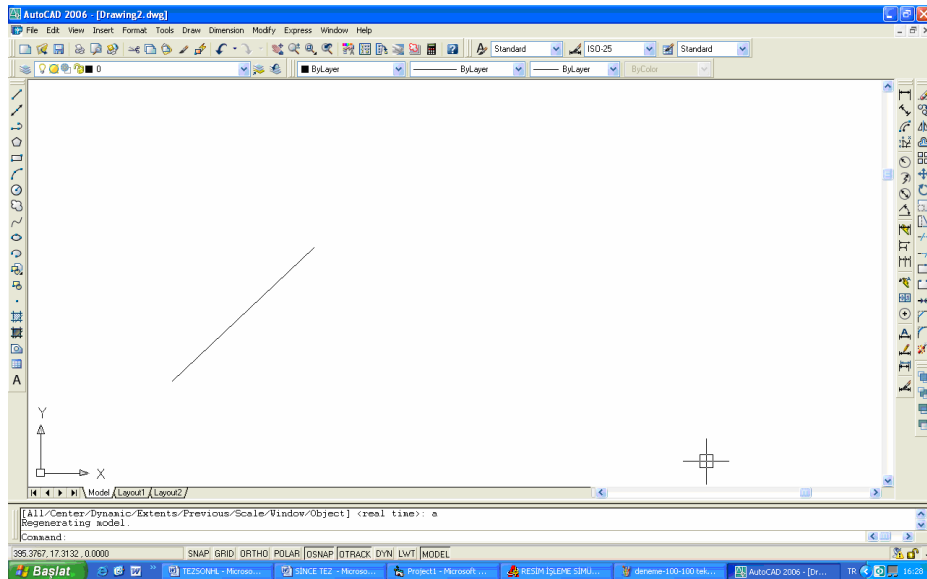
Set objline=acaddoc. ModelSpace. AddLine(PT1,PT2)

Açılı doğrunun paint programında görünümü Şekil 8.28' de ki gibidir.



Şekil 8.28. Açılı doğrunun paint programında gösterilmesi

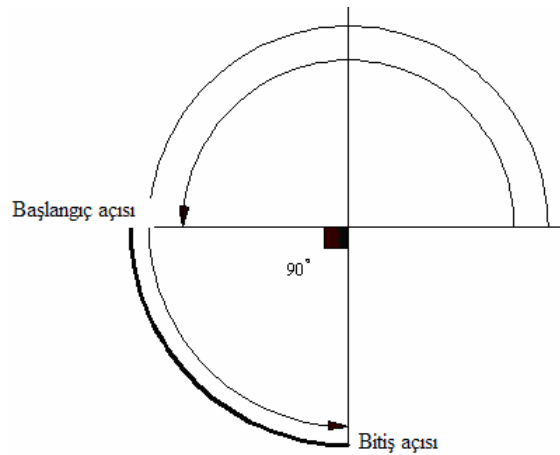
İlgili koordinatlar arası AddLine komutu ile birleştirilir. AutoCAD çalışır ve ekranda açılı çizgi belirir (Şekil 8.29).



Şekil 8.29. AutoCAD programında açılı doğru uygulaması

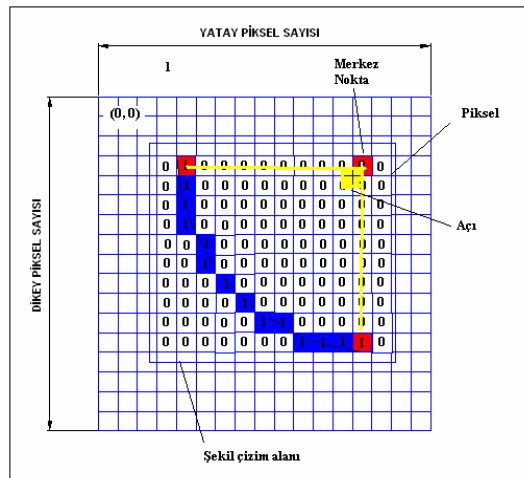
8.2.3. Yay ve çemberlerin yorumlanması

Resim yorumlanırken 1 değeri olan piksele komşu, 1 değerli diğer pikseller bir doğru üzerinde sıralanmıyor ve belli yarıçaplı bir yay şekli oluşturuyorsa bu nesne yay olarak tanımlanır (Şekil 8.30).



Şekil 8.30. Yay parçası

Şekil 8.30'deki yayın başlangıç açısı 180° ve bitiş açısı 270° dir. Şekil taranarak değeri 1 olan pikseller tespit edilir ve koordinatları hafızaya alınır (Şekil 8.31).



Şekil 8.31. Yay parçasında piksellerin durumu

Tarama yapıldıktan sonra list8 listeleme kutusuna X koordinatları, list9 listeleme kutusuna Y koordinatları yazdırılır (Şekil 8.32). Yay parçasının tanınması için i ve j olmak üzere değişkenler tanımlanır.

Dim i

Dim j

KOORDİNATLAR		
X VE Y	X	Y
0 X=50,Y=29	50	29
1 X=50,Y=30	50	30
2 X=50,Y=31	50	31
3 X=50,Y=32	50	32
4 X=51,Y=33	51	33
5 X=51,Y=34	51	34
6 X=51,Y=35	51	35
7 X=51,Y=36	51	36
8 X=51,Y=37	51	37
9 X=51,Y=38	51	38

Şekil 8.32. Yayın X ve Y koordinatları

Listeleme kutularında görüldüğü gibi X ve Y de değişim vardır. Ama artış başlangıçta belirli bir koordinata kadar X' de sabit, satırda Y'de Y+1 şeklindedir. Yayın orta kısımlarında ise hem X'de hemde Y'de artış vardır. Yayın son kısmında x'de artış olmasına karşın Y değeri sabit olabilir. Listedeki ilk elemanlar yayın başlangıç noktasını, son elemanlar yayın bitiş noktasını belirler. Başlangıç noktası için;

List8.List(0)

List9.List(0)

Son eleman için ise;

List8.list(i)

List9.list(j) komutları kullanılmaktadır.

Tüm noktalar $x^2+y^2=r^2$ çember formülünü yaklaşık olarak sağlıyorsa bu noktalar yardımı ile yayın yarıçapı belirlenir. Burada X, Y yayın üzerindeki bir noktanın koordinatları, Xc, Yc yay merkezinin koordinatları, r yay yarıçapı ve θ yay üzerindeki noktanın açısıdır.

Aşağıdaki formülden yararlanılır.

$$X = X_c + r \cdot \cos \theta$$

$$Y = Y_c + r \cdot \sin \theta$$

Listeleme kutularındaki koordinat sayıları i ve j değişkenlerine aktarılır.

$$i = \text{List8.ListCount} - 1$$

$$j = \text{List9.ListCount} - 1$$

Bu değişkenler yay elemanlarını bulmak için gereklidir. Tanımlanan merkez nokta (mn) diye tanımladığımız değer de üç adet koordinattan ibarettir.

Dim mn(0 To 2) As Double

0: X koordinatı

1: Y koordinatı

2: Z koordinatını ifade etmektedir.

$mn(0) = X_c$ 'X koordinatı

$mn(1) = Y_c$ 'Y koordinatı

$mn(2) = 0$ ' Z koordinatı

Yay parçasının yarıçapı olarak da;

Dim yarıcap As Double

$$\text{yarıcap} = \text{SQR} \left((\text{List8.List}(0))^2 + (\text{List9.List}(0))^2 \right)$$

Burada SQR karekök komutudur. İlk değer yayın ilk noktasının X koordinatı ve ikinci değer yayın ilk Y koordinatıdır. Yayın başlangıç ve bitiş açısı değerleri radyan cinsinden yazılır.

$$baslangiacisi = Pi$$

$$bitisacisi = 270*(Pi/180)$$

Burada başlangıç açısı 180^0 dir. Bitiş açısı ise 270^0 radyana çevirmek için $Pi/180$ ile çarpılmıştır. Yay parçasına AutoCAD ortamında çizilmesi için değişken aşağıdaki gibi tanımlanır.

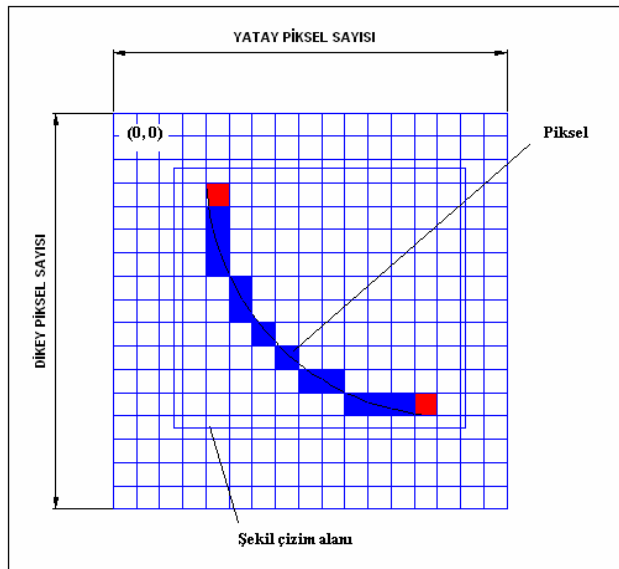
Dim yay As AcadArc

AutoCAD komutunun çalışması için AcadArc değeri atanır. AcadArc yay çizilmesi için gerekli bir değerdir.

Çizimin yapılması için set değişkenine yay komutu olarak atanır.

Set yay = acaddoc. ModelSpace. AddArc(mn, yaricap, baslangiacisi, bitisacisi)

Bu komut ile AutoCAD ortamında merkez noktası, yarıçapı, başlangıç ve bitiş açıları verilen yay parçası çizilir (Şekil 8.33).



Şekil 8.33. Yay parçası uygulaması

Resmin düz çıkması için aynalama kodları kullanılmaktadır. Aynalama işlemi için iki nokta gereklidir. Noktaları değişken olarak tanımlamak için aşağıdaki komutlar kullanılır.

Dim nokta1(0 To 2) As Double

Dim nokta2(0 To 2) As Double

Sonra yayın başlangıç ve bitiş koordinatları nokta1 ve nokta2 olarak atanır.

nokta1(0) = (List1.List(0)

nokta1(1) = (List2.List(0)

nokta1(2) = 0

nokta2(0) = List3.List(i)

nokta2(1) = List4.List(j)

nokta2(2) = 0

Aynalacak nesne olarak yay parçası atanır.

Dim aynalanacak_nesne As AcadArc

Ve son olarak set komutu ile aynalam işlemi gerçekleştirilir.

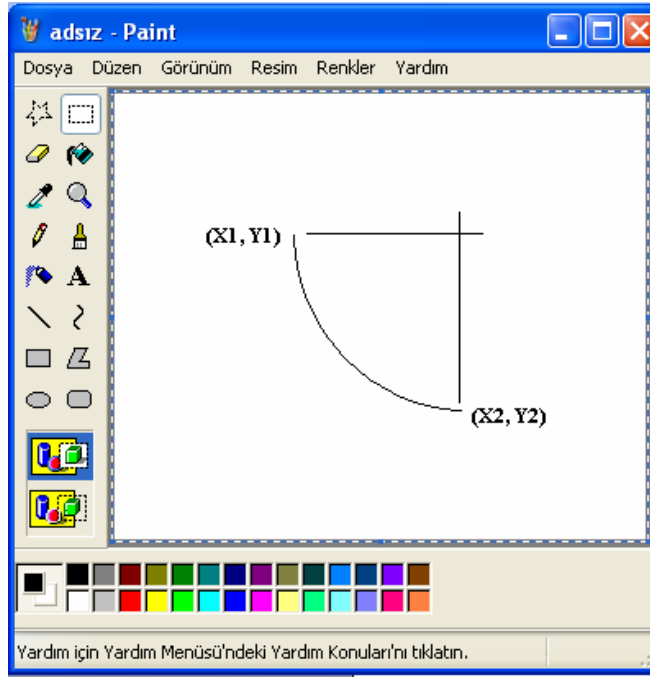
Set aynalanacak_nesne = yay. Mirror(nokta1, nokta2)

Son olarak yay parçasının rengi mavi yapılarak ve aynalanan nesne silindi.

aynalanacak_nesne. Color = acBlue

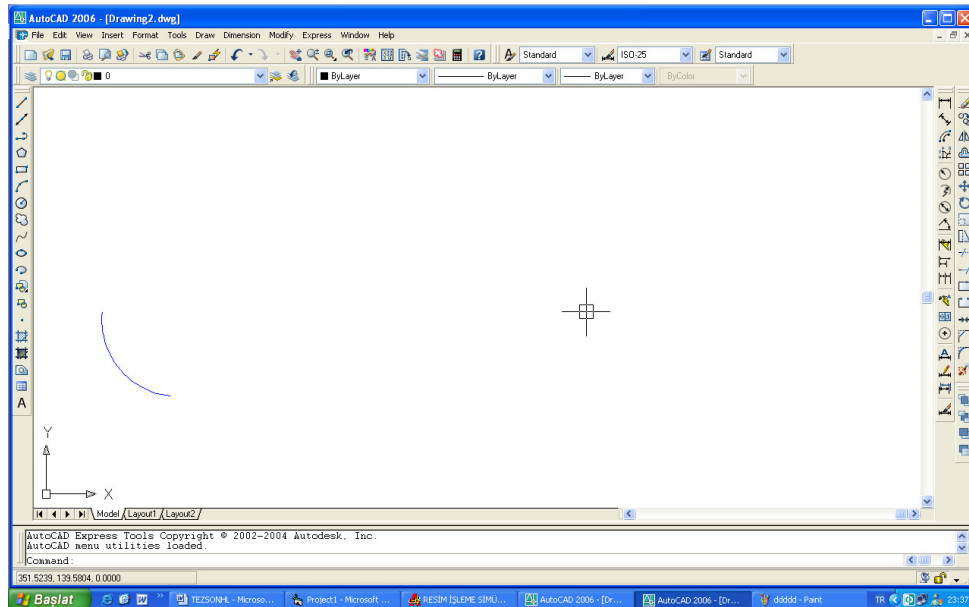
yay. Delete

Yay parçası paint programında resimdeki gibi görünür (Şekil 8.34).



