

166613

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**BİLGİSAYAR DESTEKLİ İMALAT İŞLEMLERİ İÇİN GÖRÜNTÜ
ALGILAMALI PARÇA PROGRAMLAMA YAZILIMI VE UYGULAMASI**

YÜKSEK LİSANS TEZİ

Bilgisayar ve Enformatik Müh. Cengiz BALTA

Anabilim Dalı: Mekatronik Mühendisliği

Danışman: Yrd. Doç. Dr. Cüneyt OYSU

MAYIS 2006

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**BİLGİSAYAR DESTEKLİ İMALAT İŞLEMLERİ İÇİN GÖRÜNTÜ
ALGILAMALI PARÇA PROGRAMLAMA YAZILIMI VE UYGULAMASI**

YÜKSEK LİSANS TEZİ

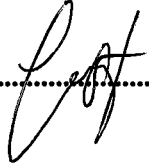
Bilgisayar ve Enformatik Müh. Cengiz BALTA

Tezin Enstitüye Verildiği Tarih: 26 Mayıs 2006

Tezin Savunulduğu Tarih: 12 Temmuz 2006


Tez Danışmanı

Yrd.Doç.Dr. Cüneyt OYSU

(..........)

Üye

Doç.Dr. Emin Murat ESİN

(..........)

Üye

Yrd.Doç.Dr. Hasan OCAK

(..........)

BİLGİSAYAR DESTEKLİ İMALAT İŞLEMLERİ İÇİN GÖRÜNTÜ ALGILAMALI PARÇA PROGRAMLAMA YAZILIMI VE UYGULAMASI

Cengiz BALTA

Anahtar Kelimeler: Çizici, kartezyen robot, hareket kartı, servo sürücü, CAD, görüntü işleme, kenar bulma, vektörizasyon, vektör resim

Özet: Bu çalışmada, iki boyutlu parçaları görebilen ve bunlar için CAD-CAM parça programlama ve imalat ortamı fonksiyonlarını sağlayan genel amaçlı bir çizici-kesici kartezyen robot uygulaması ve bilgisayar yazılımı gerçekleştirilmiştir. Üç doğrusal eksenli oluşan, servo motorlu kartezyen robot mekanizması ve hareket kontrol kartı fonksiyonları, görüntü işleme, kenar bulma, vektörizasyon ve parçaların dahili CAD parça programlama ortamına aktarılması özellikleri eklenerek, tezgah üzerindeki iki boyutlu parçaları görebilen ve bunlar üzerinde parça programlama özelliklerini sağlayan, iki boyutlu parçaların birebir kopyasını görerek üretebilen, genel amaçlı bir çizici sistem üretilmiştir.

A PART MODELLING APPLICATION FOR COMPUTER AIDED DESIGN AND MANUFACTURING WITH IMAGE RECOGNITION

Cengiz BALTA

Key words: Plotter, cartesian robot, motion card, servo driver, CAD, image processing, edge detection, vectorization, vector image

Abstract: In this work, a general purpose plotter-cutter system that can recognize 2D mechanical parts using vision methods with a CAD-CAM part programming software is designed. A cartesian robot that contains three linear axis with servo motors and a motion control card is used in the project. Image processing, edge detection, vectorization of parts and CAD design capabilities added to the software that controls mechanical system. Finally, a plotter-cutter cartesian robot that can see 2D mechanical parts on it's working area with CAD design environment is developed.

ÖNSÖZ ve TEŞEKKÜR

Bu tez çalışmasında, imalat sektöründe yaygın olarak kullanılan iki boyutlu kesme düzeneklerine, kameradan alınan iki boyutlu parça resimlerinin CAD ortamına aktarılması ve ana parça üzerinde parça programlama fonksiyonları eklenmesi hedeflenmiştir.

Tez çalışması süresince malzeme alımları ve teknik konularda destek olan Yrd. Doç.Dr. Cüneyt OYSU'ya ve Mekatronik Mühendisliği bölümüne teşekkür ederim.



İÇİNDEKİLER

ÖZET.....	ii
ABSTRACT.....	iii
ÖNSÖZ VE TEŞEKKÜR.....	iv
İÇİNDEKİLER.....	v
SİMGELER DİZİNİ VE KISALTMALAR.....	viii
ŞEKİLLER DİZİNİ.....	ix
TABLOLAR DİZİNİ.....	xii
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. TEMEL GÖRÜNTÜ İŞLEME FONKSİYONLARI.....	4
2.1. Dijital Resim Tanımı, Piksel Kavramı.....	4
2.2. Aritmetik Görüntü İşleme Fonksiyonları.....	5
2.2.1. Toplama.....	5
2.2.2. Çıkarma.....	6
2.2.3. Çarpma.....	8
2.2.4. Bölme.....	9
2.2.5. Girişim.....	9
2.2.6. Olumsuzlama.....	11
2.3. Noktasal Görüntü İşleme Fonksiyonları.....	11
2.3.1. Histogram.....	11
2.3.2. Eşikleme.....	12
2.3.3. Kontrast yayma.....	13
2.3.4. Histogram dengeleme.....	14
2.4. Geometrik Görüntü İşleme Fonksiyonları.....	15
2.4.1. Boyut değişikliği.....	15
2.4.2. Döndürme.....	16
2.4.3. Aynalama, simetri.....	16

2.4.4. Öteleme...	17
2.5. Morfolojik Görüntü İşleme Fonksiyonları.....	18
2.5.1. Yayma...	18
2.5.2. Aşındırma.....	19
2.5.3. Açma.....	19
2.5.4. Kapama.....	20
BÖLÜM 3. KENAR BULMA ALGORİTMALARI, VEKTÖRİZASYON	21
3.1. Giriş.....	21
3.2. Bitmap Resimde Kenar Tipleri	22
3.3. Kenar Bulmada Kullanılan Aşamalar	22
3.3.1. Filtreleme aşaması.....	23
3.3.2. Türevleme aşaması	25
3.3.3. Eşikleme aşaması	29
3.3.4. Maksimum olmayanların bastırılması	30
3.3.5. Eşikleme ve süreklilik	33
3.4. Kenar Bulma Sonrası Gürültü Eleme Filtresi	33
3.5. Vektörizasyon Algoritması	34
BÖLÜM 4. UYGULAMA YAZILIMI.....	37
4.1. Kartezyen Robot Kontrol Arayüzü.....	37
4.2. Kamera Arayüzü.....	38
4.3. CAD-CAM Tasarım Arayüzü.....	39
4.4. CAD Nesneleri.....	40
4.4.1. Çizgi/Line.....	41
4.4.2. Çoklu çizgi/polyline	43
4.4.3. Daire.....	43
4.4.4. Yay parçaları.....	46
4.5 Nesnelerin Alt Nesnelere Parçalanması.....	48
BÖLÜM 5. KARTEZYEN ROBOT.....	49
5.1. Servo Sürücüler.....	51
5.2. SEC-AC 305 Servo Sürücüsü.....	52

5.3. Servo Sürücü Bağlantıları	53
5.3.1. Resolver bağlantısı	53
5.3.2. Analog-dijital çıkışlar	54
5.3.3. Artımlı enkoder girişi	56
5.3.4. Artımlı enkoder çıkışı.....	56
5.3.5. Motor güç bağlantısı.....	57
5.4. Diferansiyel Sinyal Aktarma ve Darbe Tipleri.....	58
5.5. Hareket Kartı ve Servo-Sürücü Bağlantıları	59
5.6. Eksen Yapısı.....	60
5.7. Mekanik Parametre Tanımları	60
5.8. Home Parametre Tanımları.....	62
5.9. Enkoder Parametre Tanımları.....	63
5.10. Koordinat Sistemi	63
5.11. Yörünge Planlama.....	64
5.12. Çizgi, Yay, Çember	64
5.13. Gelişmiş Yörünge Planlama: Blending.....	65
BÖLÜM 6. UYGULAMA ÖRNEKLERİ.....	66
6.1. Kenar Taraması.....	66
6.2. Ana Parçadan Harf Kesme.....	69
BÖLÜM 7. SONUÇ ve ÖNERİLER	71
KAYNAKLAR	73
ÖZGEÇMİŞ	132

SİMGELER DİZİNİ ve KISALTMALAR

Pitch	Vida adımı
Gear Ratio	Dişli vites oranı
Inc	Incremental / Artımlı
Abs	Absolute / Mutlak

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CNC	Computerize Numerical Control
NC	Numerical Control
RPM	Revolute per minute / Dakikadaki tur sayısı
OpenDWG	Open Design Alliance
A/B PHASE	A/B kanalı formunda darbe
CW/CCW	İleri – geri darbeler
PULSE/DIR	Darbe – yön haberleşmesi

ŞEKİLLER DİZİNİ

Şekil 2.1. Piksel kavramı.....	4
Şekil 2.2. Tamsayılardan oluşan resim	4
Şekil 2.3. Sabit sayı ekleyerek parlaklık ayarı yapılması.....	5
Şekil 2.4. Girdi resmi ve canny algoritması ile elde edilen kenar resmi.....	6
Şekil 2.5. Toplama fonksiyonu sonucu elde edilen resim.....	6
Şekil 2.6. Sabit sayı çıkararak parlaklık ayarı.....	7
Şekil 2.7. Resimde parçaların yerleri değiştiriliyor	7
Şekil 2.8. İki resmin farkı alınarak yer değişikliği tespiti.....	8
Şekil 2.9. Çarpma işlemi, kontrast azaltma ve artırma	8
Şekil 2.10. Orijinal resim ve kenar resmi.....	9
Şekil 2.11. Orijinal resim ve kenar resminin birleştirilmesi	10
Şekil 2.12. İki girdi resmi.....	10
Şekil 2.13. Blending fonksiyonu ile resimlerin birleştirilmesi	10
Şekil 2.14. Olumsuzlama ile negatif resim elde edilmesi	11
Şekil 2.15. Histogram.....	12
Şekil 2.16. Eşikleme.....	12
Şekil 2.17. Kontrast yayma öncesi resim.....	13
Şekil 2.18. Kontrast yayma sonucu.....	13
Şekil 2.19. Histogram dengeleme öncesi girdi resmi.....	14
Şekil 2.20. Histogram dengeleme sonucu	14
Şekil 2.21. Değer tekrarlama ile boyut değiştirme.....	15
Şekil 2.22. Boyut değişikliği uygulaması	15
Şekil 2.23. Girdi resminin döndürülmesi	16
Şekil 2.24. Girdi resminin aynalanması	17
Şekil 2.25. Öteleme işlemi.....	17
Şekil 2.26. Yayma işlemi.....	18
Şekil 2.27. Aşındırma işlemi.....	19
Şekil 2.28. Açma işlemi sonucu.....	20

Şekil 2.29. Kapama işlemi sonucu.....	20
Şekil 3.1. Kenar tipleri: adım, rampa, darbe, çatı	22
Şekil 3.2. Ortalama-mean filtre.....	24
Şekil 3.3. Gauss fonksiyonu ve tamsayı yaklaşımı	24
Şekil 3.4. Bir boyutta türevleme.....	26
Şekil 3.5. Yatay ve düşey Sobel türev maskeleri	26
Şekil 3.6. Yatay ve düşey Prewit türev maskeleri.....	27
Şekil 3.7. Uygulamada kamera ile alınan görüntü	28
Şekil 3.8. Yatayda türevleme, Sobel Gx maskesi	28
Şekil 3.9. Düşeyde türevleme, Sobel Gy maskesi.....	29
Şekil 3.10. Gradient genlik resmi.....	29
Şekil 3.11. Gradient açısının dört bölgeye eşlenmesi	30
Şekil 3.12. Kenara dik doğrultuda maksimum tespiti	31
Şekil 3.13. Uygulamadan alınan gradient genlik resmi	32
Şekil 3.14. Maksimum olmayanların bastırılması sonucu kenar resmi	32
Şekil 3.15. Gürültü eleme filtresi için çekirdek yapılar	34
Şekil 3.16. Gradient açısı değeri izleme metodu ile vektörizasyon.....	35
Şekil 3.17. Vektörizasyon sonrası elde edilen CAD resmi	36
Şekil 4.1. Kartezyen robot kontrol arayüzü.....	37
Şekil 4.2. Kamera izleme arayüzü.....	38
Şekil 4.3. CAD tasarım ortamı.....	40
Şekil 4.4. Çizgi nesnesi.....	41
Şekil 4.5. Polyline nesnesi.....	43
Şekil 4.6. Dairenin merkez noktası ve yarıçapı.....	44
Şekil 4.7. Merkez/yarıçap, iki nokta, üç nokta metodları	44
Şekil 4.8. Yay parçasının CAD ortamındaki unsurları	46
Şekil 4.9. Merkez-yarıçap-başlangıç-bitiş noktası ile yay çizimi	46
Şekil 4.10. Üç nokta ile yay çizimi	46
Şekil 5.1. XYZ kartezyen robot yakın görünüm	49
Şekil 5.2. XYZ kartezyen robot ve kontrol paneli	50
Şekil 5.3. Kontrol paneli yakın görünüm	50
Şekil 5.4. Servo sistemin parçaları	51
Şekil 5.5. Servo sürücü, üstten görünüm	52