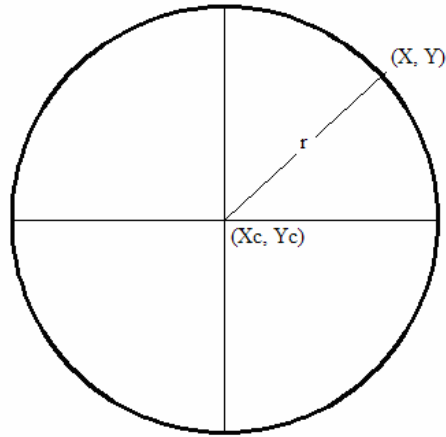
Şekil 8.34. Yay parçasının paint programında görüntülenmesi

Yay parçası AutoCAD ortamında ise (Şekil 8.35)' deki gibi görüntülenecektir



Şekil 8.35. AutoCAD ortamında yayın çizilmesi

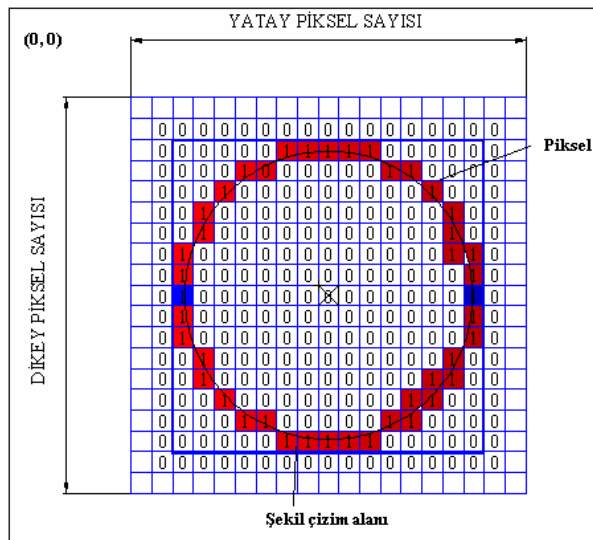
Çember: Çemberi çizebilmek için merkezi ve yarıçapının bilinmesi gerekir (Şekil 8.36).



Şekil 8.36. Çember ve elemanları

Değeri 1 olan pikseller aşağıdaki çember eşitliğini sağlarsa bu piksel grubu çember olarak kabul edilir (Şekil 8.37).

$$x^2 + y^2 = r^2$$



Şekil 8.37. Piksellerle oluşturulan çember

Tarama sonucunda çemberin dolu noktalarının X ve Y koordinatları listeleme kutularına yazılır (Şekil 8.38)

KOORDİNATLAR		
X VE Y	X	Y
41 X=14,Y=88	11	74
42 X=14,Y=89	12	75
43 X=15,Y=44	12	76
44 X=15,Y=45	12	77
45 X=15,Y=46	12	78
46 X=15,Y=90	12	79
47 X=15,Y=91	12	80
48 X=15,Y=92	12	66
49 X=16,Y=42	12	67
50 X=16,Y=43	12	68
	13	69
		70

Şeki 8.38. Çemberin X ve Y değerleri

Koordinatların ilk ve son değerini bulmak için i ve j değişkenlerine değer atanır.

Dim i

Dim j

$i = (List2.ListCount - 1)$

$j = (List3.listCoun t- 1)$ 'Listedeki son aktif elemanı gösterir.

Sonra çemberin elemanlarından olan merkez noktaları tanımlanır.

Dim merkez (0 To 2) As Double

Yay tanımlamada olduğu gibi çember merkez noktasının koordinatları girilir.

$merkez(0) = Xc$ 'X koordinatı

$merkez(1) = Yc$ 'Y koordinatı

$merkez(2) = 0$

Çemberin yarıçapı olarak da;

Dim yarıcap As Double

$$yaricap = SQR((List2.List(0)^2) + (List3.List(0)^2))$$

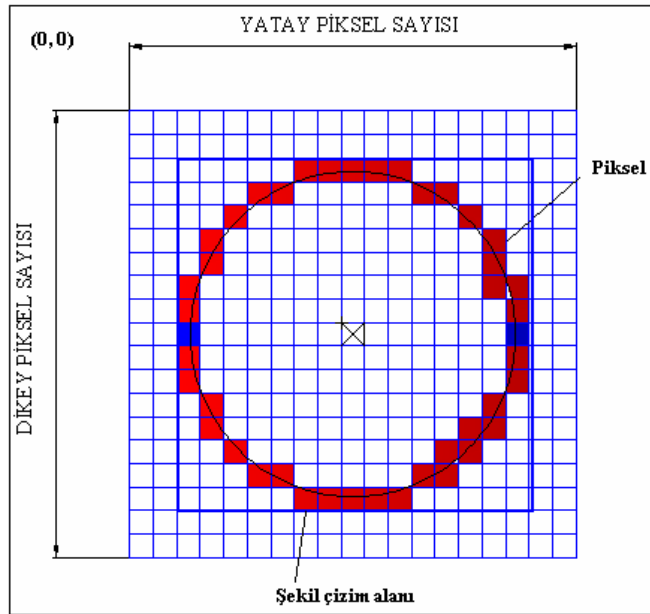
Çemberin AutoCAD ortamında çizilmesi için aşağıdaki satır yazılır.

Dim cember As Acadcircle

AddCircle çizimin yapılması için set değişkenine cember komutu olarak atanır.

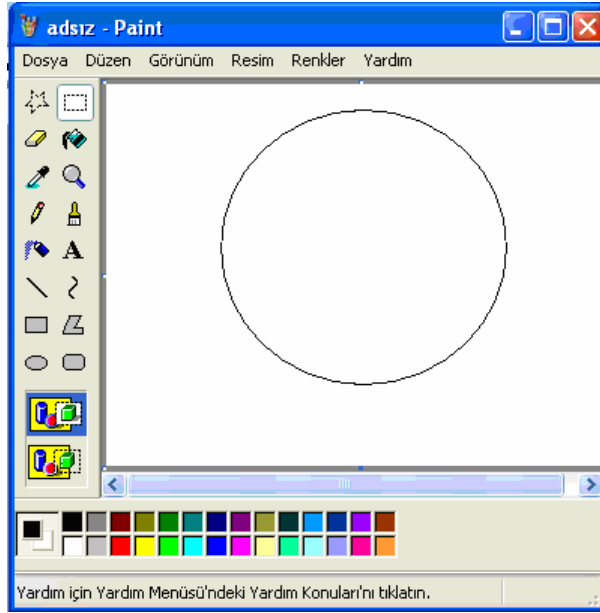
Set cember = acaddoc. ModelSpace. AddCircle(merkez, yarıcap)

Bu komutla AutoCAD ortamında merkez noktaları ve yarıçapı verilen cember çizilir (Şekil 8.39).



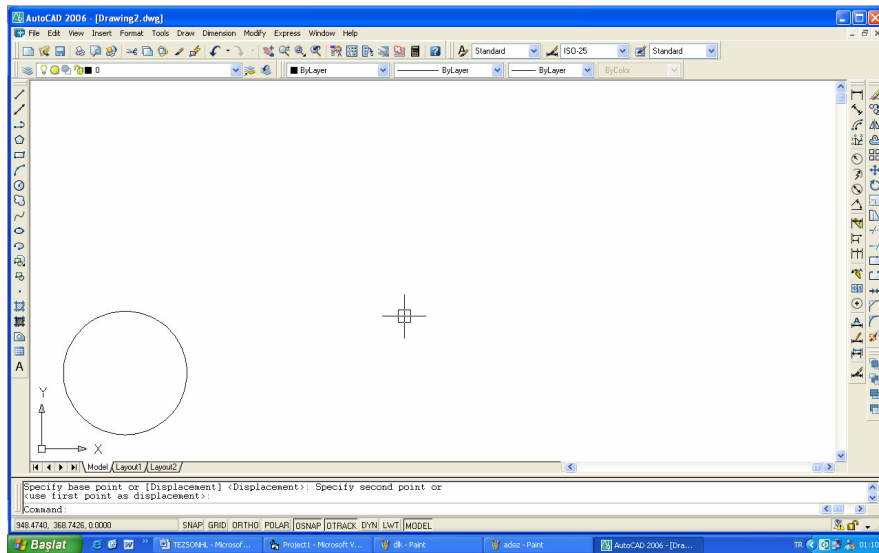
Şekil 8.39. Çember uygulaması

Çember paint programında resimdeki gibi görüntülenir (Şekil 8.40).



Şekil 8.40. Çemberin paint programında görüntülenmesi

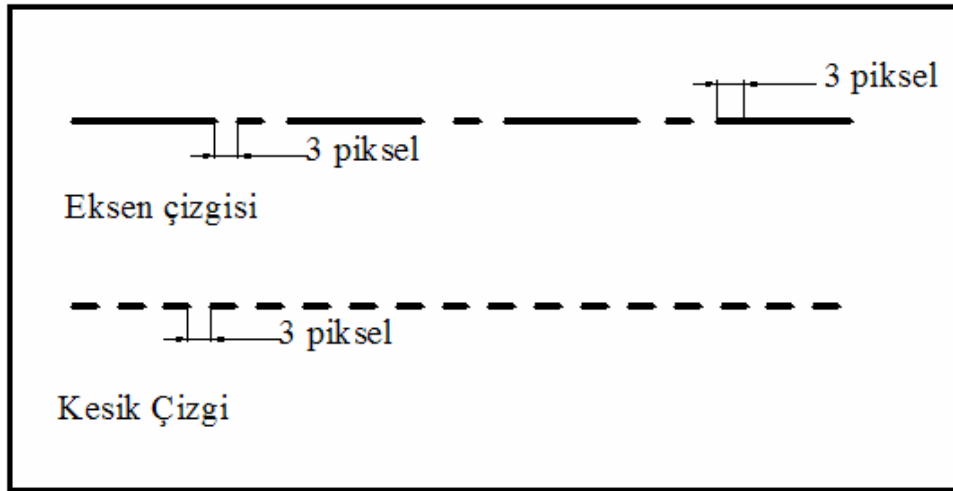
Çember AutoCAD ortamında ise Şekil 8.41' teki gibi görünecektir



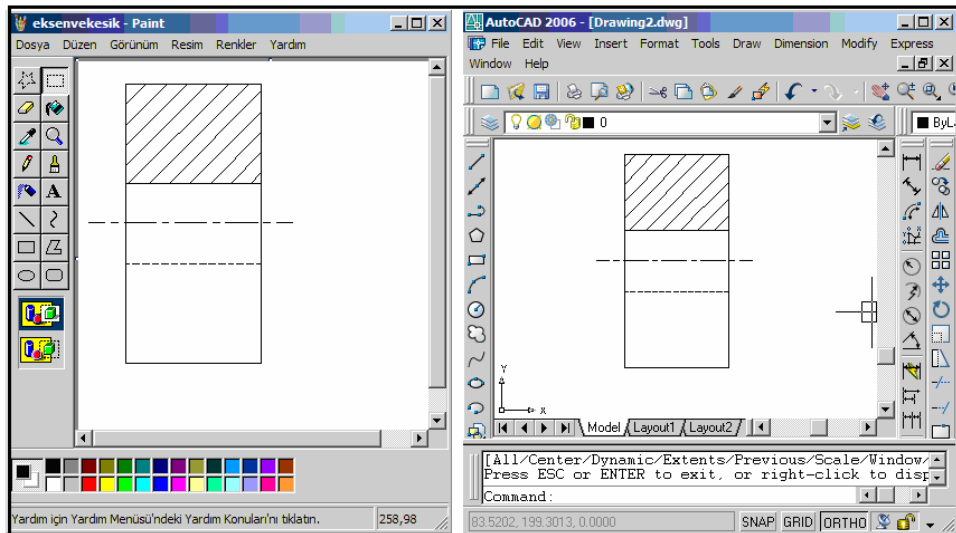
Şekil 8.41. AutoCAD ortamında çemberin çizilmesi

Eksen çizgisi ve kesik çizgi: Çizgi çizmek için minimum piksel sayısı 3 adet olarak sabit alınmıştır. Bunun sonucunda geliştirilen program 3 piksel ve altındaki pikselleri görmemekte ve çizim yapmamaktadır.

Bu da kesik veya eksen çizgisi oluşturmaktadır (Şekil 8.42). AutoCAD programında taranan şekil açılabilir (Şekil 8.43).

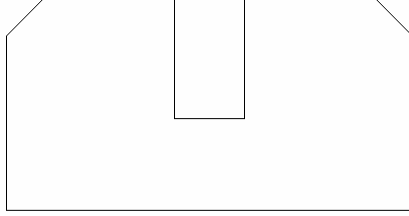


Şekil 8.42. Eksen ve kesik çizgi



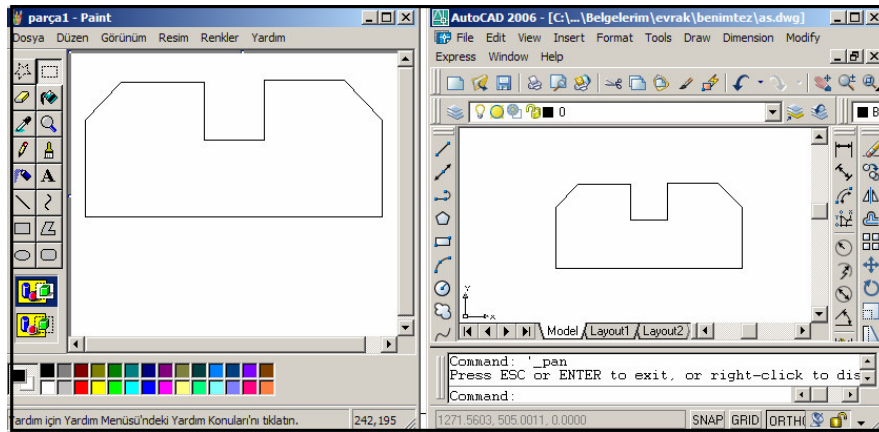
Şekil 8.43. Eksen ve kesik çizgi uygulaması

Parça çizimi: Parça profillerini oluşturan nesneler tek tek yorumlanarak tanımlama işlemi yapılır ve ekranda çizilir (Şekil 8.44).



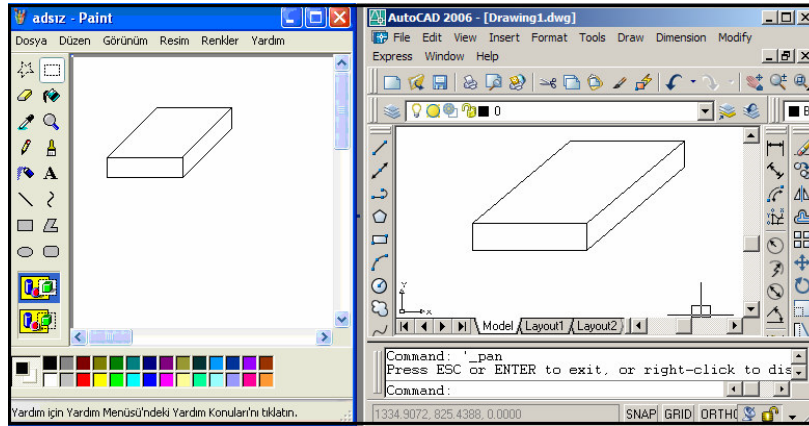
Şekil 8.44. Parça resmi

Resimde işlenen piksel koordinatları listeleme kutularında belirir. Noktaların koordinatları X ekseninde değişiyor ve Y değerleri sabit ise yatay doğru çizilir. Noktaların koordinatları Y değerinde değişiyor ve X değeri sabit ise dikey doğru çizilir. Noktaların koordinatları X ve Y değerleri ikisinde değişiyorsa açılı doğru çizilir. Yay parçası tanımlanmasında ise, yayın başlangıç ve bitiş açlarına, merkez noktasına ve yayın yarıçapına ihtiyaç vardır. Parça içinde iken yayın en az üç noktasının bulunması gerekmektedir. Y ekseninde değer artarken X ekseninde değer azalmaktadır. X ekseninde değer artarken ise Y yine azalmaktadır. Tamamen ters orantı vardır. Çemberde ise merkez noktasına ve çemberin yarıçap değerine ihtiyaç vardır. Piksel olarak taranan nesne koordinatlarına atanan çizim nesneleri ile resim meydana gelmektedir (Şekil 8.45).



Şekil 8.45. Paint ve AutoCAD programında doğrulardan oluşmuş parça

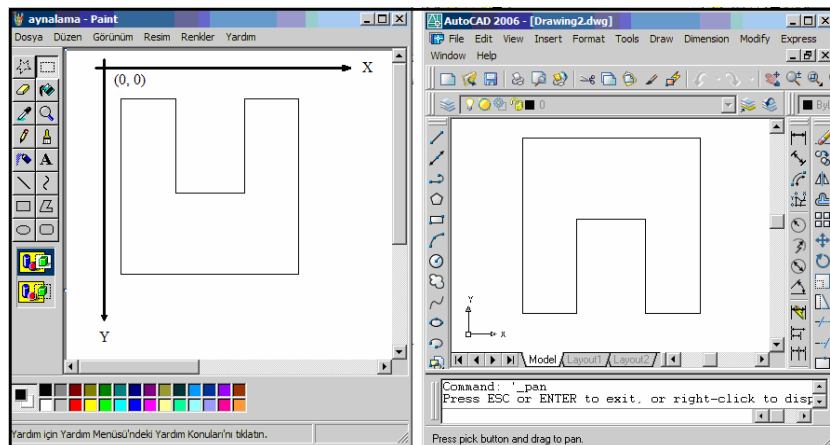
Prizmatik parçalar için ise yatay, dikey ve açılı doğru tanımlanarak çizim yapılır (Şekil 8.46).



Şekil 8.46. Paint ve AutoCAD programında prizmatik parça gösterilmesi

8.3. Aynalama

Tasarlanan programda işlenen Bitmap resmin koordinatları kullanılarak ActiveX komutları ile çizim AutoCAD ortamına aktarılmaktadır. Fakat bitmap resmin koordinat başlangıcı sol üst köşe, AutoCAD programının ise sağ alt köşedir. Yani çizim 180° ters oluşmaktadır. X ekseninde herhangi bir problem bulunmamaktadır problem Y eksenindedir (Şekil 8.47).



Şekil 8.47. Aynalama uygulaması

Çizimin Y ekseninde yönü (- List2.list(i)) ters belirttiğimiz zaman çizim orijinal hali Şekil 8.48’deki AutoCAD ekranında görüntülenecektir.

Fakat çizim başlama noktasından aşağıda belirtmektedir. Bunun için ise listedeki son eleman sayısını 5 katsayısı ile çarparak başlama noktasından yukarı tarafda çıkmasını sağladık. Bu katsayı resim çizim halinde iken AutoCAD ekranında negatif bölgede çıkmaması için tahminen kullanılmaktadır.

$$sp(0) = List1.List(i)$$

$$sp(1) = -List2.List(i) + (5 * List2.ListCount) \text{ ‘Çizimin doğru çıkması için}$$

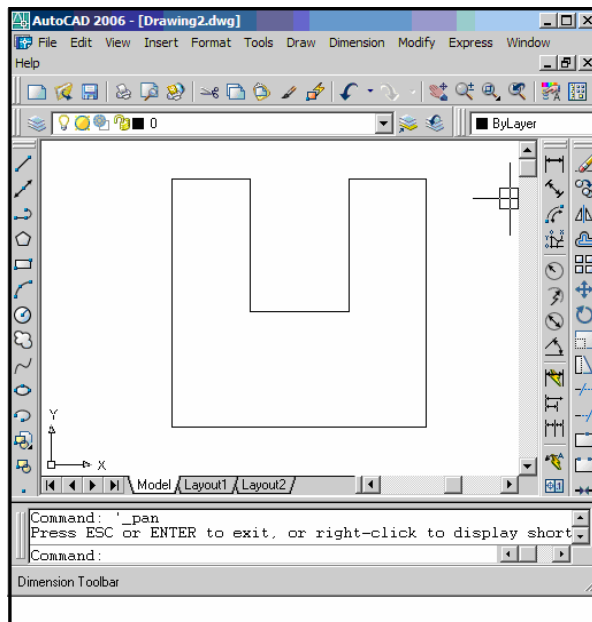
$$sp(2) = 0$$

$$ep(0) = List3.List(i)$$

$$ep(1) = -List4.List(i) + (5 * List4.ListCount) \text{ ‘Çizimin doğru çıkması için}$$

$$ep(2) = 0$$

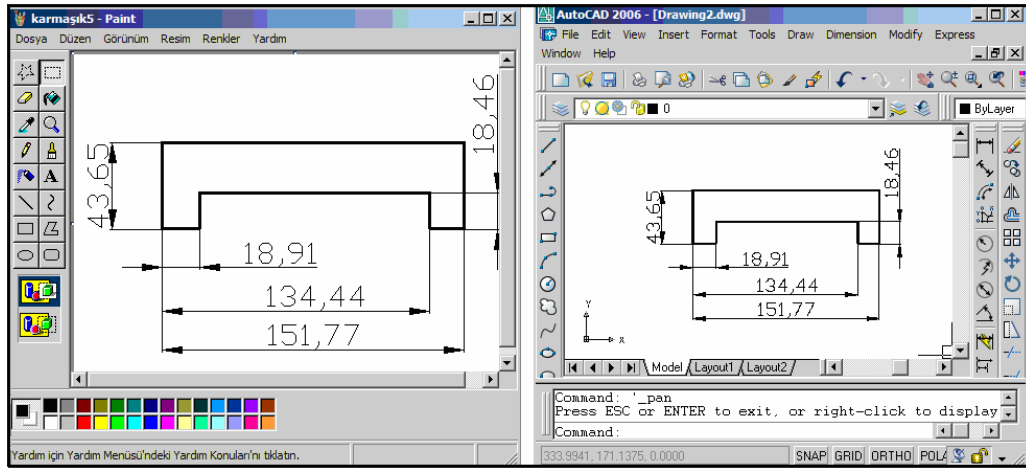
Yukarıdaki kodlar sonucunda aşağıdaki çizim görüntülenmektedir.



Şekil 8.48. Aynalama sonucu çizimin gösterilmesi

8.4. Karmaşık Parçaların, Ölçülerin ve Yazıların Noktasal Olarak Çizilmesi

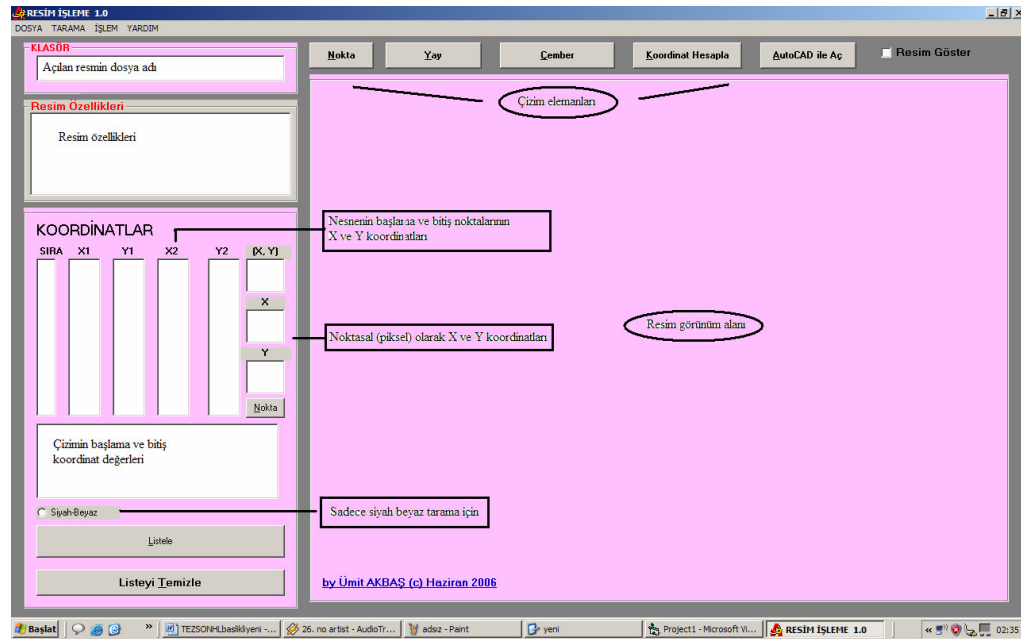
Çok karmaşık parçalar geliştirilen programda çizilememektedir. Fakat her pikselin koordinatına nokta atanarak çizim noktasal olarak yapılabilir. Fakat ölçülerin ve yazıların belirli bir büyüklüğünün olması gerekmektedir (Şekil 8.49).



Şekil 8.49. Ölçülendirilmiş parça çizimi

9. PROGRAMIN KULLANIMI

Program çalıştırılınca ekrana Şekil 9.1’ deki ana menü gelir. Ekranda dosya isminin görüneceği Text kutucuğu vardır. Klasöre erişim için Commondialog nesnesi kullanılmıştır. Üzerinde çalışılacak resim dosyası açılınca klasör başlığı altındaki metin kutusunda dosya adı görüntülenmektedir.