Şekil 5.6. Servo sürücü, alttan görünüm	53
Şekil 5.7. Servo sürücü resolver bağlantıları	53
Şekil 5.8. Analog giriş-çıkış soketi	54
Şekil 5.9. Servo motor güç bağlantısı	57
Şekil 5.10. Diferansiyel sinyal aktarma metodu	58
Şekil 5.11. Desteklenen darbe giriş-çıkış formatları.....	59
Şekil 5.12. Hareket kartı-servo sürücü bağlantıları, bir eksen için.....	59
Şekil-5.13. Hareket kartının eksen yapısı.....	60
Şekil 5.14. Mekanik parametre ayarları	60
Şekil 5.15. İvme seçenekleri.....	64
Şekil 5.16. Blending modunda hız grafiği	65
Şekil 5.17. Blending modunda geçişler.....	65
Şekil 6.1. Kameradan görüntü yakalama	66
Şekil 6.2. Kenar resmi eldesi.....	67
Şekil 6.3. Vektör resmi, CAD-CAM ortamı	68
Şekil 6.4. Örnek kenar taraması..	68
Şekil 6.5. Ham parçanın kameradan okunması	69
Şekil 6.6. Harflerin ana parça üzerine yerleştirilmesi	70

TABLÖLAR DİZİNİ

Tablo 5.1. Servo sürücü resolver bağlantı tablosu	36
Tablo 5.2. Servo sürücü analog giriş/çıkış soketi bağlantıları	38
Tablo 5.3. Servo sürücü artımlı enkoder girişı.....	46
Tablo 5.4. Servo sürücü artımlı enkoder çıkışı	48
Tablo 5.5. Servo sürücü motor güç bağlantısı.....	49



BÖLÜM 1. GİRİŞ

Bilgisayar sistemlerinin imalatta, katı model parçalar oluşturma, tanımlama, analiz ve tasarımın optimizasyonu gibi işlerde kullanılması CAD (Computer Aided Design) olarak adlandırılır. Bu sistemler genel olarak yazılım ve donanım kısımlarından oluşur. Yazılım kısmı, parçaların gerilme-şekil değişimi analizinin yapılabildiği programlar, mekanizmaların dinamik cevapları, ısı transferi hesapları ve NC parça programlama gibi modülleri kapsamaktadır. CAM (Computer Aided Manufacturing) olarak isimlendirilen süreç ise, bilgisayar sistemlerinin planlama, yönetme ve bir imalat işleminin kontrolünün doğrudan ya da dolaylı olarak kullanılmasıdır.[11]

CAD/CAM sistemleri imalatta, tasarım, analiz, süreç planlama, parça programlama, program doğrulama, parça işleme, ve muayene gibi fonksiyonları etkin ve doğru bir şekilde yerine getirebilmektedir.

Üç doğrusal eksenenden oluşan kartezyen yapıdaki robotlar (gantry) endüstriyel uygulamalarda ve imalat amaçlı olarak yaygın olarak kullanılmaktadır. Kartezyen robotların paketleme, lojistik, freze, lazer, plazma, oksijen asetilen kesme makinaları, su jeti ile kesme, elektro erime gibi uygulama alanları bulunur.

Metal işleme, tekstil, dericilik ve benzeri imalat endüstrilerinde iki boyutlu kesme problemleri oldukça geniş bir yer tutmaktadır. Bu sistemlerde, doğrusal eksenler ve servo motorlar yardımı ile CAD ortamında çizilen iki boyutlu parçaların seri üretimleri gerçekleştirilir.

Bu projede, imalat endüstrisinde yaygın olarak kullanılan iki boyutlu kesme makinalarına kamera desteği ile parçaları görme ve CAD-CAM tasarım-imalat ortamına ölçekli olarak aktarabilme özelliği kazandırılması hedeflenmiştir. İmalata yönelik CAM uygulamalarında, işlenmemiş, ham haldeki ana parçanın, boyutları ve konumunun üretimi gerçekleştirecek olan makinarya hassas bir şekilde tanıtılması gerekir. Daha sonra ana parçadan imal edilecek olan parçalar CAM uygulamasında tasarlanır. Bilgisayar ortamında işleme hareketlerinin görsel olarak benzetiminden sonra kontrol ünitesine yüklenen hareket komutları makinada otomatik olarak işlenir.

Projenin uygulama aşamalarında ilk olarak, doğrusal eksenler, servo motorlar, servo sürücüler ve hareket kartından oluşan kartezyen robotun kurulması ve elektrik bağlantılarının yapılması yer almıştır. Daha sonra, robotun çalışma alanından ana parçanın iki boyutlu görüntüsünü almak üzere kameranın yerleştirilmesi ve kameradan alınan görüntünün vektör resmine dönüştürülmesi aşaması yer alır. Kameradan elde edilen ham haldeki resimden, Canny [4] kenar bulma algoritması kullanılarak, tezgah üzerindeki parçaların kenar resminin elde edilmesinden sonra, geliştirilen kenar resmi pikselleri üzerinde gradient – iki boyutlu türev-eğimi izleme metodu ile tezgah üzerindeki parçaların vektör resmi elde edilir. Vektör resmi eldesi öncesinde Stagg[6] tarafından kullanılmış olan bazı kenar resminden gürültü eleme filtreleri kenar resmine uygulanmış, vektörlerin başlangıç ve bitiş noktalarındaki piksel çakışmalarının ve kaymalarının engellenmesi için Parker[8] tarafından geliştirilen vektörizasyon algoritmasının bazı noktalarından yararlanılmıştır. Ayrıca Susan[7] kenar bulma algoritmasında kullanılan inceltme/thinning metodlarındaki fikirlerden istifade ile, elde edilen vektör resimlerdeki bazı parazitlerin giderilmesi temin edilmiştir.

Ana parçanın vektör resmi elde edildikten sonra, bu vektör resmi, kullanıcıya tasarım ve imalat arayüzü sağlayan CAD-CAM windows uygulamasına aktarılır. Uygulama programı, ana parçanın CAD vektör resmi üzerinde kullanıcının imal edilecek parçaları çizmesini ve kullanıcı tarafından çizilen parçaların, kartezyen robota hareket komutları halinde göndererek imal edilmesini sağlar. Uygulama programı, CAD ortamında parçaların tasarımı için kullanılan çizgi, yay parçası,

ember ve benzeri CAD nesnelerini, tasarım ekranından tarayıp, kartezyen robota hareket komutları Őeklinde aktarır.

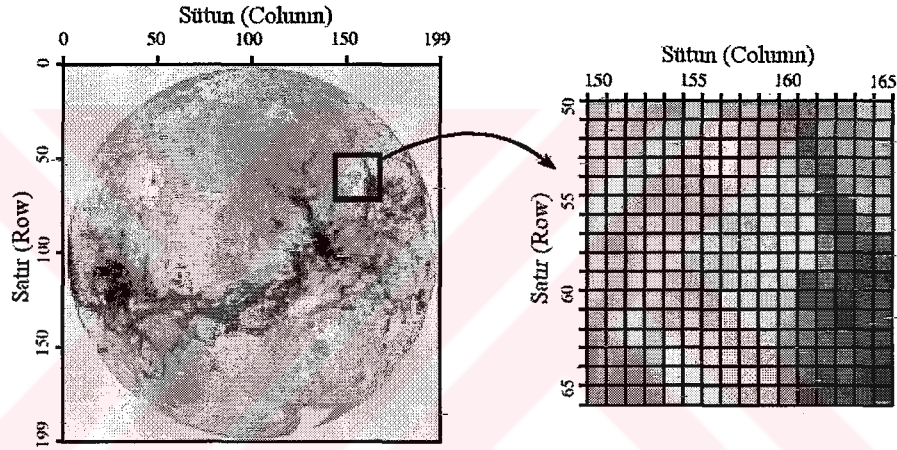
Projede gerekleŐtirilen aŐamalar Őu Őekilde zetlenebilir:

- XYZ kartezyen robot ve servo motorların kurulması
- Servo sŐrŐcŐler ve motor baŐlantılarının yapılması
- Hareket kartı ve servo sŐrŐcŐ haberleŐmelerinin temin edilmesi
- GrŐntŐ iŐleme, kenar bulma, vektörizasyon algoritmalarının geliŐtirilmesi
- Elde edilen vektör resminin CAD-CAM tasarım-imalat ortamına aktarılması
- GeliŐtirilen iki boyutlu CAD-CAM arayŐzŐ ile kullanıciya para programlama imkanı saŐlanması
- Kullanıcı tarafından ana para Őzerinde izilen iki boyutlu paraların tasarım ortamından XYZ kartezyen robota gnderilmesi

BÖLÜM 2. TEMEL GÖRÜNTÜ İŞLEME FONKSİYONLARI

2.1. Dijital Resim Tanımı, Piksel Kavramı

Dijital resimler, bilgisayar ortamında rakamlardan oluşan iki boyutlu diziler şeklinde saklanır. Renk içermeyen gri tonlu resimler için bir adet iki boyutlu tamsayı dizisi kullanılırken, renkli resimler için, kırmızı, yeşil ve mavi kanalları için kullanılmak üzere üç adet iki boyutlu dizi yapısı gereklidir.



Şekil 2.1 Piksel Kavramı

	150	155	160	165												
50	183	183	181	184	177	200	200	189	159	135	94	105	160	174	191	196
	186	195	190	195	191	205	216	206	174	153	112	80	134	157	174	196
	194	196	198	201	206	209	215	216	199	175	140	77	106	142	170	186
	184	212	200	204	201	202	214	214	214	205	173	102	84	120	134	150
	202	215	203	179	163	165	199	207	202	208	197	129	73	112	131	146
55	203	208	166	159	160	168	166	137	174	211	204	158	60	79	127	143
	174	149	143	151	156	148	146	123	118	203	208	162	81	58	101	125
	143	137	147	153	150	146	121	133	157	184	203	164	94	56	66	80
	164	165	159	179	188	159	126	134	150	199	174	119	100	41	41	58
	173	187	193	181	167	151	162	182	192	175	129	60	88	47	37	50
60	172	184	179	153	158	172	163	207	205	188	127	63	56	43	42	55
	156	191	196	159	167	195	178	203	214	201	143	101	69	38	44	52
	154	163	175	165	207	211	197	201	201	199	138	79	76	67	51	53
	144	150	143	162	215	212	211	209	197	193	153	71	69	77	63	53
	140	151	150	185	215	214	210	210	211	209	135	80	45	69	66	60
65	133	143	151	179	213	216	214	191	201	205	138	61	59	61	77	63

Şekil 2.2 Tamsayılardan Oluşan Resim

İki boyutlu tamsayı dizilerinden oluşan dijital resim üzerinde, piksel değerleri üzerinde çeşitli işlemler yapılarak bilgi elde edilmeye çalışılır. Piksel değerlerine dayanan bu metodlar, konvansiyonel görüntü işleme olarak isimlendirilir.

2.2. Aritmetik Görüntü İşleme Fonksiyonları

Bu kategoriye dahil görüntü işleme fonksiyonlarında, iki resmin karşılıklı piksel değerleri basit matematik fonksiyonlara tabi tutularak yeni bir resim elde edilir.

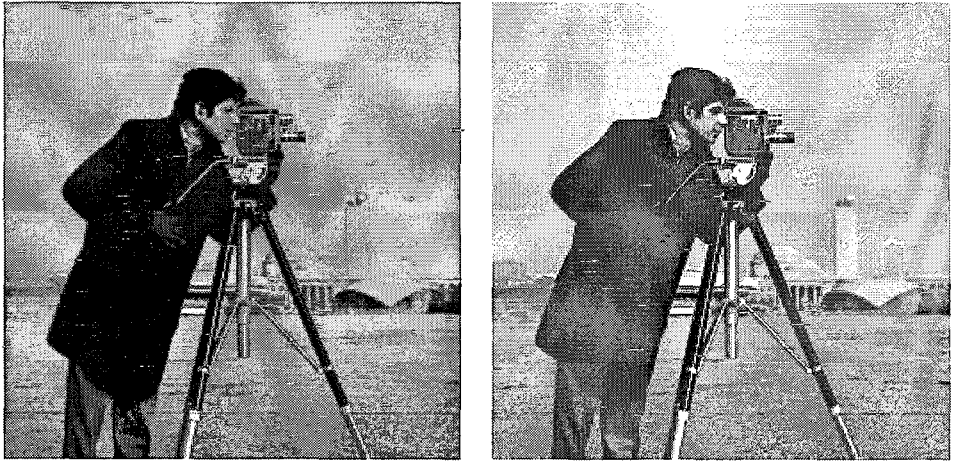
2.2.1. Toplama

Toplama fonksiyonu, iki resimdeki piksel değerlerinin birebir toplanması ile uygulanır. Bu fonksiyon tek başına çok yararlı olmayıp, genelde iki resmin bir arada gösterilmesi amacıyla kullanılır.

$$Q(i, j) = P_1(i, j) + P_2(i, j) \quad (2.1)$$

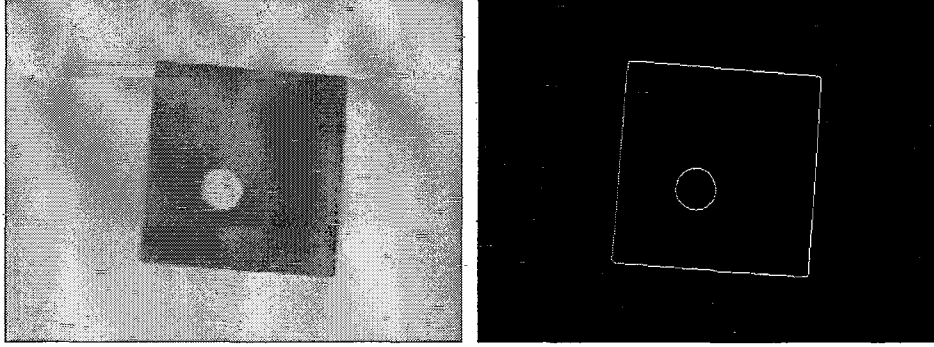
Resim üzerindeki piksel değerlerinin bir C sabiti kadar yükseltilmesi de gerekli olabilir. Resme bir sabitin eklenmesi veya çıkarılması parlaklık ayarı olarak kullanılır.