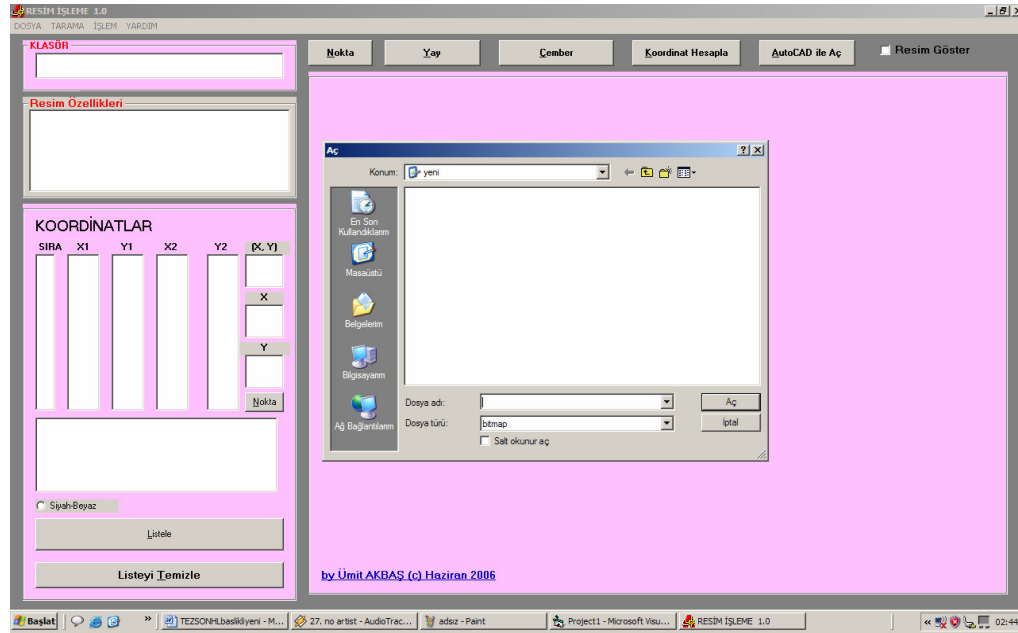


Şekil 9.1. Program ana penceresi

Resim seçme işlemi “DOSYA” kayar menüsü altında “Aç” komutu ile yapılmaktadır.

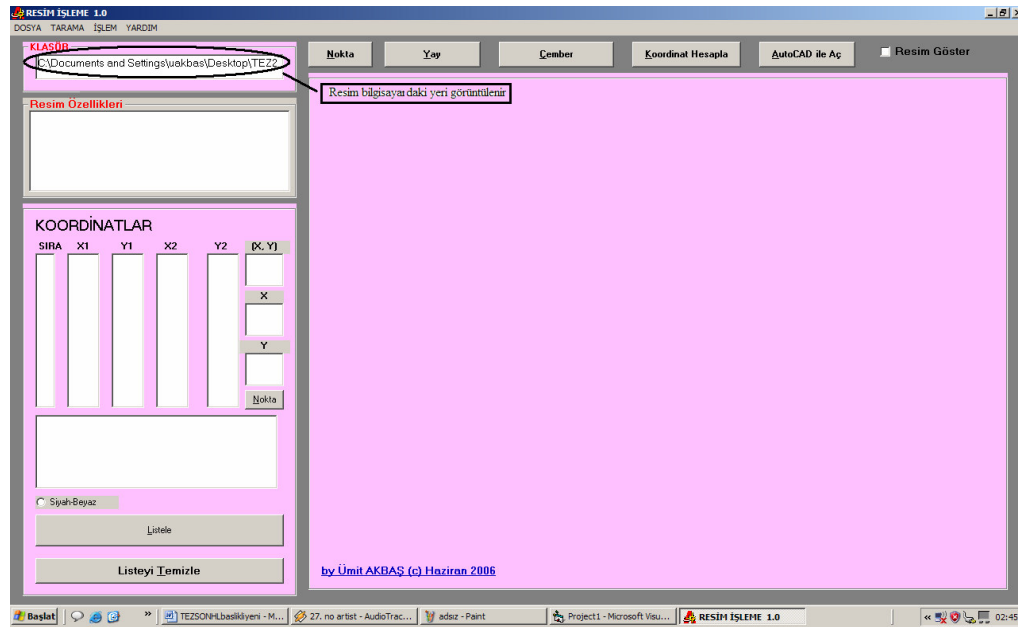
Bu düğme tıklandığında kullanıcının bilgisayarın herhangi bir dizininde bulunan bmp uzantılı resmi seçmesi için “Windows resim aç” penceresi açılacaktır.

Aç penceresinden istenilen resim seçilir. “Aç” düğmesine basılınca resim seçilmiş olur (Şekil 9.2).



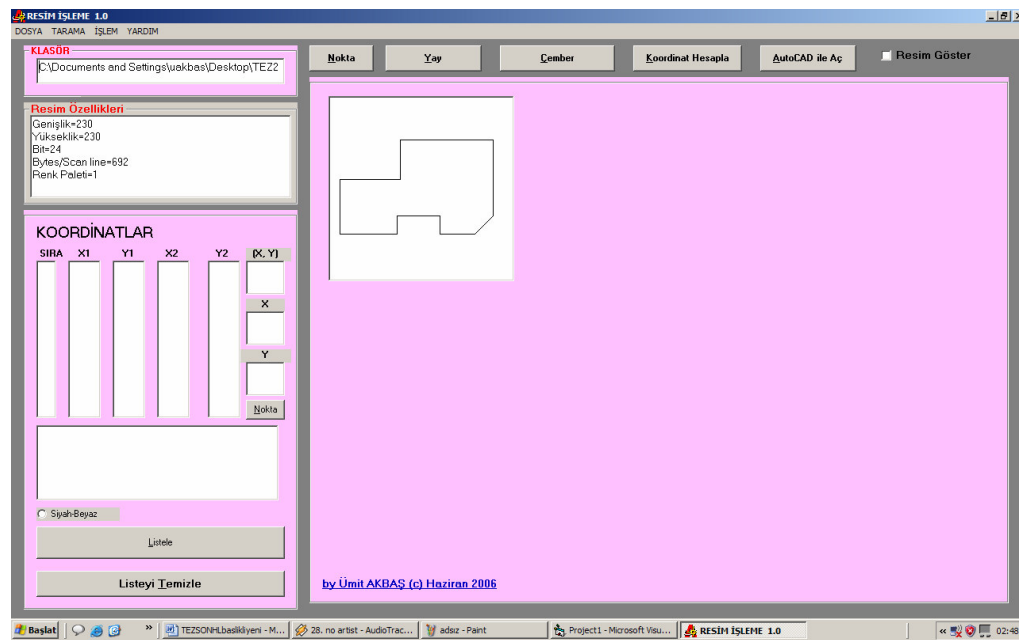
Şekil 9.2. Dosya açma penceresi

Seçilen resmin bulunduğu klasör text kutusunda görünür (Şekil 9.3).



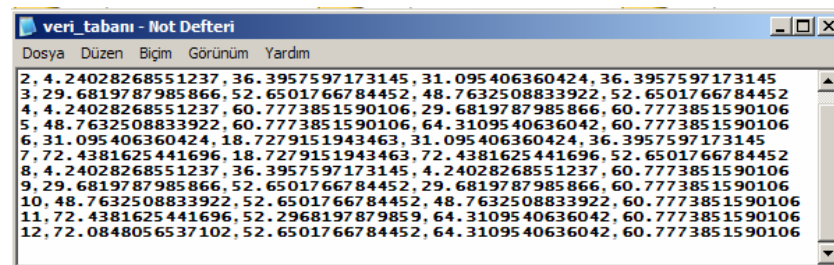
Şekil 9.3. Dosya isminin görüntülenmesi

Tarama menüsü altından “Resim özellikleri” tıklanır. Resim özellikleri text kutucuğunda özellikler belirir. “Resim göster “ komutu tıklandığı zaman “Resim göster” ekranında resim görünür.

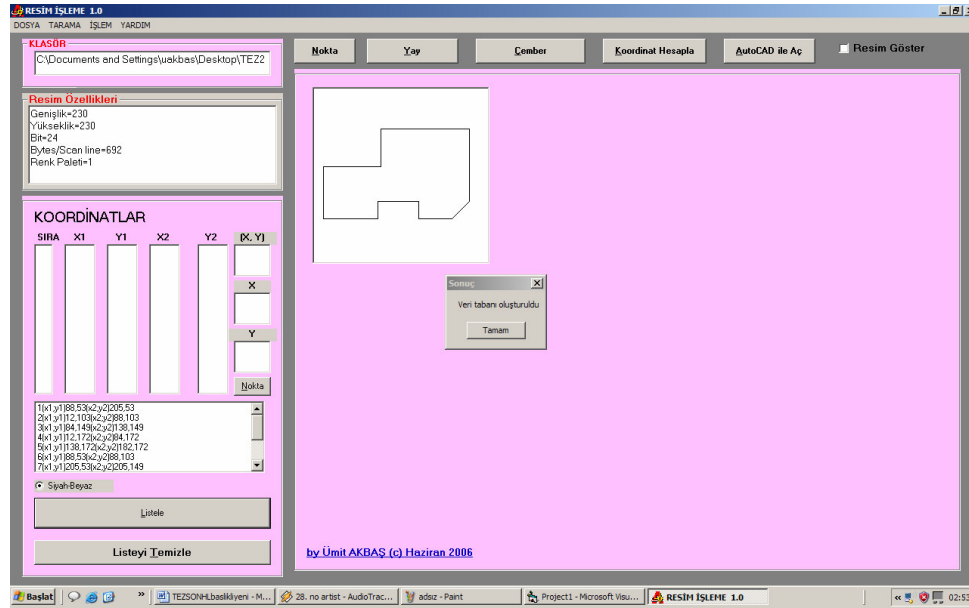


Şekil 9.4. Resim işleme menüsü

“Resim göster” menüsünden veya “Resim göster” onay kutucuğu işaretlenir. Resim görüntülenir. “Resim göster” menüsü ile “Resim göster” onay kutusu aynı işlevi yapmaktadır. Listelerin altındaki “Listeyi sil” düğmesi ile listedeki veriler silinebilir (Şekil 9.4). Listele düğmesi tıklandığı zaman ise nesnelerin koordinat değerleri liste kutularında görüntülenir (Şekil 9.6).

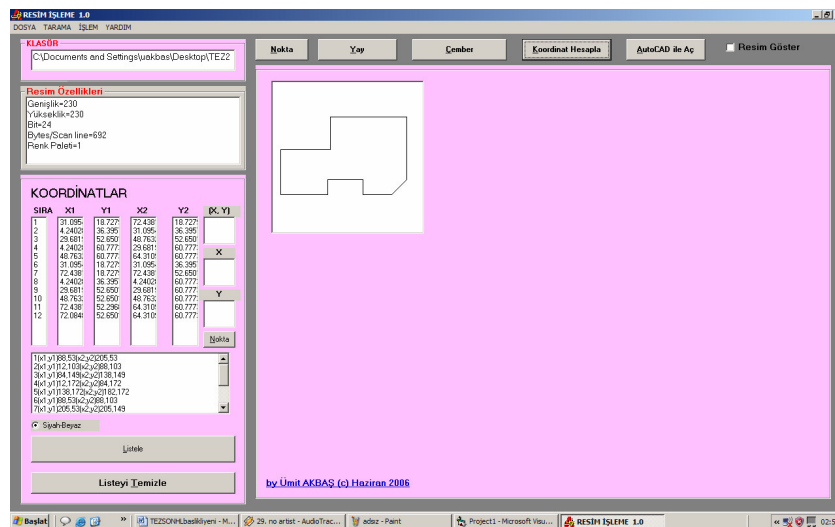


Şekil 9.5. Txt veri oluşturulması

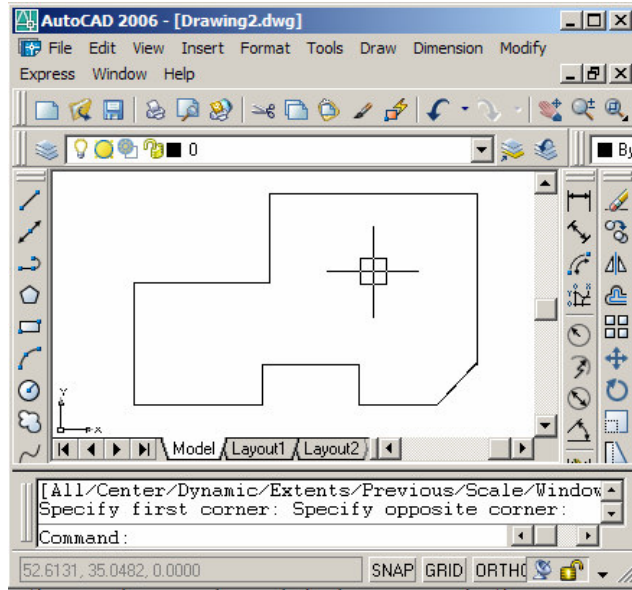


Şekil 9.6. İşlem menüsü

Ekranda görünen siyah-beyaz onay kutusu seçilir. Listele düğmesine tıklanır. List kutusunda başlangıç ve bitiş nokta koordinatları görüntülenecektir. Tamam, düğmesi tıklanır veri tabanı txt olarak oluşturulur (Şekil 9.5). Koordinatları hesapla düğmesi seçilerek X1, Y1, X2 ve Y2 list kutularına koordinatlar girilir. AutoCAD 'de açılarak görüntülenir (Şekil 9.7).

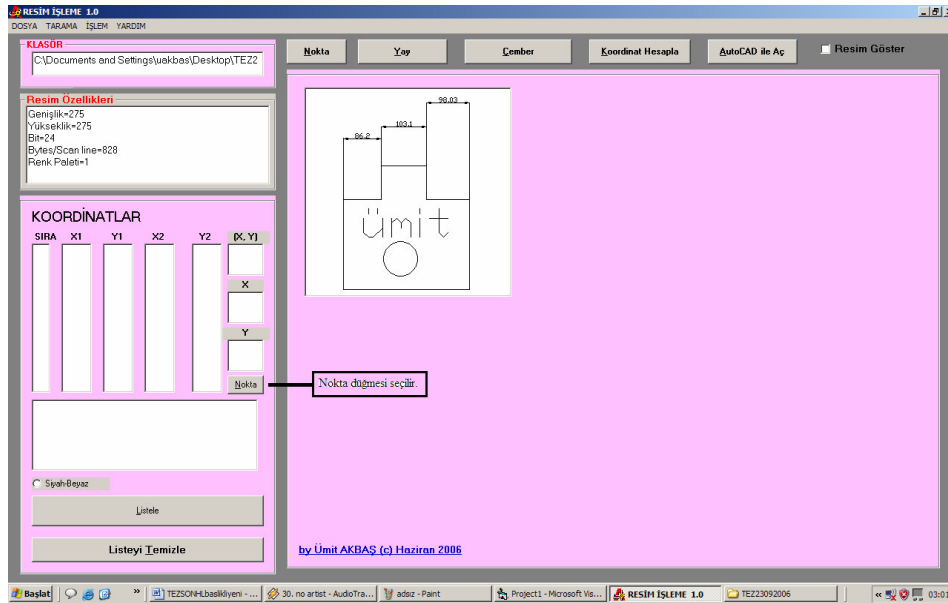


Şekil 9.7. Başlangıç ve bitiş noktalarının listelenmesi



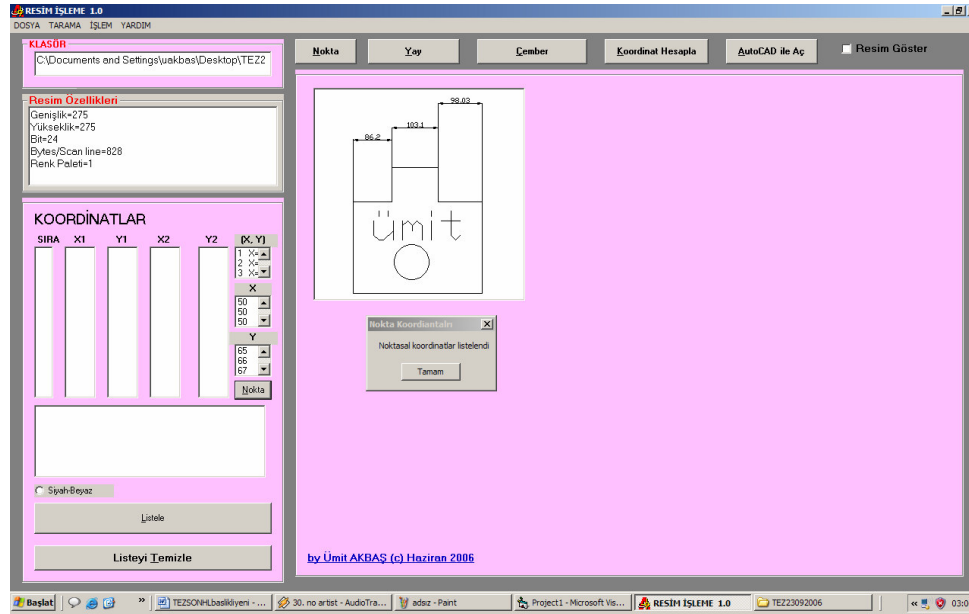
Şekil 9.8. AutoCAD programına resmin gönderilmesi

Noktasal olarak bir resmi AutoCAD ortamında açmak için ise resim açıldıktan sonra Nokta düğmesi tıklanmalıdır (Şekil 9.9).



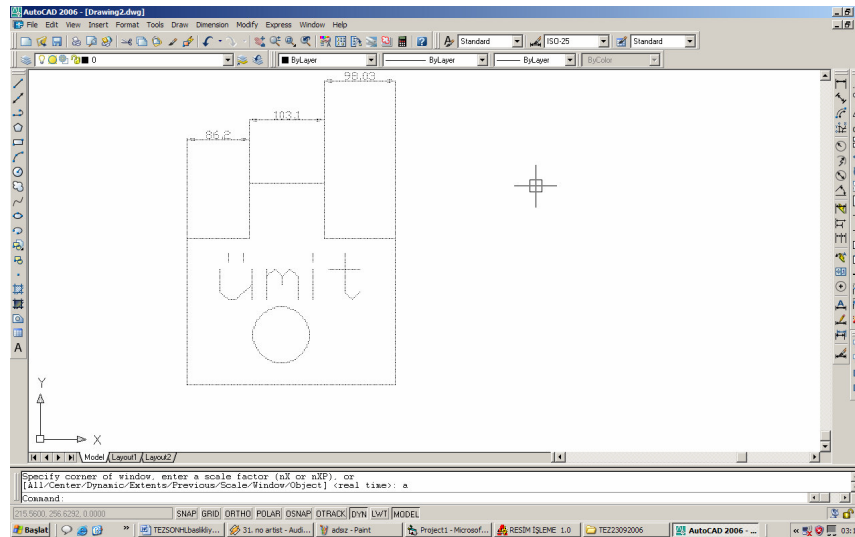
Şekil 9.9. Nokta koordinatı uygulaması

X ve Y listeleme kutularında noktasal koordinatlar belirlemektedir (Şekil 9.10).



Şekil 9.10. Nokta koordinatları

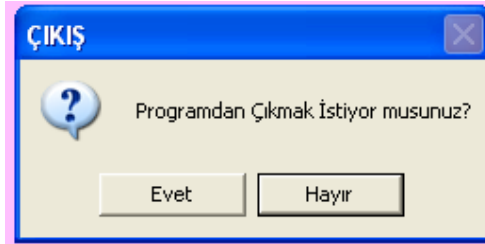
Üstteki Nokta düğmesi tıklandığı zaman veriler Şekil 9.11’de ki gibi AutoCAD ortamına gönderilir.



Şekil 9.11. AutoCAD ortamına nokta koordinatların gönderilmesi

“Dosya” kayar menüsü altında “Çıkış” seçildiği zaman alttaki diyalog kutusu ortaya çıkar.

“Evet “ seçilirse programdan çıkılır. Köşedeki çarpı işaretinden de programdan çıkılabilir (Şekil 9.12).



Şekil 9.12. Çıkış menüsü

10. SONUÇ VE ÖNERİLER

Önceden elle çizilmiş veya elde çtısı olan ancak CAD ortamında dosyası olmayan resimler taranarak bmp resimler geliştirilen sistemle yorumlanıp AutoCAD ortamına aktarılmaktadır. Böylece teknik resimlerin CAD ortamında yeniden çizilmesi yerine geliştirilen program aracılığı ile otomatik olarak CAD çizimleri elde edilelerek zaman ve iş kaybı tasarrufu sağlanmaktadır. Çalışmanın sonucunda aşağıdaki sonuç elde edilmiş öneriler sunulmuştur.

10.1. Sonuçlar

- 1) Bmp uzantılı iki boyutlu ve izometrik resimler yorumlanarak AutoCAD ortamında çizilebilmektedir.
- 2) Sadece iki boyutlu nesneler programda işlenebilmektedir. Doğru, yay, çember vb nesneler ayrı ayrı işlenebilmektedir.
- 3) Bu çalışmada siyah ve beyaz dışındaki renkler ihmal edilmiştir.
- 4) Taranan nesnelerin referans noktaya göre tüm çizim öğelerinin çizim değerleri (başlangıç ve bitiş noktaları, yarıçap merkez noktaları vb.) belirlenebilmekte, işlemler bu yeni değerler üzerinden oluşturulmaktadır.
- 5) Oluşturulan tüm veriler alt dosya olarak saklandığından, kullanıcı daha sonra bu verileri kullanabilmektedir.
- 6) Karmaşık profilli şekiller noktalar ile AutoCAD ortamında açılabilir.

10.2. Öneriler

- 1) Diğer dosya formatlarının tümünü okuyabilen herhangi bir tarama modülü oluşturulabilir.
- 2) Tüm değerler sütunlar halinde bölümlendirilebilir istenildiğinde dosyalar içindeki herhangi bir veriye ulaşılabilme ve verinin alınabilme imkânı sunulabilir.
- 3) Program sonucunda elde edilen verilere göre üç boyutlu nesneler oluşturabilir.
- 4) Program verileri, kapsamlı bir veritabanı kullanılarak (Microsoft Access, SQL gibi) daha kullanışlı hale getirilebilir.
- 5) Program tasarımında başka programlama dilleri (C, C++ gibi) kullanılarak esneklik sağlanabilir.
- 6) Renkli çizimlerin piksel piksel okutulup işlenmesi sağlanabilir.
- 7) Yazı, ölçülendirme, konum ve yüzey toleranslarını yorumlama eklenebilir.

KAYNAKLAR

1. Ceylan D, “Paint Brush Programında Çizilmiş Eğrilerin Koordinatlarını Bulan Bilgisayar Programı”, Yüksek Lisans Tezi, *G. Ü. Fen Bilimleri Enstitüsü*, Ankara, 30–45, (1995).
2. Idesawa, M., “A System to Generate A Solid Figure From Three View”, *Bulletin of JSME*, 16: 216–225, (1973).
3. Akbulut T., Kurt O. , “Taranmış Görüntü Üzerindeki Kontrol Nokta Doğruluklarının Artırılması İçin Basit Algoritmalar”, *TMMOB Harita ve Kadastro Mühendisleri Odası 10. Türkiye Harita Bilimsel ve Teknik Kurultayı*, Ankara, 1-6, (2005).
4. Glynn Earl F. ,Thorpe Danny , “Manipulating Pixels With Delphi's ScanLine Property”, *Delphi Developer*, 1: 715–722, (1998).
5. Yaman K., Sarucan A., Atak M., Aktürk N., “Dinamik Çizelgeleme İçin Görüntü İşleme Ve Arıtma Modelleri Yardımıyla Veri Hazırlama”, *Gazi Üniv. Müh. Mim. Fak. Dergisi*, 16 (1) : 19–40, (2001).
6. Akbulut T., Eser M., “Raster Bilgilerin Vektör Bilgilere Dönüşümü”, Diploma Çalışması, ZKÜ Mühendislik Fakültesi, *Jeodezi ve Fotogrametri Mühendisliği Bölümü*, Zonguldak, 10-56, (2004).
7. Silver Bill, “An Introduction to Digital Image Processing”, *Technology Officer Cognex Corporation, Modular Vision Systems Divisio*, Natick, 1–9, (2000).
8. The MathWorks, “Matlab Image Processing Toolbox”, Natick, 18–24, (2005).
9. Kılıçarslan, E., “Taranan Resim Ve Projelerin Cad Ortamına Aktarılması”, *Kocaeli Deprem Sempozyumu*, Kocaeli, 139, (2005).
10. Başak, H, Şahin İ, “VBA ve AutoCAD Uygulamaları”, *Asil Yayın Dağıtım*, Ankara, 159–166, (2005)
11. Pal S. K. “Fuzzy sets in image processing and recognition”, *In Proc. First IEEE International Conference on Fuzzy Systems*, San Diego, 119–126, (1992).
12. Raphael, J. & Greenberg, R. “Image Processing: A state-of-the-art way to learn science”, *Educational Leadership* 53(2): 34–37 (1995).
13. Boomgard, R., Richard van B., “Methods for Fast Morphological Image Transforms Using Bitmapped Images,” *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 54 (3): 252–254, (1992)

14. Adams, R., “Radial Decomposition of Discs and Spheres,” *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 55(5): 325–332, (1993).
15. Karagülle, İ, Pala, Z, “Visual Basic 6.0”, *Türkmen Kitabevi*, İstanbul, 130–302, (2001).
16. Arslan H., Arslan O., “Uygulamalı AutoCAD”, *Özkan Matbaacılık*, Adana 10-24, (2005).
17. Sakurai M., Kurihara Y., and Karasawa S. “Color classification using fuzzy inference and genetic algorithm”, *In Proc. Third IEEE International Conference on Fuzzy Systems*, Orlando, 1975–1978, (1994).
18. Ankenbrandt C. A., Buckles B. P., and Petry F. E. “Scene recognition using genetic algorithms with semantic nets”, *Pattern Recognition Letters* 11(4): 285-293, (2004).
19. Ünsaçar, F., Çoklar A.N., “Cadd-Cam Bilgisayar Destekli Çizim ve Üretimin Temelleri”, *Atlas Yayın Dağıtım*, İstanbul, 45-78, (2004).
20. Brandimonte, M.A., Hitch, G.J., & Bishop, D.V.M. “Influence of Short-Term Memory Codes on Visual Image Processing: Evidence from Image Transformation Tasks”, *Journal of Experimental Psychology*, 18(1): 157-165 (1992).
21. Bourke, P., “Bitmap formats”, *A Beginners Guide to Bitmaps*, 1-12, (1993)
22. Julie A. K., “Amatörler İçin Digital Fotoğrafçılık”, *Dünya Yayıncılık*, İstanbul, 5-15, (2001)
23. Caylor, D., “Photoshop 5.5”, *Thinkden*, 13-16, (2001)