$$Q(i, j) = P_1(i, j) + C \quad (2.2)$$

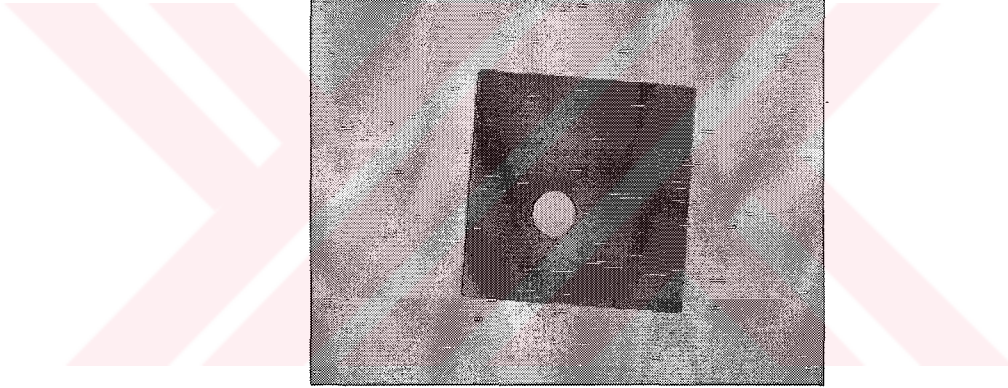


Şekil 2.3 Resme sabit sayı ekleyerek parlaklık ayarı yapılması, C=50

Her bir noktanın 8 bit ile ifade edildiği gri tonlu resimde, her pikselin değeri 0 ile 255 arasında olabilir. Toplama işlemi neticesi, piksel 255 değerini aşar ve doyuma (saturation) ulaşır ise, genelde izin verilen en yüksek seviye olan 255 ile sınırlanır.



Şekil 2.4 Girdi resmi ve canny algoritması ile elde edilen kenar resmi



Şekil 2.5 Toplama fonksiyonu ile orijinal resim ve kenar resminin birlikte gösterimi

2.2.2. Çıkarma

Çıkarma işlemi, iki resmin piksel değerleri birebir çıkarılması ile yapılır.

$$Q(i, j) = P_1(i, j) - P_2(i, j) \quad (2.3)$$

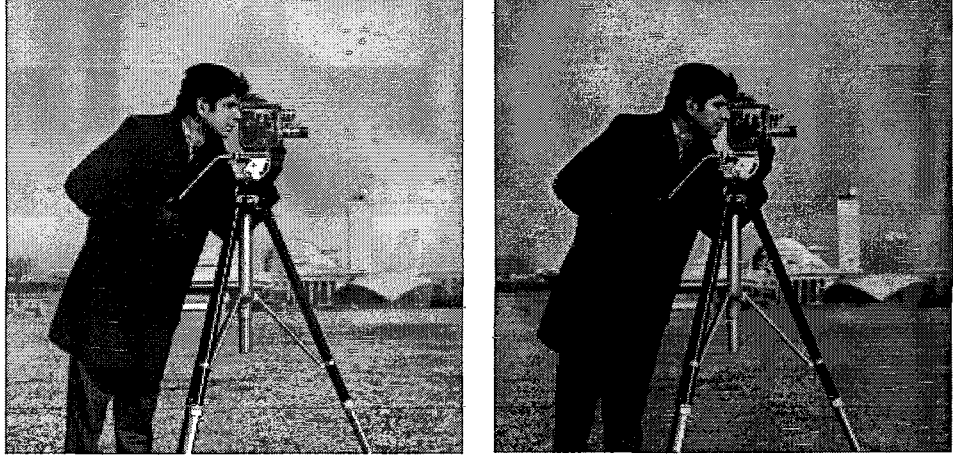
İşlemin sonucu pozitif olması isteneceğinden, sonucun mutlak değeri alınmalıdır.

$$Q(i, j) = |P_1(i, j) - P_2(i, j)| \quad (2.4)$$

Veya, sabit bir C değeri resimden çıkarılması gerekli olabilir. Resimden sabit bir sayı çıkarılarak parlaklık kısılabılır.

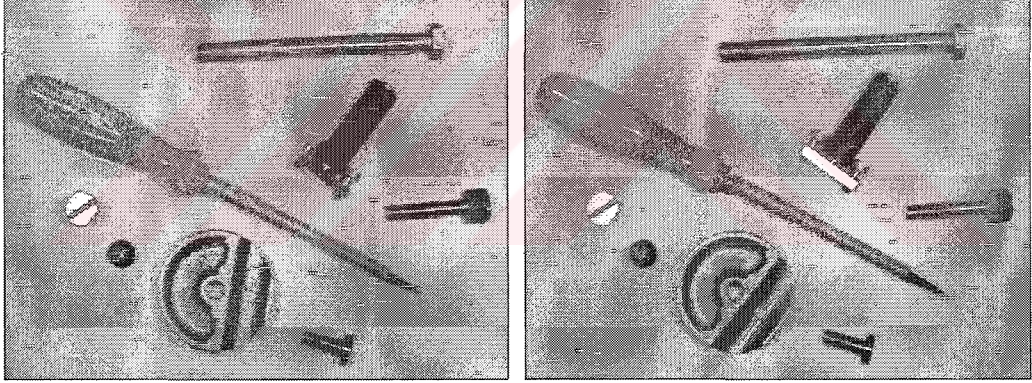
$$Q(i, j) = P_1(i, j) - C$$

(2.5)

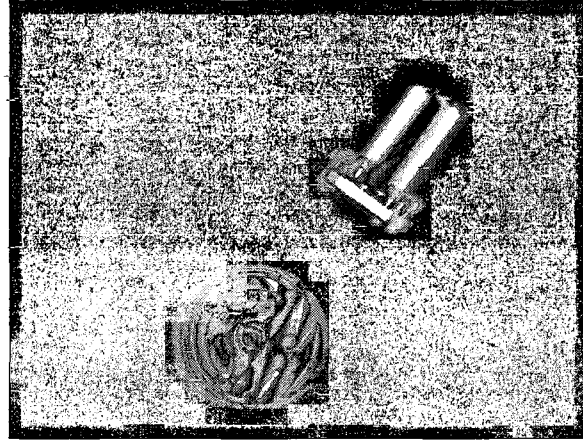


Şekil 2.6 Resimden sabit sayı çıkararak parlaklık ayarı yapılması, $C = 50$

Çıkarma işlemi, iki resim üzerinde cisimlerin yer değiştirmeleri tespit etmek üzere kullanılabilir.



Şekil 2.7 Resimde bazı parçaların yerleri değiştiriliyor [2]



Şekil 2.8 İki resmin farkı alınarak yer değişikliklerinin tespit edilmesi [2]

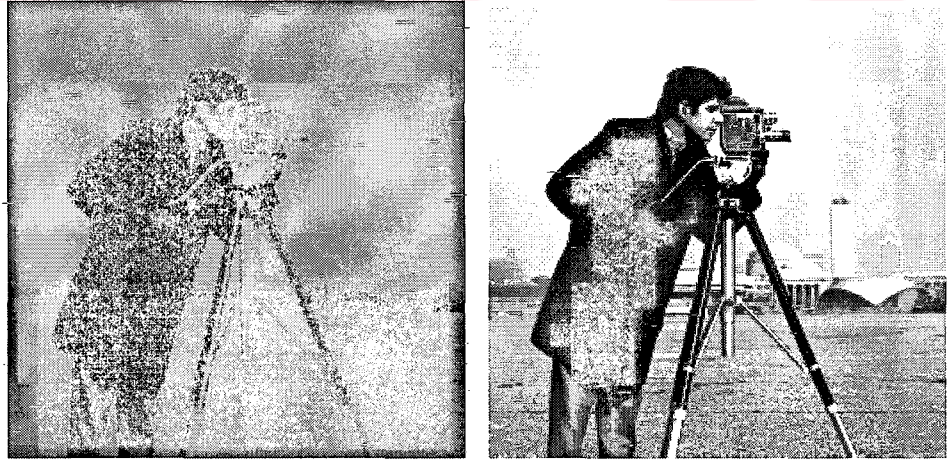
2.2.3. Çarpma

İki resmin çarpımı, karşılıklı her bir piksel değerinin birbiri ile çarpılması ile elde edilir.

$$Q(i, j) = P_1(i, j) \times P_2(i, j) \quad (2.6)$$

Resim piksel değerleri bir sabit sayı ile çarpılarak resmin ışıklılığı (kontrast, zıtlık) değiştirilebilir.

$$Q(i, j) = P_1(i, j) \times C \quad (2.7)$$



Şekil 2.9 Çarpa örneği, $C = 0.5$ ile kontrast azaltma, $C = 1.5$ ile kontrast artırma

2.2.4. Bölme

Bu işlem, iki resmin her bir piksel değeri karşılıklı olarak birbirine bölünerek veya tek resmin piksel değerleri bir sabite bölünerek yapılır.

$$Q(i, j) = P_1(i, j) / P_2(i, j) \quad (2.8)$$

Bir resim, sabit bir değere de bölünebilir.

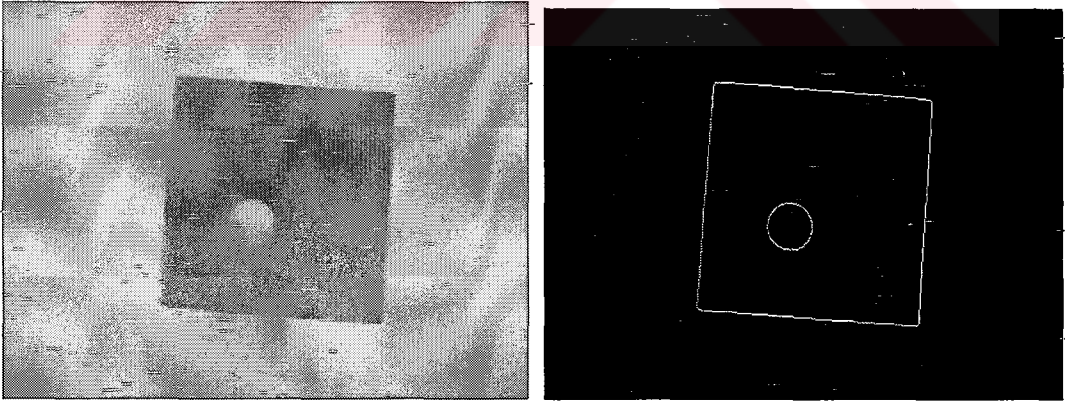
$$Q(i, j) = P_1(i, j) / C \quad (2.9)$$

Bölme işlemi de, çıkarmada olduğu gibi iki resim arasındaki farkları elde etmek üzere kullanılabilir.

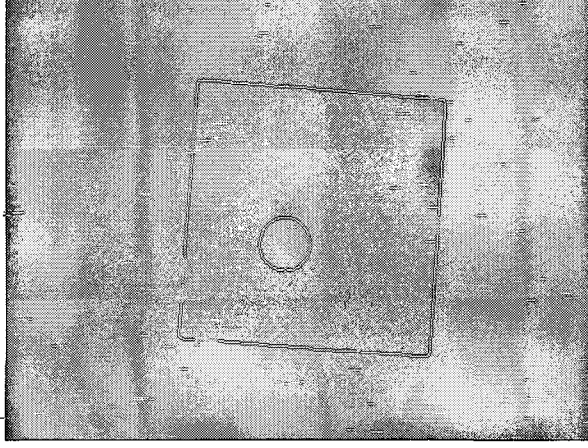
2.2.5. Girişim

Bu işlemde, iki resmin piksel değerleri belirli oranlarda birbirine katılarak yeni bir resim elde edilir. Aşağıdaki ifadede P1 ve P2 kaynak resimler, Q ise çıkış resmini ifade ediyor.