EKLER

EK-1 AutoCAD çizim sözlüğü

2D- 2 BOYUT
3D- 3BOYUT
2P- 2 NOKTA
3P- 3 NOKTA
3DARRAY- 3BOYUT DİZİ
3DFACE- 3BOYUT YÜZ
3DPOLY- 3BOYUT BİLEŞİK-ÇİZGİ
ADD- EKLE
ALIGN- HİZALA
ALL- TÜMÜ
ANGLE- AÇI
ANNOTATION- KISALTMA
APEX- TEPE
ARC- YAY
AREA –ALAN
ARRAY- DİZİ
ARROW- OK
ASPECT- ORAN
ATTDEF- NİTELİK TANIMI
ATTDISP- NİTELİK GÖRÜNTÜLE
ATTEDIT- NİTELİK DÜZENLE
ATTEX- NİTELİK ÇIKARIMI
AUTO- OTOMATİK
BLIPMODE -İŞARET MODU
BLOCK- BLOK
BLUE- MAVİ
BOTTOM- ALT
BOTTOMCENTER- ALTMERKEZ
BOTTOMLEFT- SOLALT
BOTTOMRIGHT- SAĞALT
BOX- KUTU
BREAK –KIR
BYBLOCK- BLOKDAN
BYLAYER- KATMANDAN
CAMERA- KAMERA
CDF- VİRGÜL AYRAÇLI FORMAT
CENTER- MERKEZ
CHAIN- ZİNCİR
CHAMFER –PAHKIR
CHANGE- DEĞİŞTİR
CHPROP -ÖZELLİK DEĞİŞTİR
CIRCLE –ÇEMBER
CIRCUMSCRIBED- ÇEMBERE DIŞTAN TEĞET
CLIP- KIRP
CLOSE- KAPAT
COLOR -RENK-
COMMANDS- KOMUTLAR
CONE- KONİ
CONSTANT- SABİT
COPY- KOPYALA
CORNER- KÖŞE

EK-1 (Devam) AutoCAD çizim sözlüğü

CUBE- KÜP
CURRENT- GÜNCEL
CYAN- TURKUVAZ
CYLINDER –SİLİNDİR
DECURVE- EĞRİ KALDIR
DELETE- YOKET, SİL
DELTA- DELTA
DIAMETER- ÇAP
DIRECTION- YÖN
DISH- YARIM KÜRE
DISPLAY- GÖRÜNTÜLE
DISTANCE- UZAKLIK, MESAFE
DISTANCES- MESAFELER
DIVIDE- BÖL
DOMED- TERS YARIM KÜRE
DRAGMODE- SÜRÜKLEME MODU
DTEXT- DİNAMİK YAZI
DVIEW- DİNAMİK GÖRÜNÜM
DXF- ÇİZİM ARA DEĞİŞİM FORMATI
DYNAMIC- DİNAMİK
EDGE- KENAR
EDIT- DÜZENLE
ELEVATION- YÜKSEKLİK
ELLIPSE- ELİPS
ELLIPTICAL- ELİPTİK
END- SONNOKTA
ENTITIES- ÖGELER
ENTITY- ÖGE
EXIT- ÇIK
EXTEND- UZAT
EXTENTS- KAPSAM
EYE- GÖZ
FILLED- YUVARLAT
FIRST- İLK NOKTADAN
FIT- EĞRİ UYDUR
FIT TOLERANCE- TOLERANS UYDUR
FIT- UYDUR
FLOAT- YÜZER
FORMAT- FORMAT
GREEN- YEŞİL
GRID- İZGARA
HATCH- TARA
HEIGHT- YÜKSEKLİK
HIDE- GİZLE
IGNORE- GÖZARDI
INCLUDED- KAPSANAN
INSCRIBED- ÇEMBERE İÇTEN TEĞET
INSERT- YERLEŞTİR
INVISIBLE- GÖRÜNMEZ
ISOCIRCLE- İZOÇEMBER
ISOPLANE- İZOMETRİK DÜZLEM

EK-1 (Devam) AutoCAD çizim sözlüğü

JOIN- BİRLEŞTİR
JUSTIFICATION- HİZALAMA
JUSTIFY- YANAŞTIR
LAST- SON
LAYER- KATMAN
LEADER- ÖNCÜ
LEFT- SOL
LENGTH OF CHORD- KİRİŞ UZUNLUĞU
LENGTH- UZUNLUK
LENGTHEN- UZATMA
LIMITS- SINIRLAR
LINE- ÇİZGİ
LOAD- YÜKLE
LOOP- DÖNGÜ
LTSCALE- ÇİZGİTİPİ ÖLÇEĞİ
LTYPE- ÇİZGİ TİPİ
MAGENTA- MOR
MEASURE- ÖLÇ
MESH- GÖZ
METHOD- METOD
MIDDLE- ORTA
MIDDLECENTER- ORTAMERKEZ
MIDDLELEFT- SOLORTA
MIRROR3D- 3BOYUT AYNALA
MLINE- ÇOKLU ÇİZGİ
MTEXT- PARAGRAF YAZI
MULTIPLE- ÇOKLU
NEXT- SONRAKİ
NO EXTEND- UZATMA YOK
NO TRIM- BUDAMA YOK
NONE- HİÇBİRİ
NORMAL- NORMAL
OBJECT- NESNE
OFF- KAPALI
OFF- KAPAT
OFFSET- KOŞUTLA, PARALELİNİ AL
ON- AÇIK
OPEN- AÇIK
OPTIONS- SEÇENEKLER
ORIGIN- MERKEZ
ORTHO- DİKEY
OUTER- EN DIŞ
PAN- KAYDIR
PARAMETER- PARAMETRE
PEDIT- BİLEŞİK-ÇİZGİ DÜZENLEME
PERCENT- YÜZDE
PLAN- PLAN
POINTS- NOKTALAR
POLAR- KUTUPSAL
POLYGON- ÇOKGEN
POLYLINE- BİLEŞİK ÇİZGİ

EK-1 (Devam) AutoCAD çizim sözlüğü

POSITION- KONUM
PRESET- ÖNAYAR
PREVIOUS- ÖNCEKİ
PROJECT- YANSIT
PROPERTIES- ÖZELLİKLER
PYRAMID- PİRAMİT
QTEXT- ÇABUK YAZI
RADIUS- YARIÇAP
REALTIME- GERÇEK ZAMANLI
RECTANGULAR- DİKDÖRTGENEL
RED- KIRMIZI
REFERENCE- REFERANS
RESTORE- YENİDEN YÜKLE
REVOLVE- ÇEVİR
RIGHT- SAĞ
ROTATION- DÖNÜŞ
SAVE- SAKLA
SCALE- ÖLÇEK
SDF- BOŞLUK AYRAÇLI FORMAT
SELECT- SEÇ
SHOW- GÖSTER
SNAP- KENETLE, YAKALA
SPHERE- KÜRE
SPLINE- SPLINE
STRAIGHT- DÜZ
STYLE- STİL
SUBTRACT- ÇIKAR
TARGET- HEDEF
TEXT- YAZI
THICKNESS- KALINLIK
THROUGH- GEÇİŞ NOKTASI
TOGGLE- GEÇİŞ
TOLERANCE- TOLERANS
TOOLBAR- ARAÇ ÇUBUĞU
TOP- ÜST
TOPCENTER- ÜSTMERKEZ
TOPLLEFT- SOLÜST
TOPRIGHT- SAĞÜST
TORUS- SİMİT
TOTAL- TOPLAM
TRIM- BUDAMA
TTR- TEĞET TEĞET YARIÇAP
TWIST- ÇEVİR
U- KULLANICI
UCS- KULLANICI KOORDİNAT SİSTEMİ, KKS
UNDO- GERİAL
UNLOAD- ÇIKAR
VALUE- DEĞER
VERIFY- DOĞRULA
VIEW- GÖRÜNÜM
WEDGE- TAKOZ

EK-1 (Devam) AutoCAD çizim sözlüğü

WHITE- BEYAZ
WIDTH- GENİŞLİK
WINDOW- PENCERE
WORLD- YER
XYZ- XYZ
YELLOW- SARI
ZAXIS- Z EKSENİ
ZERO- SIFIR
ZOOM- ZUM

EK-2 Dwg formatı kodları örneği

```

typedef word word :=
byte : b1, byte : b2
return (word)f | ((word)s << 8).
typedef longword longp :=
byte : b1, byte : b2, byte : b3, byte : b4
return (longword)b1 | ((longword)b2 << 8)
| ((longword)b3 << 16) | ((longword)b4 << 24).
typedef longword longword :=
byte : b1, byte : b2, byte : b3, byte : b4
return (longword)b1 | ((longword)b2 << 8)
| ((longword)b3 << 16) | ((longword)b4 << 24).
Definition of the whole file
root dwg_file :=
[begin : end](
char[12] : version,
byte, word, word, word, byte,
longp : p_entities, longp : p_entend,
longp : p_blocksec, byte[4], longp : p_bsend, byte[4],
tablepos : block_table,
tablepos : layer_table,
tablepos : style_table,
tablepos : ltype_table,
tablepos : view_table,
header, [cur : 0x3EF]byte*,
tablepos : ucs_table, [cur : 0x500]byte*,
tablepos : vport_table, byte[8],
tablepos : appid_table, byte[6],
tablepos : dimstyle_table, [cur : 0x69F]byte*,
tablepos : p13_table, bytes[38],
[p_entities : p_entend]entities : ents, byte[19],

```

EK-2 (Devam) Dwg formatı kodları örneği

```
[block_table.start : ]blocks : block_table,
[layer_table.start : ]layers : layer_table,
[style_table.start : ]styles : style_table,
[ltype_table.start : ]ltypes : ltype_table,
[view_table.start : ]table : view_table,
[ucs_table.start : ]table : ucs_table,
[vport_table.start : vport_table.end]table : vport_table,
[appid_table.start : ]appids : appid_table,
[dimstyle_table.start : ]table : dimstyle_table,
[p13_table.start : ]table : p13_table,
[p_blocksec : p_bsend]entities : blocks, bytes[36],
longp = p_entities, longp = p_entend,
longp = blocksec, longp = bsend,
bytes[12],
bytes[6],
longp = block_table.start, bytes[6],
longp = layer_table.start, bytes[6],
longp = style_table.start, bytes[6],
longp = ltype_table.start, bytes[6],
longp = view_table.start, bytes[6],
longp = ucs_table.start, bytes[6],
longp = vport_table.start, bytes[6],
longp = appid_table.start, bytes[6],
longp = dimstyle_table.start, bytes[6],
longp = p13_table.start, bytes[6],
ongp bytes*, ).
A table position
tablepos :=
word : size,
long : nr,
```

EK-2 (Devam) Dwg formatı kodları örneği

```

long : start,
The header
header :=
word,
point(TRUE) : inbase,
point(TRUE) : extmin,
point(TRUE) : extmax,
point(FALSE) : limmin,
point(FALSE) : limmax,
double[4],
byte[2],
double[2],
byte[56],
double[3],
byte[18],
double .
The block table
blocks :=
( [size]( byte : flag, char[32] : name,
word : used,
byte, word, byte, word,
check_2  )
)[nr] : blocks_info,
check_32.
check_2 := byte[2].
check_32 := byte[32].
The layer table
layers :=
( [size](

```

EK-2 (Devam) Dwg formatı kodları örneği

```

byte : flag,
char[32] : name,
word : used,
word : color,
word : style,
check_2
)
)[nr] : layer_info,
check_32.

```

The style table

```

styles :=
( [size](
byte : flag,
char[32] : name, word, double[3],
byte, double, char[128], check_2)
)[nr] : style_info,
check_32.

```

The line-type table

```

ltypes :=
( [size](
byte : flag,
char[32] : name,
word, char[48], byte,
byte, double[13],
check_2 )
)[nr] : ltype_info,
check_32 .

```

The application identifier table

```

appids :=
( [size](

```

EK-2 (Devam) Dwg formatı kodları örneği

```
byte : flag,
char[32] : name,
word,
check_2 ) ) [nr] : appid_info,
check_32 .
```

The other tables

```
table :=
( [size](
byte : flag,
[size - 3] byte*,
check_2 ) ) [nr],
check_32 .
```

The entities

(Experimental)

```
entities := ( byte : kind,
byte : flag,
word : length,
[length - 4](
word : layer,
word : opts,
if (flag & 1) then byte : color else color = 0 fi,
if (flag & 0x40) then byte : extra else extra = 0 fi,
if (extra & 2) then xdata fi,
if (flag & 2) then word : type fi,
if (flag & 4 && kind > 2 && kind != 22) then double : z fi,
if (flag & 8) then double : th fi,
if (flag & 0x20) then handle fi,
if (extra & 4) then word : paper fi,
switch (kind)
case 1: /* LINE */
```

EK-2 (Devam) Dwg formatı kodları örneği

```

point(!(flag & 4)) : l10,
point(!(flag & 4)) : l11,
if (opts & 1) then point(TRUE) : l210 fi,
if (opts & 2) then double : l38 fi,
case 2: /* POINT */
point(!(flag & 4)) : l10,
if (opts & 1) then point(TRUE) : l210 fi,
if (opts & 2) then double : l38 fi,
case 3: /* CIRCLE */
point(FALSE) : l10,
double : l40,
if (opts & 1) then point(TRUE) : l210 fi,
if (opts & 2) then double : l38 fi,
case 4: /* SHAPE */
point(FALSE) : l10,
word : l2,
if (opts & 1) then point(TRUE) : l210 fi,
if (opts & 2) then double : l38 fi,
case 7: /* TEXT */
point(FALSE) : l10,
double : l40,
string : l1,
if (opts & 1) then double : l50 fi,
if (opts & 2) then double : l41 fi,
if (opts & 4) then double : l51 fi,      /*?*/
if (opts & 8) then byte : l7 fi,
if (opts & 0x10) then byte : l71 fi,
if (opts & 0x20) then byte : l72 fi,
if (opts & 0x40) then point(FALSE) : l11 fi,
if (opts & 0x100) then byte : l73 fi,

```

EK-2 (Devam) Dwg formatı kodları örneği

```

case 8: /* ARC */
point(FALSE) : l10,
double : l40,
double : l50,
double : l51,
if (opts & 1) then point(TRUE) : l210 fi,
if (opts & 2) then double : l38 fi,
case 9: /* TRACE */
point(FALSE) : l10,
point(FALSE) : l11,
point(FALSE) : l12,
point(FALSE) : l13,
if (opts & 1) then point(TRUE) : l210 fi,
if (opts & 2) then double : l38 fi,
case 11: /* SOLID */
point(FALSE) : l11,
point(FALSE) : l12,
point(FALSE) : l13,
point(FALSE) : l14,
if (opts & 1) then point(TRUE) : l210 fi,
if (opts & 2) then double : l38 fi
case 12: /* BLOCK */
point(FALSE) : l10, /*?*/ string : l1, /* if (opts & 1) then ? */
if (opts & 2) then string : l3 fi
case 13: /* ENDBLK */
case 14: /* INSERT */
word : l1,
point(FALSE) : l10,
if (opts & 1) then double : l41 fi,
if (opts & 2) then double : l42 fi,

```

EK-2 (Devam) Dwg formatı kodları örneği

```

if (opts & 4) then double : l43 fi,
if (opts & 8) then double : l50 fi,
if (opts & 0x10) then byte : l70 fi,          /*?*/
if (opts & 0x20) then byte : l71 fi,          /*?*/
if (opts & 0x40) then double : l44 fi,        /*?*/
if (opts & 0x80) then double : l45 fi        /*?*/
case 15: /* ATTDEF */
point(FALSE) : l10,
double : l40,
string : l1,
string : l3,
string : l2,
byte : l70,
if (opts & 1) then byte : l73 fi,             /*?*/
if (opts & 2) then double : l50 fi,           /*?*/
if (opts & 4) then double : l41 fi,
if (opts & 8) then double : l42 fi,
if (opts & 0x10) then byte : l7 fi,
if (opts & 0x20) then byte : l71 fi,
if (opts & 0x40) then byte : l72 fi,
if (opts & 0x80) then point(FALSE) : l11 fi, /*?*/
if (opts & 0x100) then point(TRUE) : l210 fi,
if (opts & 0x200) then double : l38 fi       /*?*/
case 16: /* ATTRIB */
point(FALSE) : l10,
double : l40,
string : l1, string : l2,
byte : l70,
if (opts & 1) then byte : l73 fi,             /*?*/
if (opts & 2) then double : l50 fi,           /*?*/

```

EK-2 (Devam) Dwg formatı kodları örneği

```

if (opts & 4) then double : l41 fi,
if (opts & 8) then double : l42 fi,
if (opts & 0x10) then byte : l7 fi,
if (opts & 0x20) then byte : l71 fi,
if (opts & 0x40) then byte : l72 fi,
if (opts & 0x80) then point(FALSE) : l11 fi, /*?*/
if (opts & 0x100) then point(TRUE) : l210 fi,
if (opts & 0x200) then double : l38 fi /*?*/
case 17: /* S/BEND */long
case 19: /* PLINE */
if (opts & 1) then byte : l70 fi,
if (opts & 2) then double : l40 fi, /*?*/
if (opts & 4) then byte : l71 fi, /*?*/
if (opts & 8) then byte : l72 fi, /*?*/
if (opts & 0x10) then byte : l73 fi, /*?*/
if (opts & 0x20) then byte : l74 fi, /*?*/
if (opts & 0x40) then byte : l75 fi /*?*/
case 20: /* VERTEX */
point(FALSE) : l10,
if (opts & 1) then double : l40 fi, /*?*/
if (opts & 2) then double : l41 fi, /*?*/
if (opts & 4) then byte : l70 fi, /*?*/
if (opts & 8) then double : l50 fi /*?*/
case 22: /* 3DFACE */
point(!(flag & 4)) : l10,
point(!(flag & 4)) : l11,
point(!(flag & 4)) : l12,
point(!(flag & 4)) : l13
case 23: /* DIM */
word : l1,

```

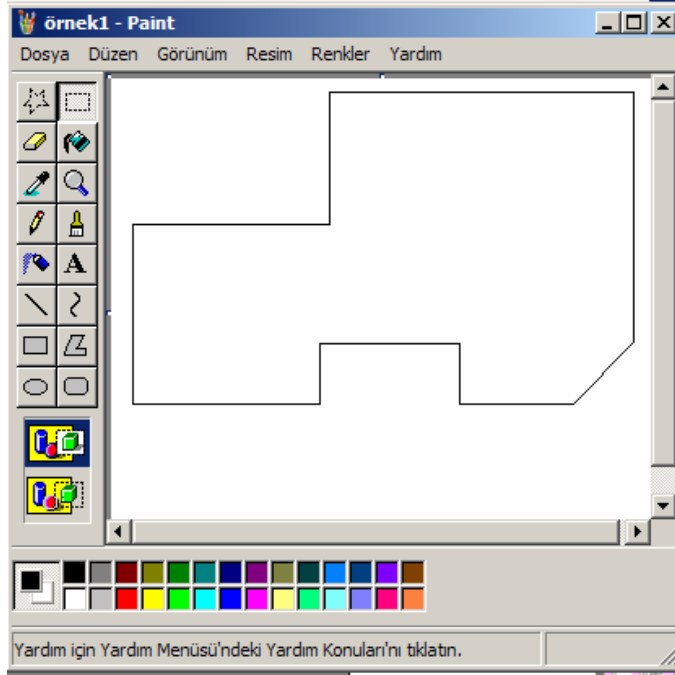
EK-2 (Devam) Dwg formatı kodları örneği

```

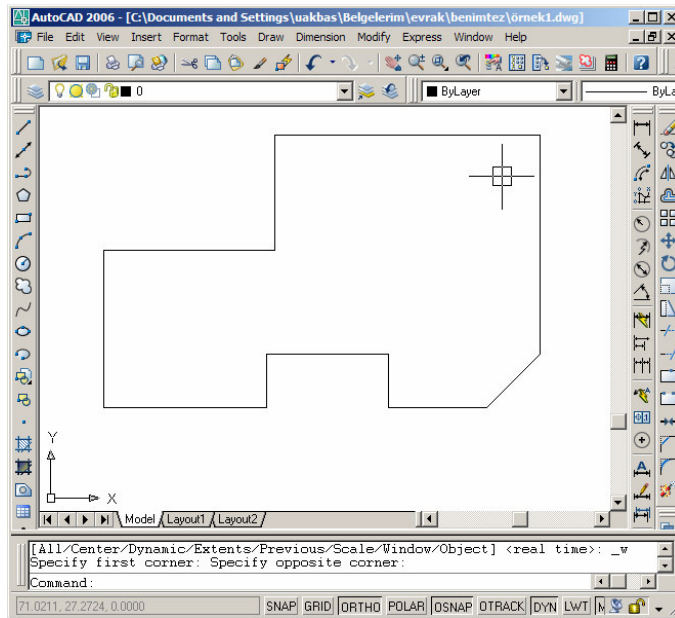
point(TRUE) : l10,
point(FALSE) : l11, /*?*/
if (opts & 2) then byte : l70 fi,
if (opts & 1) then point(TRUE) : l12 fi, /*?*/
if (opts & 4) then string : l1 fi,
if (opts & 8) then point(TRUE) : l13 fi,
if (opts & 0x10) then point(TRUE) : l14 fi,
if (opts & 0x20) then point(TRUE) : l15 fi,
if (opts & 0x40) then point(TRUE) : l16 fi,
if (opts & 0x80) then double : l40 fi,
if (opts & 0x100) then double : l50 fi,
if (opts & 0x200) then double : l51 fi,
if (opts & 0x400) then double : l52 fi,
if (opts & 0x800) then double : l53 fi
case 24: /* VPORT */
point(TRUE) : l10,
double : l40,
double : l41,
word : l68
endswitch
check_2 ) ) * : entities.