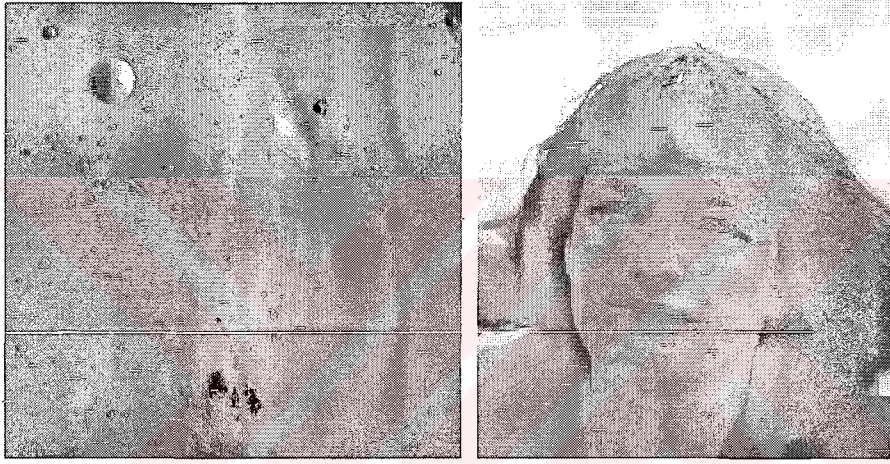
$$Q(i, j) = X \times P_1(i, j) + (1 - X) \times P_2(i, j) \quad (2.10)$$



Şekil 2.10 Orijinal resim ve canny algoritması ile elde edilen kenar resmi



Şekil 2.11 Orijinal resim ve kenar resminin birleştirilmesi, $X = 0.5$



Şekil 2.12 İki girdi resmi [2]

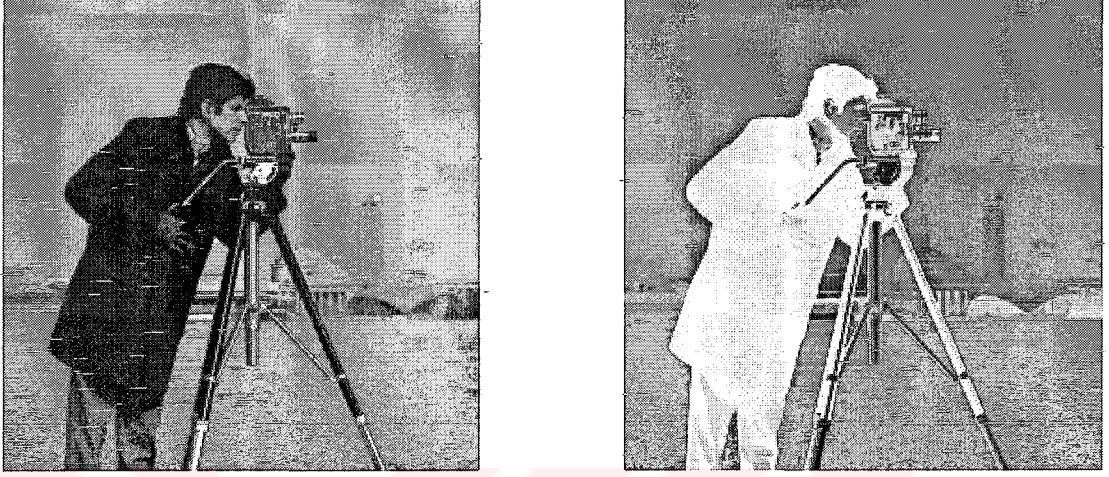


Şekil 2.13 Girişim fonksiyonu ile resimlerin birleştirilmesi, $X = 0.5$

2.2.6. Olumsuzlama

Bu fonksiyon resmin negatifini elde etmek üzere kullanılır.

$$Q(i, j) = 255 - P(i, j) \quad (2.11)$$



Şekil 2.14 Olumsuzlama ile negatif resim elde edilmesi

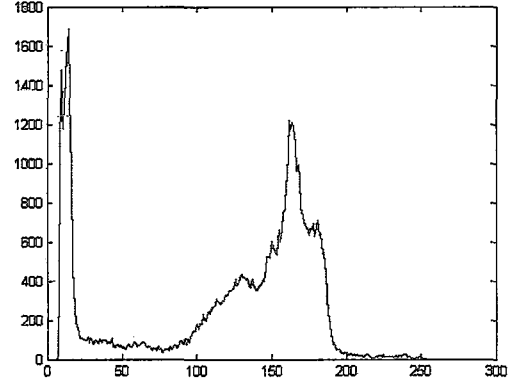
2.3. Noktasal Görüntü İşleme Fonksiyonları

Bu kategoriye dahil fonksiyonlarda, girdi resminin tüm piksellerine bazı fonksiyonlar uygulanarak, orijinal resmin piksel değerleri değiştirilir.

2.3.1. Histogram

Resimdeki her bir piksel değeri sayılarak, her bir piksel değerinin resimde kaç adet bulunduğu tespit edilir. Bu bilgi, resimde yoğun olarak bulunan gri seviyelerinin tespit edilmesine yarar.

Örnek olarak, her bir pikselin 8 bitlik bir tamsayı ile ifade edildiği gri tonlu bir resimde, piksel değerleri 0 – 255 arasında değişir. Aşağıda, orijinal resimdeki 0 - 255 arasındaki her bir gri seviyesinin resimde kaç adet bulunduğu grafik olarak gösteriliyor.



Şekil 2.15 Histogram. Yatay eksen gri seviyeleri, düşey eksen gri seviyelerin adetlerini gösteriyor

2.3.2. Eşikleme

Eşikleme fonksiyonu, resimdeki belirli genliği aşan piksellerin alınıp, diğer piksellerin sıfıra çekilmesi şeklinde uygulanır.

Çoğu görüntü işleme uygulamasında, görüntüdeki ana parçalar ile zeminin birbirinden ayrıştırılması gerekebilir. Bu durumda eşikleme metodu kullanılarak, resimdeki parçalar zeminden ayrı olarak elde edilebilir.



Şekil 2.16 Eşikleme metodu ile piksel değeri 20 ve altında olan noktalar alınıyor

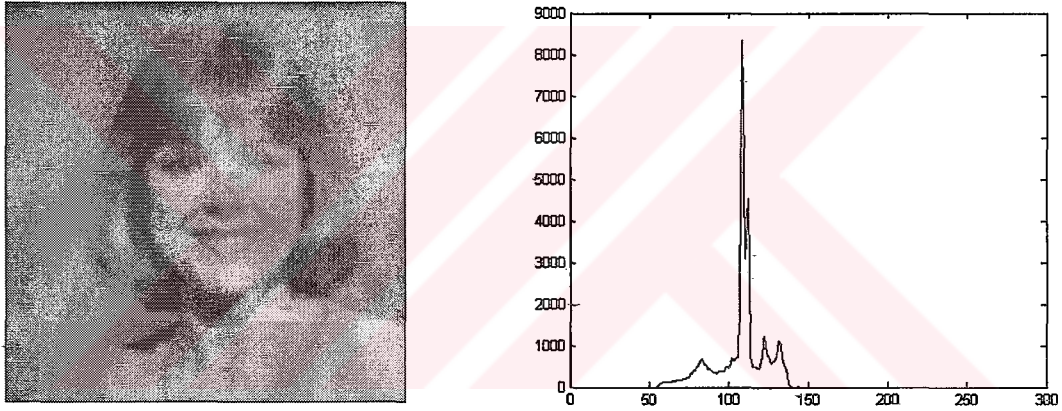
Eşikleme metodu, Canny ve benzeri kenar bulma algoritmalarında, kenar noktalarını içeren ikili (0 ve 255 değerlerinden oluşan ikili-binary resim) elde etmek üzere kullanılır.

Eşik seviyesinin dinamik olarak tespiti görüntü işleme uygulamalarında ayrı bir problemi teşkil eder.

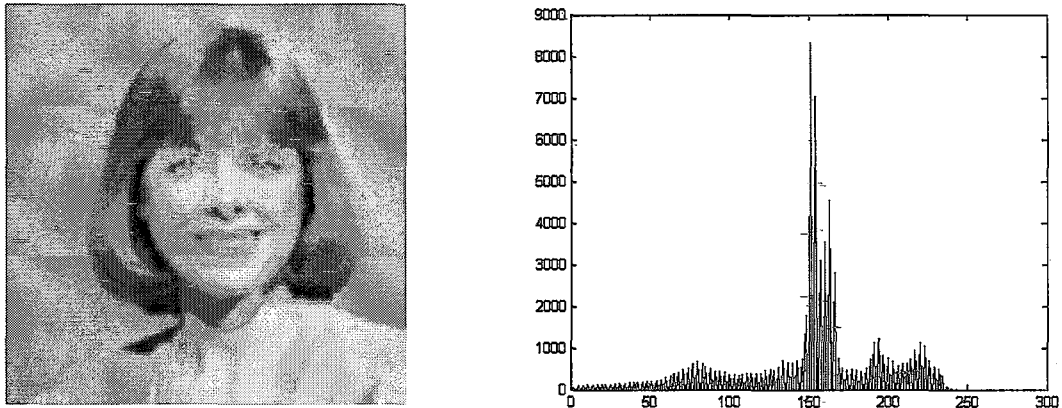
2.3.3. Kontrast yayma (normalizasyon)

Kontrast yayma işleme, resimdeki piksel değerlerinin istenen bir aralığa doğrusal olarak dağıtılması şeklinde uygulanır. Örnek olarak verilen resimde alt eşik değeri 100, üst eşik değeri 200 olarak giriliyor ve resmin piksel değerleri bu aralığa doğrusal olarak yayılıyor.

$$imge[i, j] = \frac{imge[i - j] - altesik}{üsteesik - altesik} \times max_gri_ton_seviyesi \quad (2.12)$$



Şekil 2.17 Kontrast yayma öncesi resim



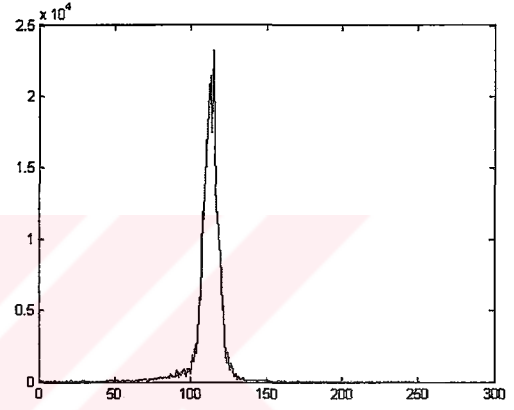
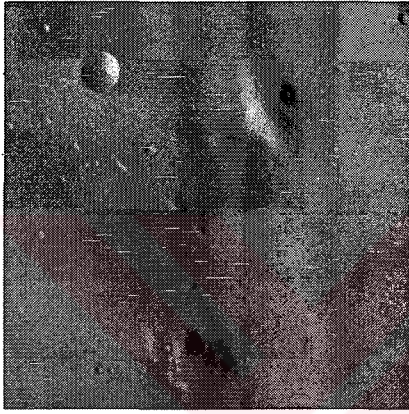
Şekil 2.18 Kontrast yayma sonucu elde edilen resim ve histogram grafiği

2.3.4. Histogram dengeleme

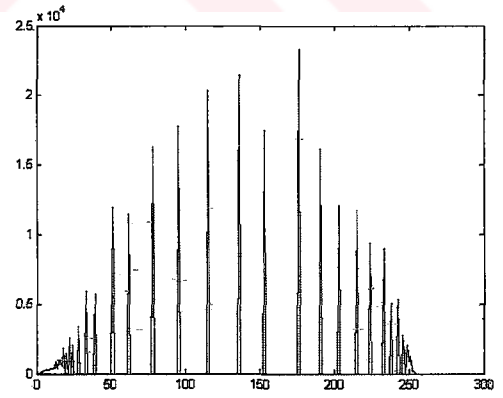
Histogramdaki frekans bilgisine bağı olarak yapılan doğrusal olmayan eşlemedir.

Yüksek frekanslı piksel seviyesi geniş piksel alanına, düşük frekanslı piksel seviyesi ise dar piksel alanına genişletilmektedir. Bu sayede çok kullanılan piksel seviyeleri belirgin hale dönüştürülmektedir.

$$G_{yeni} = \frac{gri_ton_seviyesi \leq G}{toplamlam_piksel_sayisi} \times max_gri_ton_seviyesi \quad (2.13)$$



Şekil 2.19 Histogram dengeleme öncesi- girdi resmi



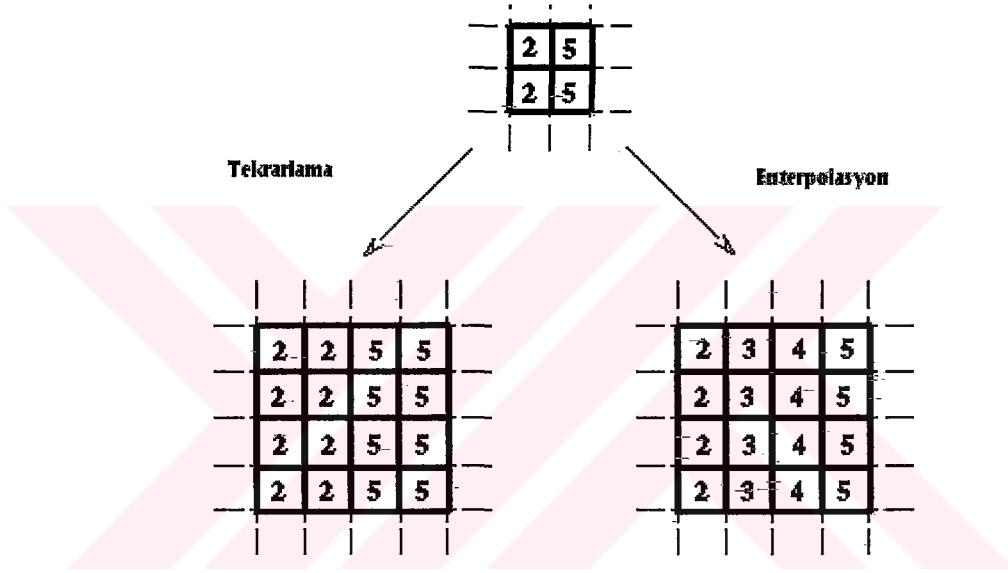
Şekil 2.20 Histogram dengeleme sonrası elde edilen resim ve histogramı

2.4. Geometrik Görüntü İşleme Fonksiyonları

Bu kategorideki görüntü işleme fonksiyonları resmin boyutunun değiştirilmesi, ters çevrilmesi, aynalanması gibi çeşitli geometrik işlevleri yerine getirir.