```

EK-3 Örnek resimler

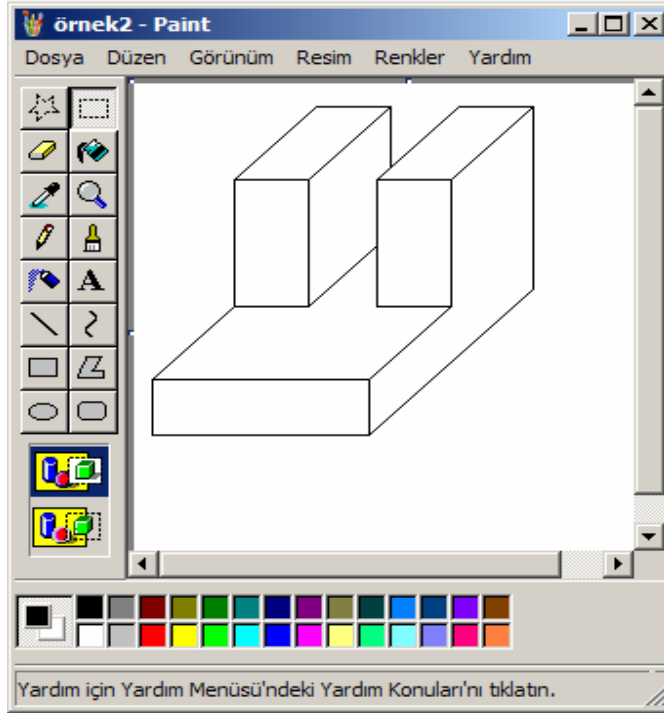


Şekil 3.1. Örnek-1 Bmp Şekil

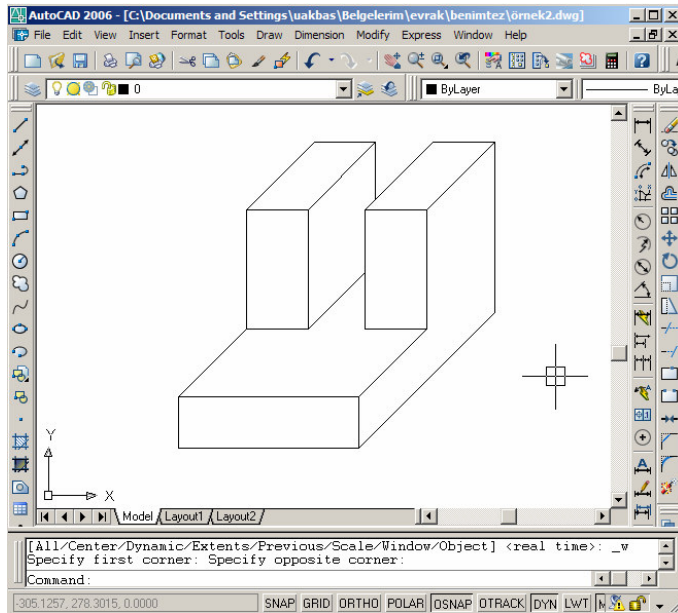


Şekil 3.2. Örnek-1 Bmp resmin AutoCAD programında gösterilmesi

EK-3 (Devam) Örnek resimler

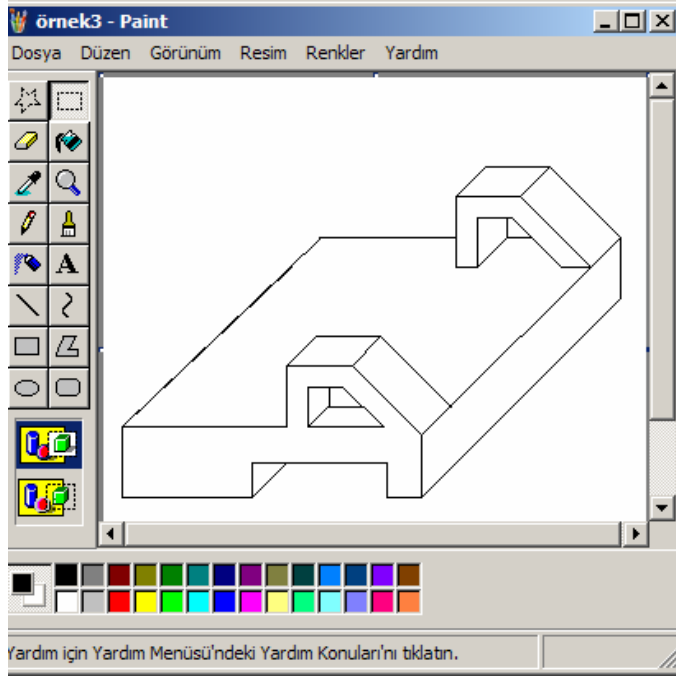


Şekil 3.3. Örnek-2 Bmp resim

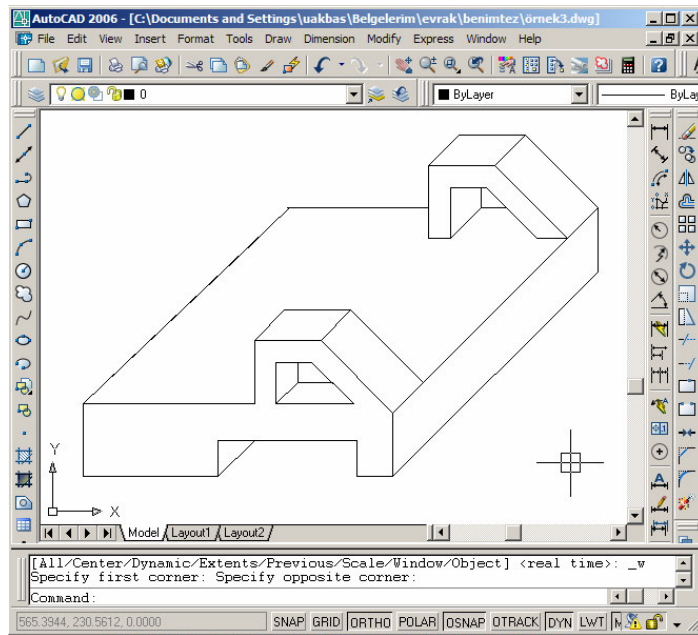


Şekil 3.4. Örnek-2 Bmp resmin AutoCAD programında gösterilmesi

EK-3 (Devam) Örnek resimler

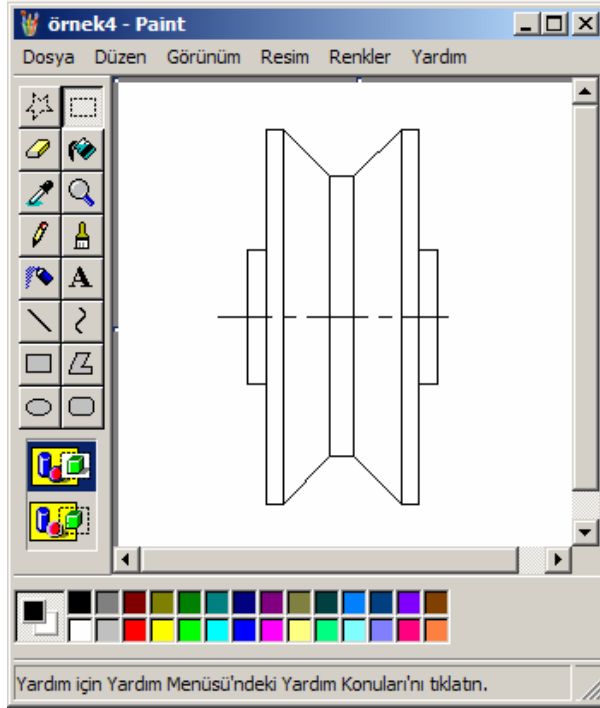


Şekil 3.5. Örnek-3 Bmp resim

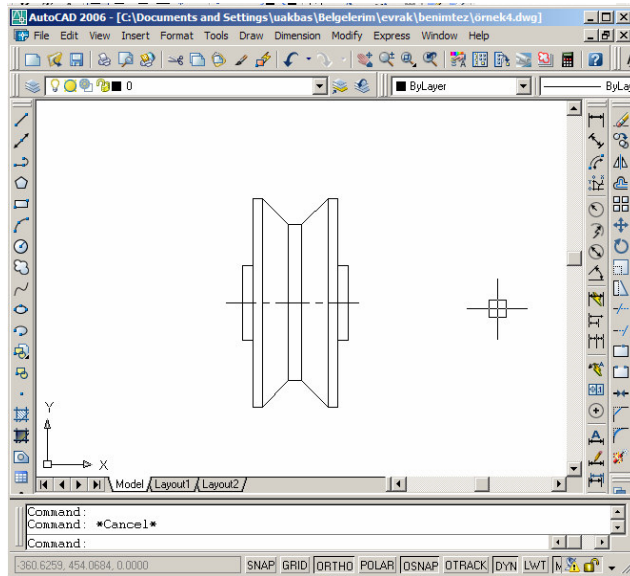


Şekil 3.6. Örnek-3 Bmp resmin AutoCAD programında gösterilmesi

EK-3 (Devam) Örnek resimler

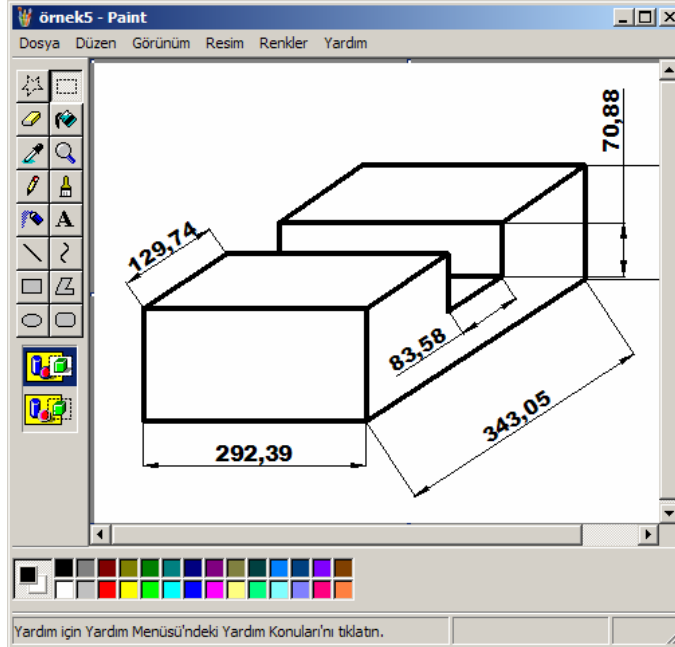


Şekil 3.7. Örnek-4 Bmp resim

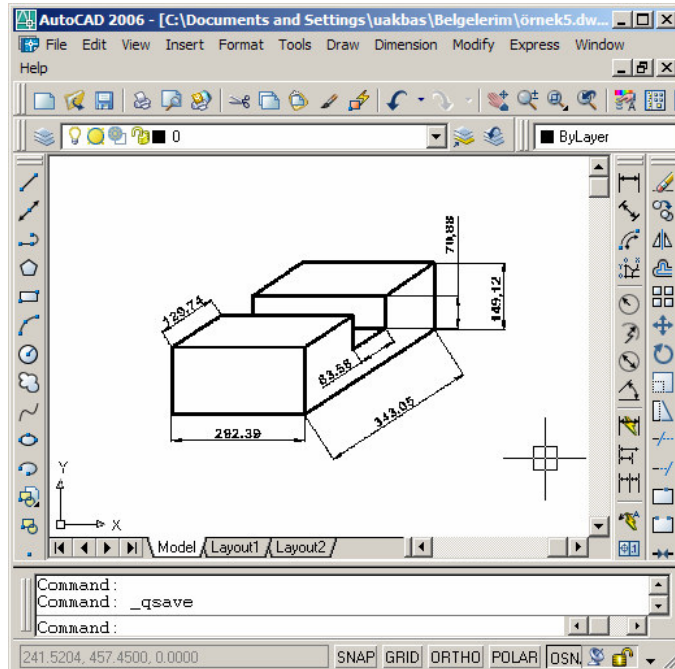


Şekil 3.8. Örnek-4 Bmp resmin AutoCAD programında gösterilmesi

EK-3 (Devam) Örnek resimler

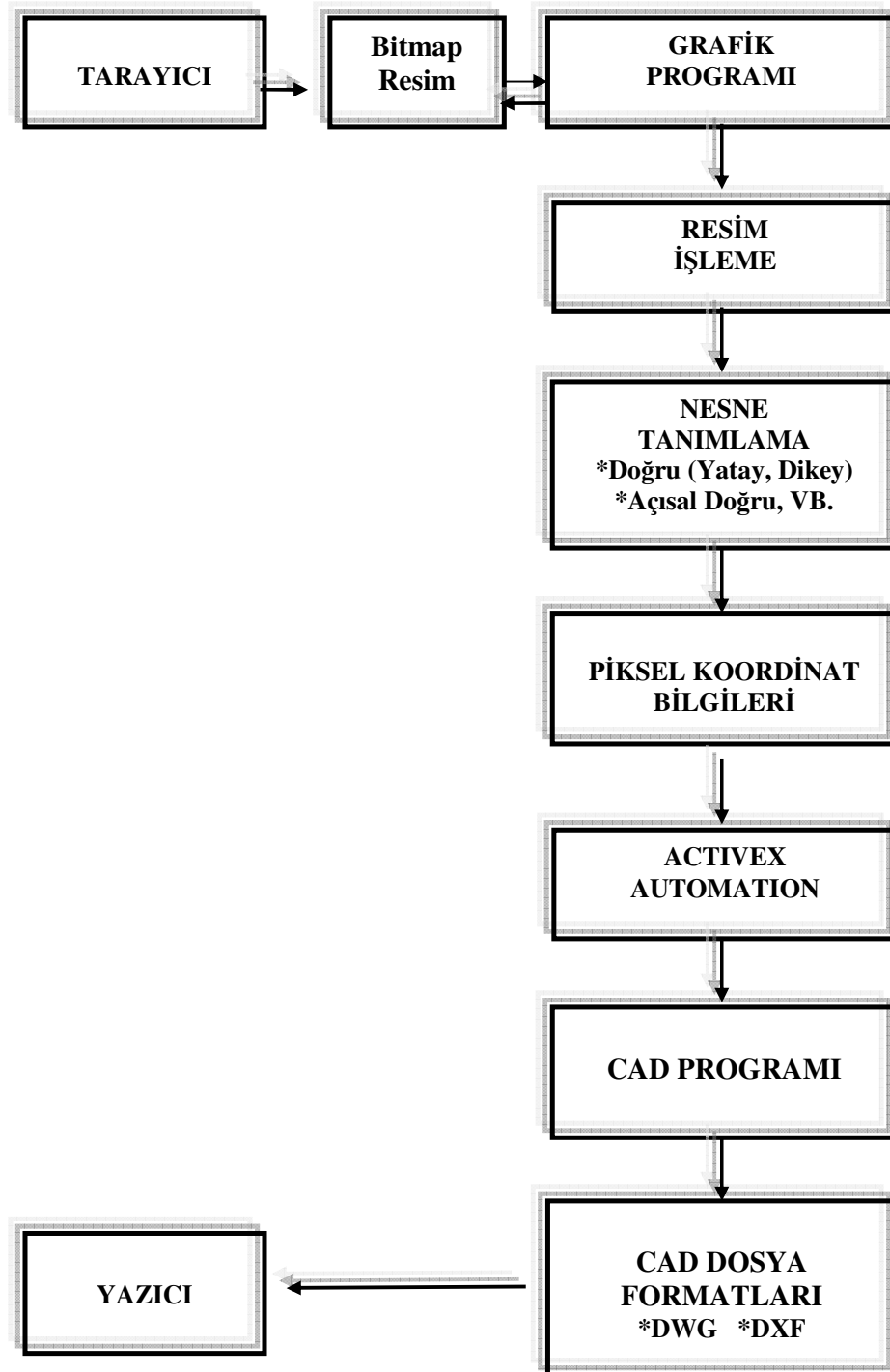


Şekil 3.9. Örnek-5 Bmp resim



Şekil 3.10. Örnek-5 Bmp resmin AutoCAD programında gösterilmesi

EK-4 Akış diyagramı



ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : AKBAŞ, Ümit
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 26.01.1975 Seydişehir
 Medeni hali : Evli
 Telefon : 0 (224) 5733985
 Faks : 0 (224) 5735226
 e-mail : umit_akbasii@yahoo.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üni./ Tek. Eğt. Fak./ Makine Öğr.	1998
Lise	Seydişehir Endüstri Meslek Lisesi	1992

İş Deneyimi

Yıl	Yer	Görev
1999–2006	Orhangazi And. Tek. ve End. Mes. Lisesi	Teknik Öğretmen

Yabancı Dil

İngilizce

Hobiler

Bilgisayar teknolojileri, Yürüyüş, Seyehat