2.4.1. Boyut değişikliği

Boyut değişikliği, resmin boyutlarının belirli bir oranda büyütülmesi için uygulanır. Yeni piksel değerleri, resmin mevcut piksel değerleri tekrarlanarak veya komşu piksel değerlerinin ortalaması alınarak elde edilir.



Şekil 2.21 Değerlerin tekrarlanması veya enterpole edilmesi ile boyut değişikliği



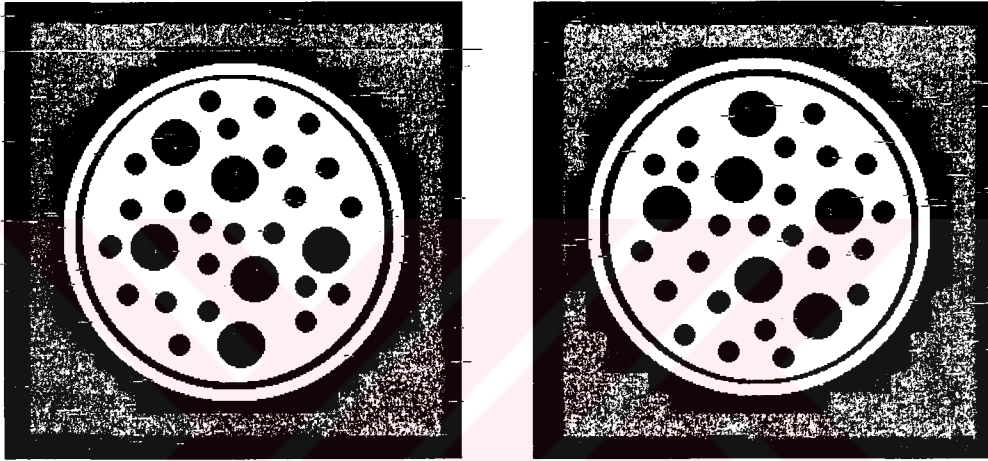
Şekil 2.22 Boyut değişikliği uygulaması

2.4.2. Döndürme

(x_0, y_0) noktası girdi resminin merkez koordinatları olmak üzere, girdi resminin (x_1, y_1) noktası, çıktı resminde (x_2, y_2) noktasına eşlenir. Bu şekilde resim üzerinde bir merkez etrafında döndürme işlemi yapılır.

$$x_2 = \cos(\theta) * (x_1 - x_0) - \sin(\theta) * (y_1 - y_0) + x_0 \quad (2.14)$$

$$y_2 = \sin(\theta) * (x_1 - x_0) + \cos(\theta) * (y_1 - y_0) + y_0 \quad (2.15)$$



Şekil 2.23 Girdi resminin dairenin merkezi etrafında 180 derece döndürülmesi

2.4.3. Aynalama, simetri

Bu fonksiyonda, girdi resminin her (x_1, y_1) pikseli, bir nokta veya eksene göre simetriği alınarak, çıktı resminde (x_2, y_2) noktasına eşlenir.

$x = x_0$ dikey ekseni etrafında aynalama:

$$x_2 = -x_1 + (2 * x_0), y_2 = y_1 \quad (2.16)$$

$y = y_0$ yatay ekseni etrafında aynalama:

$$x_2 = x_1, y_2 = -y_1 + (2 * y_0) \quad (2.17)$$



Şekil 2.24 Girdi resminin aynalanması

2.4.4. Öteleme

Öteleme fonksiyonunda, girdi resminin her (x_1, y_1) pikseli yatay veya düşeyde belirli bir miktar kaydırılarak, çıktı resmindeki (x_2, y_2) noktasına eşlenir. Yatay ve düşey eksenlerdeki öteleme miktarı (β_x, β_y) olmak üzere, çıktı resmi şu şekilde elde edilir:

$$x_2 = x_1 + \beta_x, \quad y_2 = y_1 + \beta_y \quad (2.18)$$



Şekil 2.25 Öteleme işlemi, girdi resmine $\beta_x = 15, \beta_y = 30$ öteleme uygulanıyor

2.5. Morfolojik Görüntü İşleme Fonksiyonları

Morfoloji, canlıların yapıları ve şekilleri ile ilgilenen biyoloji dalıdır. Görüntü işlemede ise, küme işlemlerine dayanan, ikili-binary resimler üzerinde uygulanan bir kısım metodlar bu isimle anılır.

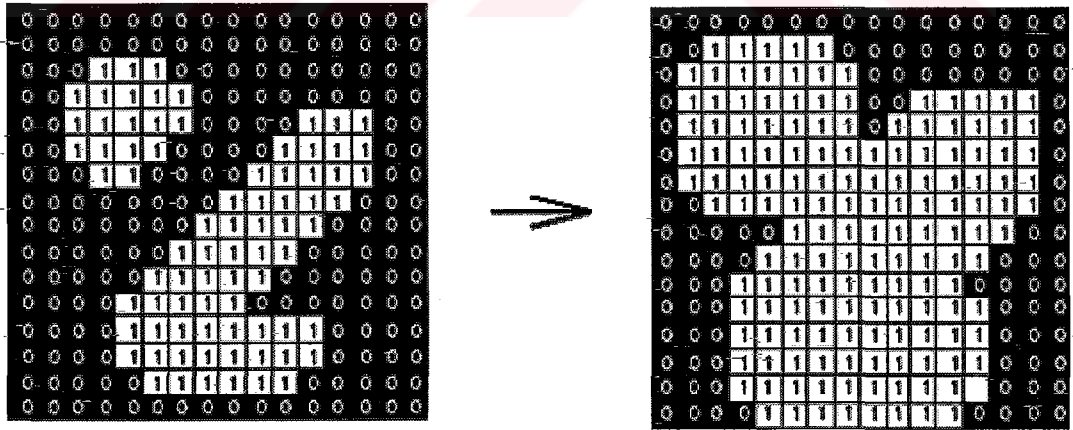
Yayma ve aşındırma fonksiyonları ana morfolojik işlemlerdir. Açma ve kapama işlemleri ise, bu iki ana işlem kullanılarak uygulanır.

2.5.1. Yayma - dilation

Yayma işlemi, cisimdeki nesnelere genişletmek üzere uygulanır. Yayma şekli, yapı elemanı olarak adlandırılan matris ile belirlenir.

Aşağıdaki uygulamada 3x3 kare yapı elemanı $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ girdi resmine uygulanarak

yayma işlemi uygulanıyor. Yapı elemanının merkez noktası girdi resmi üzerinde kaydırılırken, girdi resminde piksel değeri 1 olan noktaların komşu pikselleri, yapı elemanının tüm değerlerine eşitlenir.



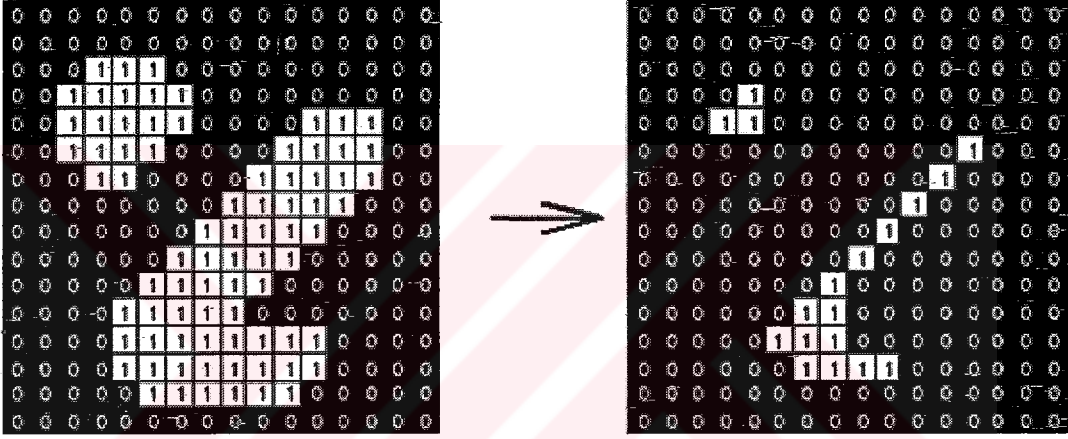
Şekil 2.26 Yayma işlemi ile resimdeki nesnelere genişletilmesi

2.5.2. Aşındırma - Erosion

Aşındırma işlemi, resimdeki nesneyi inceltmek amacıyla kullanılır. Aşağıdaki

örnekte 3x3 kare yapı elemanı $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ girdi resmine uygulanıyor. Yapı elemanının

merkez noktası girdi resmi üzerinde kaydırılırken, girdi resminde yapı elemanı ile tüm olarak çalışan 3x3 bölgelerin sadece merkez noktası 1 değerine eşitlenir ve diğer noktaları sıfıra çekilir.

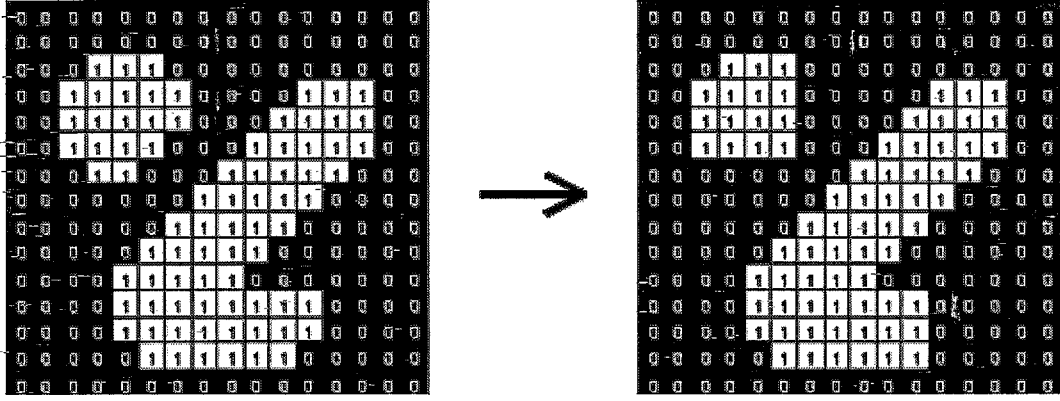


Şekil 2.27 Aşındırma işlemi ile nesnelerin erozyonu

2.5.3. Açma

Açma işlemi, girdi resmine aşındırma ve sonrasında yayma uygulanarak yapılır.

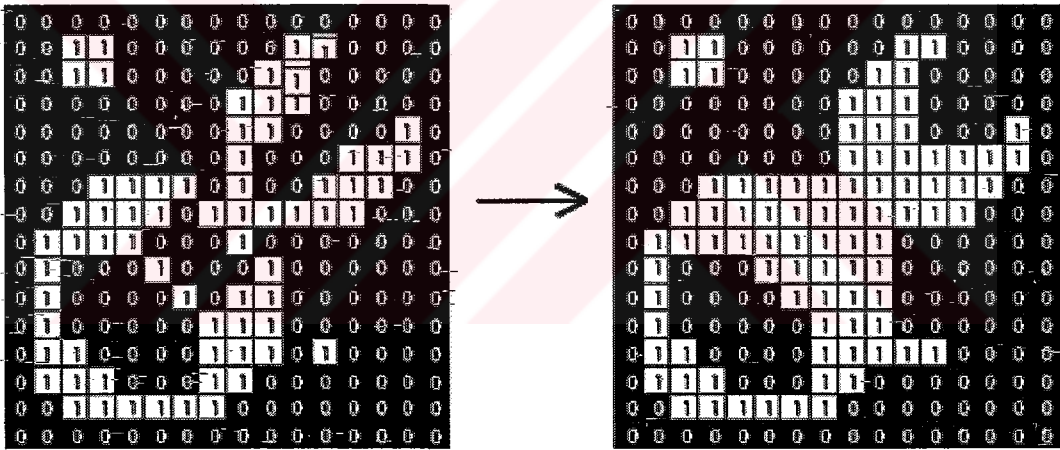
Aşındırma ve yayma işlemlerinde kullanılan yapı elemanları aynı olmalıdır. Bu işlem ile resimdeki nesnelerin çevre pikselleri yok edilmeye çalışılır.



Şekil 2.28 Açma işlemi sonucu elde edilen resim

2.5.4. Kapama

Kapama işlemi, girdi resmine yayma ve sonrasında aşındırma uygulanarak yapılır. Aşındırma ve yayma işlemlerinde aynı yapı elemanları kullanılmalıdır. Bu işlem ile resimdeki nesnelere çevre pikselleri tamamlanır.



Şekil 2.29 Kapama işlemi sonucu elde edilen çıktı resmi

BÖLÜM 3. KENAR BULMA ALGORİTMALARI, VEKTORİZASYON

3.1. Giriş

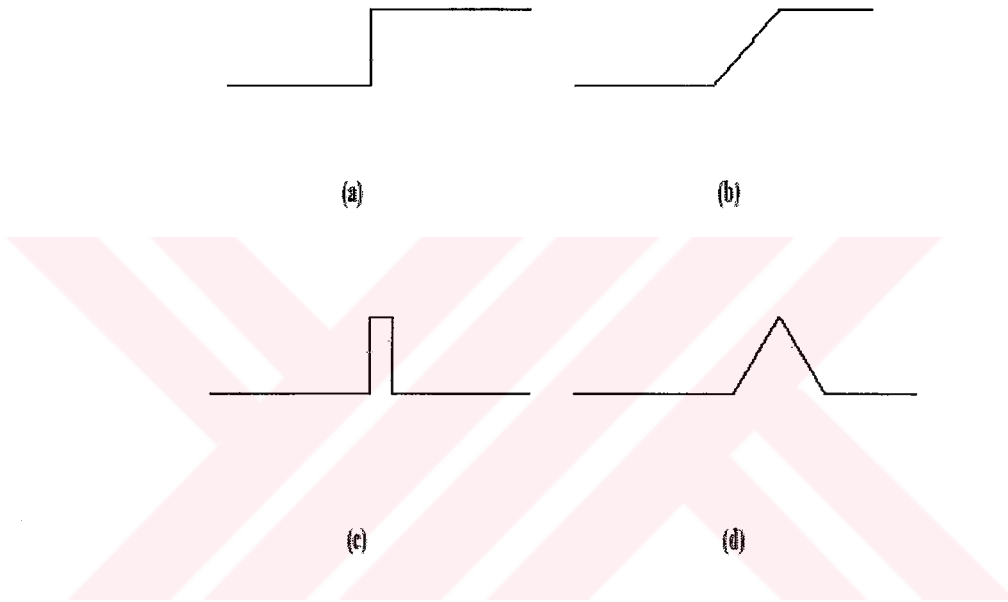
Bilgisayarlı görme problemlerinde kenar bulma algoritmaları sürekli olarak önemli bir problemi teşkil etmiş ve araştırmacıların ilgisini çekmiştir. Genel anlamda bakıldığında kenar bulma problemi halihazırda tam ve mükemmel olarak çözülebilmüş sayılmaz. Kenar bulma problemi görüntü işlemede alt seviye bir problemi teşkil eder. Üst seviyedeki görme problemleri kenar ve köşe bulma algoritmaları ile elde edilen verilere bağımlı olduğu için, kenar bulma aşamasının çok iyi sonuçlar üretmesi istenir. Görme işlevinin gerçekleştirilebilmesi için resimdeki nesnelere ve nesnelere ayıran hatların bitmap resim bilgisinden ayrılması öncelikli problem olarak karşımıza çıkar.

Örneğin stereo görme tekniklerinde, derinlik bilgisini elde etmek için, sağ ve sol kanal olmak üzere iki farklı kaynaktan alınan resimlerin özellik noktaları bulunarak birbiri ile eşleştirilmeye çalışılır. Özellik noktalarının bulunmasında kenar ve köşe bilgisi önemli bir yer tutar. Hareket algılama problemlerinde ise, hareket eden nesnenin bulunması amacıyla, resimdeki kenar ve köşe bilgisinin zaman içindeki değişimi izlenmeye çalışılır. Üç boyutlu nesnelere algılanması problemlerinde de, farklı iki kaynaktan alınan resimler üzerinde kenar ve köşelerin eşleştirilmesi metodu kullanılır. İki boyutlu nesne algılama problemlerinde ise, kenar bilgisi anahtar durumdadır.

Bu çalışmada temel problem, iki boyutlu cisimlerin CAD ortamına aktarılması ve üç eksenli robot tarafından düzlemde çizdirilmesidir. İlk aşamada kameradan alınan resim, kenar bulma algoritmaları ile kenar resmine dönüştürülür ve tezgahdaki parçalar elde edilir. Sonrasında ise, kenar resmi CAD formatlı vektör resmine dönüştürülerek robota iletilebilir bir formata taşınır.

3.2. Bitmap Resimde Kenar Tipleri

Gri tonlu bir resimde, nesnelerin kenarlarının diki boyunca gri tonunun seviyesi hatırı sayılır oranda deęiřir. İdeal bir kenar, adım – step fonksiyonu ile modellenenebilir. Gerçek bir resimde ise, kenarın olduęu istikamette piksel deęeri bir anda deęil, kademeli olarak deęiřir. Dolayısıyla ancak bir rampa řeklinde modellenenebilir. Bazen de iki adım veya rampa kenar yan yana gelerek darbe veya çatı řeklinde bir sinyal üretebilirler.



Şekil 3.1 Kenar tipleri [3] (a) Adım, (b) Rampa, (c) Darbe, (d) Çatı

3.3. Kenar Bulmada Kullanılan Ařamalar

Çoęu kenar bulma algoritması üç ařamadan oluşur: filtreleme, türevleme ve kenar bulma. Filtreleme ařamasında resim üzerindeki gürültüleri yok etmek üzere resim bir süzgeçten geçirilir. Resim üzerindeki gürültüler kameranın ve nesnelerin fiziki yapısından kaynaklanabileceęi gibi, dijital resmin elde edilmesinde kullanılan metodlardan da kaynaklanabilir, örnekleme, quantalama hataları gibi. Türevleme ařamasında, kenar bölgelerindeki yoğunluk deęişiminden yararlanarak, resimdeki nesnelerin kenar bölgeleri parlak hale getirilir. Kenar bulma ařamasında ise, türevleme ile parlatılmış kenar bölgelerinden kenar resmi elde edilmeye çalışılır.

İlk olarak Sobel, türevleme aşamasından önce, resimdeki piksel değerlerini, komşu pikselleri ile ortalama alarak yumuşatmış, bu şekilde resmi bir süzgeçten geçirmiştir. Böylece Sobel operatörü daha güzel sonuçlar üretmiştir. Türevleme sonrası parlatılan kenar bölgelerinin belirli bir eşikten yukarıda olanları kenar resmine seçilir. Bu eşiğin dinamik olarak tespit edilmesi halihazırda önemli bir problemi teşkil etmektedir. Canny algoritmasında, filtreleme amaçlı olarak Gauss yumuşatması kullanılır. Sonrasında türevlemeye geçilir. En son aşamada ise, türev genliğinin kenara dik doğrultuda en yüksek değere sahip olduğu noktalar kenar resmine seçilir.

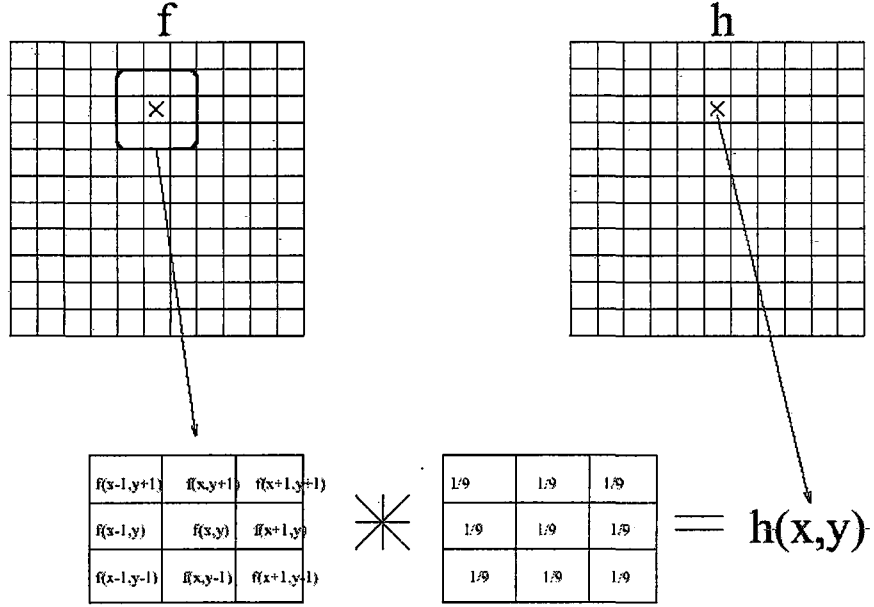
3.3.1. Filtreleme aşaması

En basit filtre, çevre piksellerin ortalamasını alarak uygulanan ortalama - mean filtresidir. Bir pikselin değeri, komşu pikseller ile ortalaması alınarak yeniden hesaplanır. Böylece resimdeki gürültü seviyesi ve keskinlikler azaltılır. f girdi resimini, h çıktı resmini, g ise filtre maskesini göstermek üzere, ortalama – mean filtresi şu şekilde hesaplanır:

$$h(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j)g(i, j) \quad (3.1)$$

$$h(x, y) = f(x, y) * g(x, y) \quad (3.2)$$

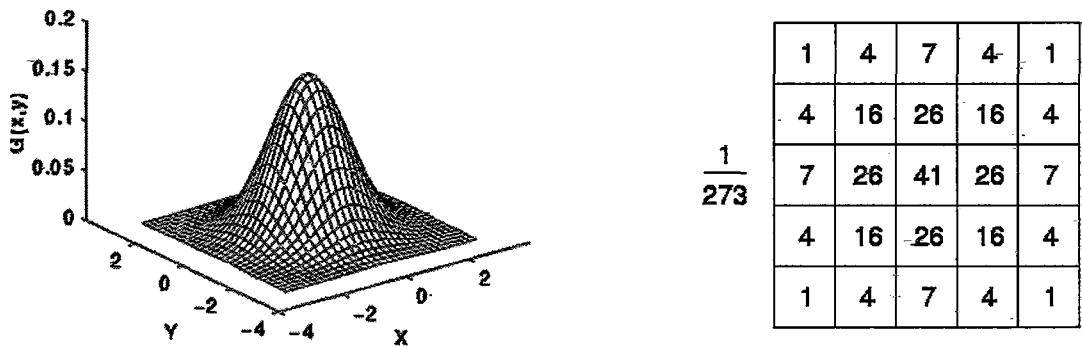
Burada g filtre maskesini, $*$ işareti ise konvolüsyon işlemini ifade ediyor. Örnek şekilde, girdi resmi, 3×3 boyutlarında ve $1/9$ değerine sahip maske ile konvolüsyona tabi tutuluyor. Filtre maskesinin merkezi resmin pikselleri boyunca sola sağa ve yukarıdan aşağıya gezdirilirken, her bir noktada, 3×3 bloğun piksel değerleri ile maskenin karşılık hücrelerindeki değerler çarpılarak toplanır. Bu şekilde merkez hücrenin yeni değeri hesaplanır ve bu değer çıktı resminde ilgili hücreye atanır.



Şekil 3.2 Ortalama – mean filtre ile resmin konvolüsyonu, 9 komşu pikselin aritmetik ortalaması yeni piksel değerini üretiyor. [3]

Ortalama olarak uygulanan mean filtrede, komşu pikseller ile eşit ağırlıkta ortalama alınarak yumuşatma uygulanır. Gauss filtresinde ise, gauss fonksiyonuna bağlı olarak merkeze yakın noktaların ağırlıkları yüksek, merkezden uzak noktaların ağırlıkları ise düşük olarak alınır. Gauss fonksiyonu şu şekilde tanımlanır:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.3)$$



Şekil 3.3 $\sigma = 1$ değeri için gauss fonksiyonu ve tamsayı yaklaşımı

Görüntü işleme uygulamalarında aritmetik işlemlerin optimizasyonu açısından gauss maskesinin tamsayı yaklaşımı işlemlerde kullanılır. Tamsayı çarpmaları işlemci açısından daha az zaman gerektirir.

Aşağıdaki örnek kod parçasında uygulamada kullanılan gauss maskesi ve konvolüsyon kodu görülmüyor:

```
//gauss maskesi
int MASK[5][5] = {2, 4, 5, 4, 2}, {4, 9, 12, 9, 4}, {5, 12, 15, 12, 5},
                 {4, 9, 12, 9, 4}, {2, 4, 5, 4, 2} }

//konvolusyon
for(I=-2; I<=2; I++) {
    for(J=-2; J<=2; J++) {
        Sum = Sum + (Bits[(x+I)*LineWidth+(c+J)] * MASK[I+2][J+2]);
    }
}
Sum = Sum/115
```

3.3.2. Türevleme aşaması

Sürekli bir fonksiyonun türevi şu şekilde tanımlanır:

$$f' = \frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} \quad (3.4)$$

Görüntü işlemede kullanılan resimler kesikli – discrete uzayda tanımlı oldukları için, Δx değeri en az 1 olabilir. Bu durumda türev formülü şu şekilde gelir:

$$f' = \frac{df}{dx} = f(x) - f(x - 1) \quad (3.5)$$

Bu formül, türevin kesikli yaklaşımını ifade eder. Görüntü işlemede türevleme, filtreleme aşamasında olduğu gibi, girdi resmi ile türev maskesinin konvolüsyonu ile elde edilir.

- (a) $f(x)$ 10 10 10 10 10 20 20 20 20 20
 (b) $f'(x)$ 0 0 0 0 0 10 0 0 0 0
 (c) $f''(x)$ 0 0 0 0 0 10 -10 0 0 0
 (d) $\begin{bmatrix} -1 & 1 \end{bmatrix}$ (e) $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$

Şekil 3.4 Bir boyutta türevleme. (a) Girdi fonksiyonu, (b) Fonksiyonun birinci türevine kesikli yaklaşım, (c) Fonksiyonun ikinci türevi, (d) Türev maskesi, (e) Diğer bir türev maskesi

Resimler iki boyutlu olduğundan yatayda ve düşey yönde iki ayrı türev alınarak genlikleri birleştirilir. Kenar açısının kolayca hesaplanabilmesi için, Canny algoritmasında simetrik Sobel türev maskesi kullanılır.

G_x yataydaki türevi, G_y düşeydeki türevi göstermek üzere, gradient genliği (G) ve açısı şu şekilde tanımlanır:

$$\theta = \arctan(G_y / G_x) \quad (3.6)$$

$$G = \sqrt{G_x^2 + G_y^2} \text{ veya yaklaşım ile } |G| = |G_x| + |G_y| \quad (3.7)$$

G gradient genliğinin tam olarak hesaplanması yerine, yatay ve düşeydeki türevlerin mutlak değerlerinin toplamı hesaplamalarda kullanılabilir. Yaklaşım ifadesi, bilgisayarlar açısından daha düşük maliyetli işlem gerektirir. Yatay ve düşeydeki (G_x , G_y) türevlerinin hesaplanması için resim Sobel türev maskesi ile konvolüsyona tabi tutulur.

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

G_x

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

G_y

Şekil 3.5 Yatay ve düşey Sobel türev maskeleri

Türevleme amacıyla Sobel maskesi kullanılabileceği gibi Prewit maskesi de kullanılabilir.

-1	0	+1
-1	0	+1
-1	0	+1

Gx

+1	+1	+1
0	0	0
-1	-1	-1

Gy

Şekil 3.6 Yatay ve düşey Prewit türev maskeleri

Aşağıda, uygulamada türevleme amaçlı kullanılan kod parçası gösteriliyor:

```

//X direction Sobel Mask
int GX[3][3] = { {-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1} };

//Y direction sobel mask
int GY[3][3] = { { 1, 2, 1}, { 0, 0, 0}, {-1, -2, -1} };

////////////////////////////////////
//X gradient yaklaşımı
////////////////////////////////////
for(I=-1; I<=1; I++) {
    for(J=-1; J<=1; J++) {
        sumX = sumX + (BitsGauss[(r+I)*LineWidth+(c+J)] *
GX[I+1][J+1]);
    }
}

////////////////////////////////////
// Y gradient yaklaşımı
////////////////////////////////////
for(I=-1; I<=1; I++) {
    for(J=-1; J<=1; J++) {
        sumY = sumY + (BitsGauss[(r+I)*LineWidth+(c+J)] *
GY[I+1][J+1]);
    }
}

//Gradient genlikleri
Sum = fabs(sumX) + fabs(sumY);

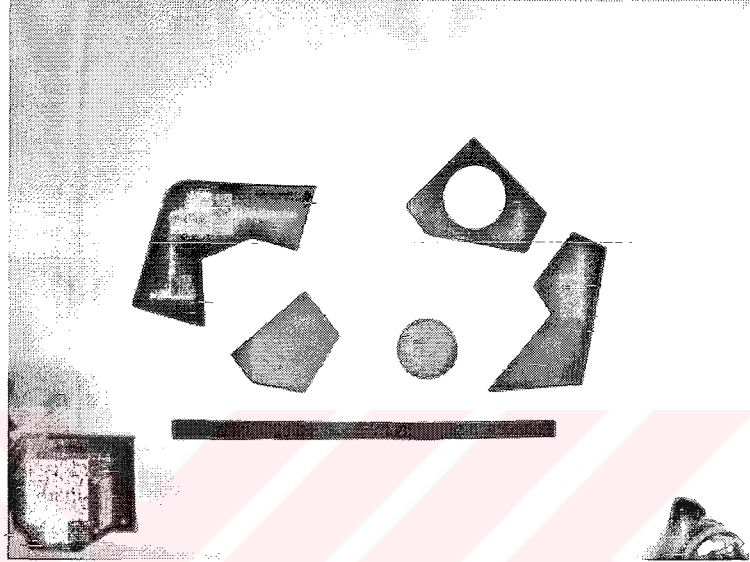
//gradient genlikleri saklanır
BitsGrad[r*LineWidth+c] = Sum;

```

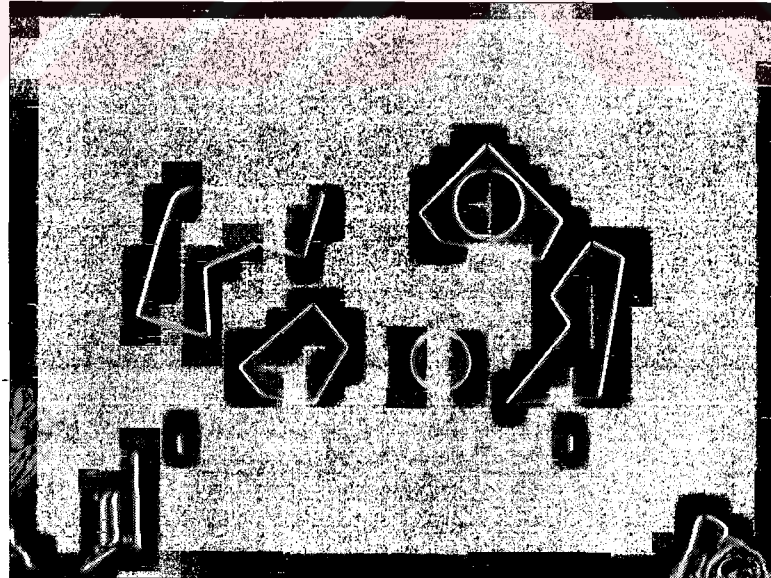


```
//gradient genlikleri surpression-hysterisis icin tekrar saklanir
BitsEdge[r*LineWidth+c] = Sum;

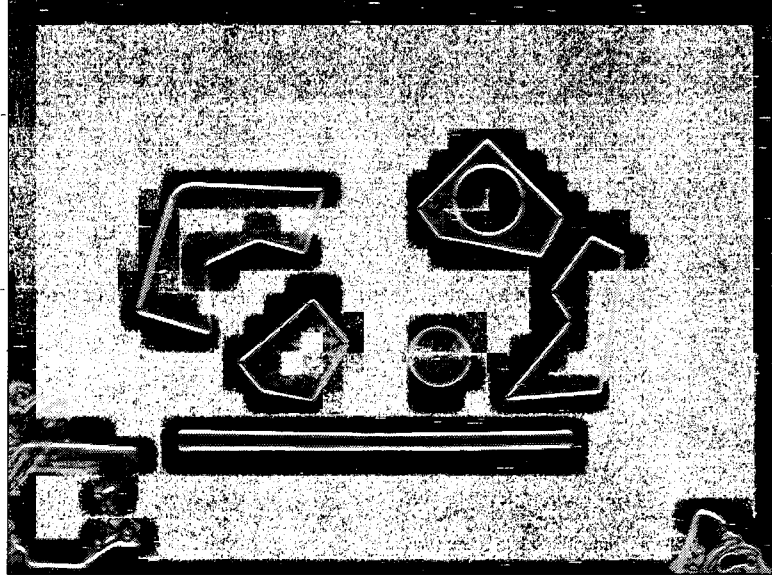
//yatay ve düsey türevler, gösterim amacli tutulur
//negatif türev degerlerine izin var
BitsGradX[r*LineWidth+c] = sumX;
BitsGradY[r*LineWidth+c] = sumY;
```



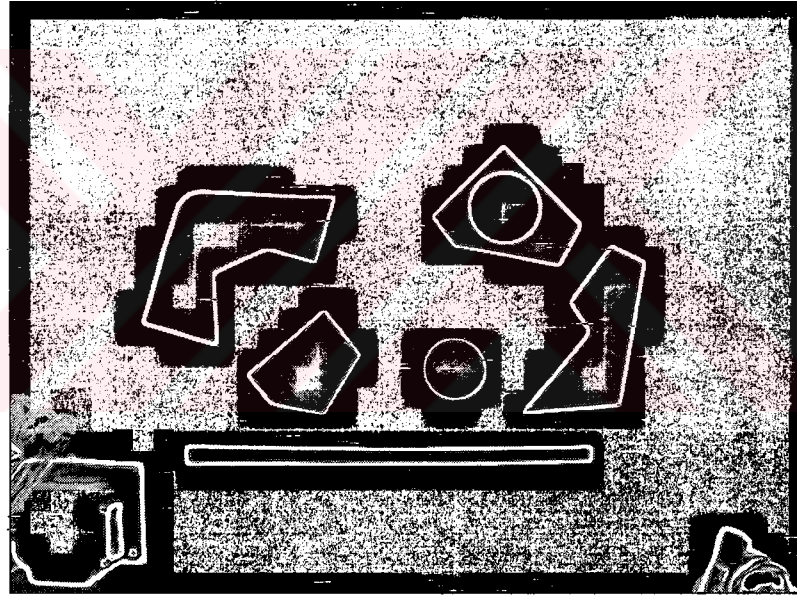
Şekil 3.7 Uygulamada kamera ile alınan görüntü



Şekil 3.8 Yatayda türevleme, Sobel Gx maskesi ile konvolüsyon



Şekil 3.9 Düşeyde türevleme, Sobel Gy maskesi ile konvolüsyon



Şekil 3.10 Gradient genlik resmi. Ekranda gösterim amaçlı olarak normalize edildi.

Kenara dik doğrultudaki orta noktalarda gradient genliği (parlaklık) artar

3.3.3 Eşikleme aşaması

Bu aşamaya kadar elde edilen gradient genlik resmi, belirli bir eşik seviyesi uygulanarak, uygulanan türev maskesine göre, Sobel veya Prewit kenar resmi elde edilebilir. $G(x,y)$ her bir piksel için gradient genliği olmak üzere, gradient genlikleri belirli bir değer aralığına normalize edilir.

$$H(x, y) = \frac{G(x, y)}{\max_{i=1..n, j=1..n} G(i, j)} \times 100 \quad (3.8)$$

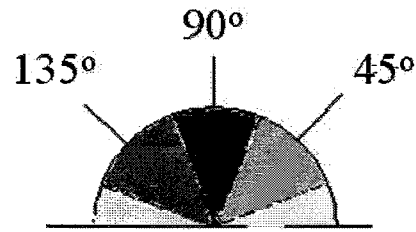
Normalizasyon sonrasında, belirlenen eşikten yüksek olan piksel değerleri Sobel veya Prewit kenar resmine seçilir. T, eşik seviyesine belirtmek üzere, eşik üzerindeki noktalar haricinde diğer noktalara sıfıra çekilerek kenar resminden elenir.

$$E(x, y) = \begin{cases} 1 & H(x, y) > T \\ 0 & \text{diğerleri} \end{cases} \quad (3.9)$$

3.3.4. Maksimum olmayanların bastırılması

Sobel ve Prewit türev maskeleri ile elde edilen kenar resmi kenarları kalın olarak gösterir. Kenarların bir piksel genişliğinde ince olarak gösterilmesi için çeşitli metodlar denenmiştir. Bunlardan en başarılı olanı, iki boyutlu türev resminde, kenara dik doğrultuda maksimum olmayan piksel değerlerinin bastırılması metodudur. Gradient -iki boyutlu türev- yönü, sürekli olarak kenara dik doğrultudadır. Gri tonlar kenarın dik doğrultusunda yoğun bir değişim gösterirler. Diğer bir deyişle gradient resminde, kenara dik doğrultuda maksimum yapan noktalar büyük ihtimalle kenarı belirler ve kenar resmine seçilirler. Gradient türev yönü, yatay ve düşeydeki türevlerle ilişkili olarak şu şekilde ifade edilmiştir: $\theta = \arctan(G_y / G_x)$

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x



Şekil 3.11 Piksel yapısından dolayı gradient açısı dört bölgeye eşlenir [5]

Kenara dik doğrultudaki gradient türev eğimi, piksel yapısından dolayı dört farklı yönde oluşabilir. Bu doğrultular şu şekilde bölgelere ayrılır:

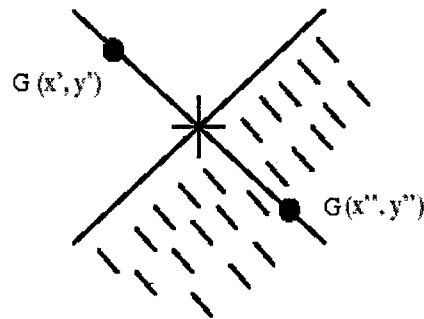
- 0 - 22.5 & 157.5 - 180 = 0 derece
- 22.5 - 67.5 = 45 derece
- 67.5 - 112.5 = 90 derece
- 112.5 - 157.5 = 135 derece

Her bir piksel için gradient yönü $\arctan(Gy/Gx)$ ifadesi ile hesaplandıktan sonra, elde edilen açı değeri istikametindeki iki ilave komşu piksel seçilerek, ortanca pikselin bunlardan yüksek değerde olması istenir. Bu şart sağlanmaz ise, ortanca piksel sıfıra çekilerek kenar resminden elenir. Bu şekilde, kenar resminde sadece kenarlara dik yönde maksimum gradient değerlerine sahip olan pikseller bırakılır.

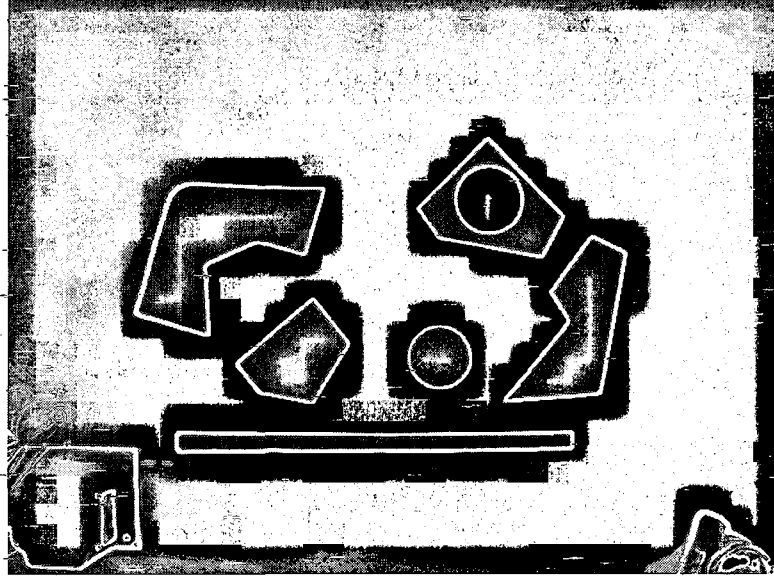
Ayrıca, her bir piksel için hesaplanan gradient türev yönü, geliştirilen vektörizasyon algoritmasında kullanılmak üzere ayrı bir dizide saklanır.

$G(x, y)$ gradient resmi, $G(x', y')$ ve $G(x'', y'')$ gradient yönündeki komşu pikseller olmak üzere, maksimum olmayanların bastırılması şu şekilde tanımlanabilir:

$$G(x, y) = \begin{cases} G(x, y) & \text{eger } G(x, y) > G(x', y') \text{ ve } G(x, y) > G(x'', y'') \\ 0 & \text{digerleri} \end{cases} \quad (3.10)$$



Şekil 3.12 Gradient resminde, kenara dik doğrultudaki komşularından yüksek olmayan gradient genlik değerlerinin olduğu noktalar kenar resminden elenir [3]



Şekil 3.13 Uygulamadan alınan gradient kenar resmi



Şekil 3.14 Maksimum olmayanların bastırılması ile elde edilen kenar resmi. Bir piksel genişliğinde kenar resmi elde ediliyor.

3.3.5. Eşikleme ve süreklilik aşaması

Gradient genlik resminin belirli bir aralığa (örneğin 0-255 arası gri tonları) normalize edilmesi ve kenarların diki boyunca maksimum olmayanların bastırılması sonrasında elde edilen kenar resmi, piksel sürekliliği aşamasına girer. Kenar resmindeki piksel sürekliliğinin test edilmesi amacıyla yüksek ve düşük seviyede iki eşik değeri kullanılır. Süreklilik testi için şu adımlar uygulanır:

- Yüksek ve düşük seviye olmak üzere iki adet eşik seviyesi belirlenir. (255 üzerinden 120 ve 40 gibi)
- Her bir nokta için,
 - Eğer kenara dik maksimum gradient genliği (piksel değeri) yüksek eşikten yüksek ise, kenar olarak bırakılır
 - Eğer kenara dik maksimum gradient genliği (piksel değeri) düşük eşikten düşük ise, sıfıra çekilir
 - Eğer, piksel değeri, yüksek ve düşük eşik arasında ise, bu pikselin yüksek eşiği aşan bir komşusu var ise kenar olarak kalmasına izin verilir.
 - Şayet eşiği aşan bir komşusu yok ise sıfıra çekilerek kenar resminden elenir.

3.4. Kenar Bulma Sonrası Gürültü Eleme Filtresi

Kenar bulma aşamasından sonra elde edilen resimde, herhangi bir nokta ile bağlantısı olmayan tek piksellerin elenmesi istenir. Bağlantısız pikseller sıfıra çekilerek bu işlem tatbik edilir.. Filtreleme sonrası bağımsız ölü pikseller kenar resminde elenir ve vektörizasyon aşamasına katılmaz.

0	0	0	0	0
0				0
0		1		0
0				0
0	0	0	0	0

	0	0	0	
	0	1	0	
	0	0	0	

	0	0	0	
	0	0	0	
	0	0	0	

Şekil 3.15 Gürültü filtresi için çekirdek yapılar. Birinci veya ikinci durum oluşur ise, pikseller sağdaki şekilde sıfıra çekilir.

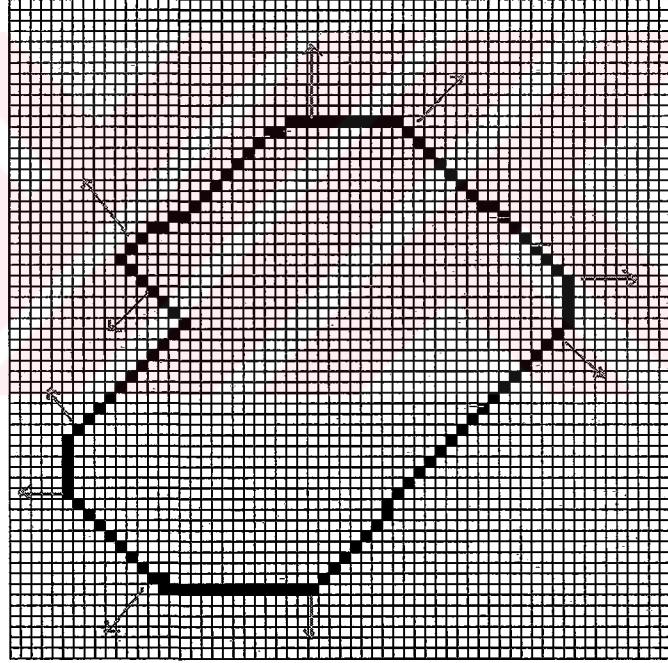
3.5. Vektörizasyon Algoritması

XYZ robotun çalışma tezgahından alınan parçaların kenar resmi bulunduğundan sonra, geliştirilmiş olan, kenar resmi üzerinde gradient - türev yönüne göre piksel izleme temelli vektörizasyon algoritması ile tezgah üzerindeki parçalar CAD ortamına aktarılır.

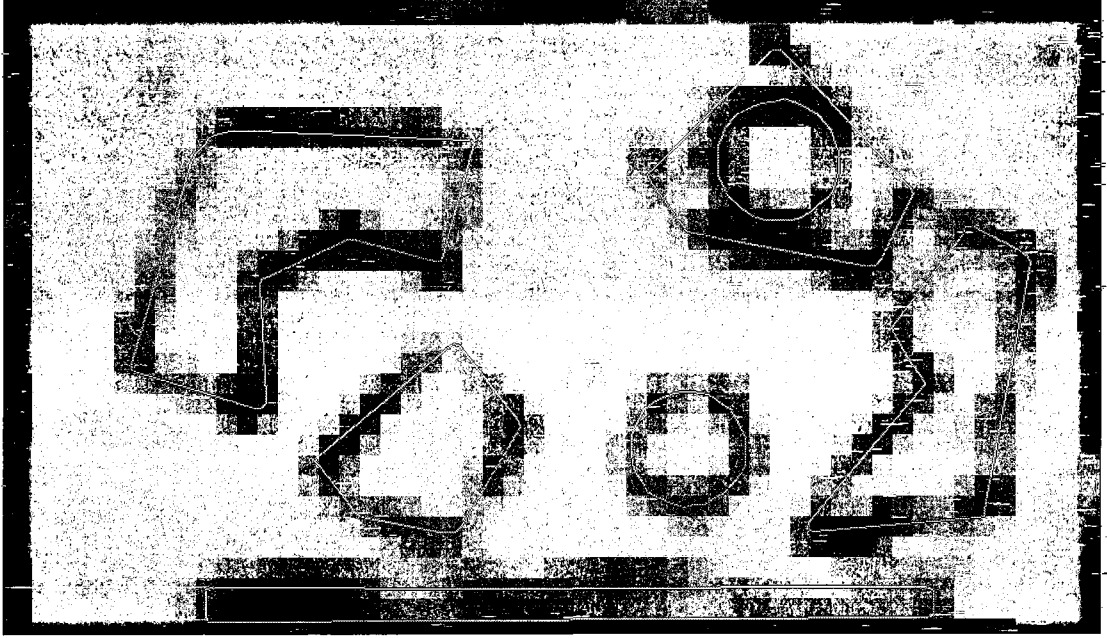
Geliştirilen vektörizasyon algoritması şu şekilde çalışır:

- Kenar resminde bulunan her bir piksel için, piksel değerini, gradient genliklerini ve gradient açılarını hafızaya al
- Tolerans açısı belirle (varsayılan 12 derece)
- Kenar resmini soldan sağa ve yukarıdan aşağıya tarayarak ilk canlı pikseli bul. Canlı piksel var iken aşağıdaki adımları tekrarla
 - Bulunan pikseli sıfıra çek
 - Vektör sayısını bir arttır, yeni vektör aç
 - Bulunan noktayı yeni vektörün başlangıcı olarak al
 - Bulunan noktanın gradient açısını vektöre baz açı olarak sakla
 - Vektör kapanana kadar aşağıdaki adımları tekrarla
 - Eğer pikselin komşusu var ise
 - Komşu pikselin gradient açısını al
 - Komşu pikseli vektörün son noktası olarak farz et
 - Komşu pikselin gradient açısı tolerans dışında ise
 - Önceki eklenen noktayı vektörün kapanışı yap

- Önceki eklenen noktanın komşu piksel sayısı birden fazla ise, diğer vektör başlangıcı için canlı bırak, değil ise sıfıra çek
- Vektörün başlangıç pikseli birden çok komşuya sahip ise, diğer vektör başlangıcı için canlı bırak, değil ise sıfıra çek
- Komşu pikselin gradient açısı tolerans sınırlarında ise
 - Komşu pikseli vektöre ekle
 - Komşu pikseli sıfıra çek
- Pikselin komşusu yok ise
 - Vektörü kapat
 - Son eklenen komşusu olmayan pikseli sıfıra çek



Şekil 3.16 Kenar pikselleri üzerinde gradient açısı değeri izlenmesi metodu ile bitmap resimden vektör resim çıkarımı.



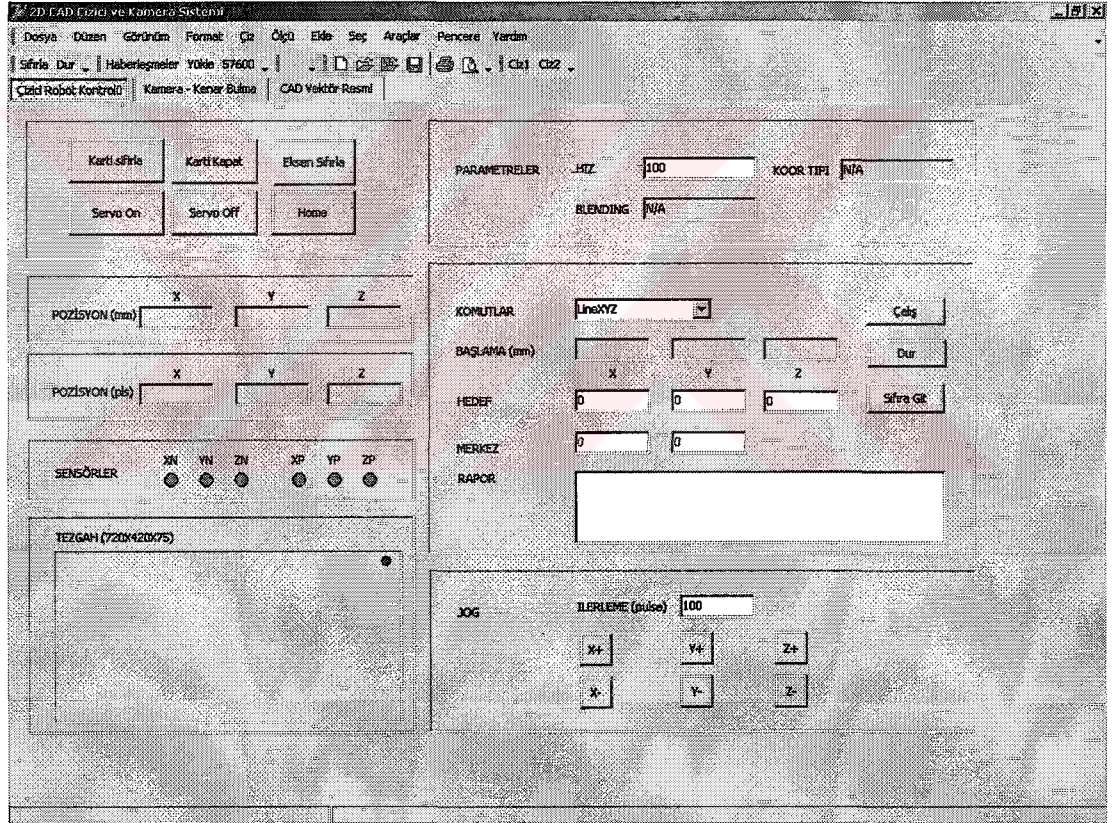
Şekil 3.17 Vektörizasyon sonrası bitmap resimden elde edilen CAD vektör resmi.



BÖLÜM 4. UYGULAMA YAZILIMI

Uygulama yazılımı, kartezyen robot kontrol ekranı, kamera arayüzü, CAD parça programlama ortamı ve CAM – imalat amaçlı olarak tasarlanan parçaların kartezyen robota gönderilmesi kısımlarından oluşur.

4.1. Kartezyen Robot Kontrol Arayüzü



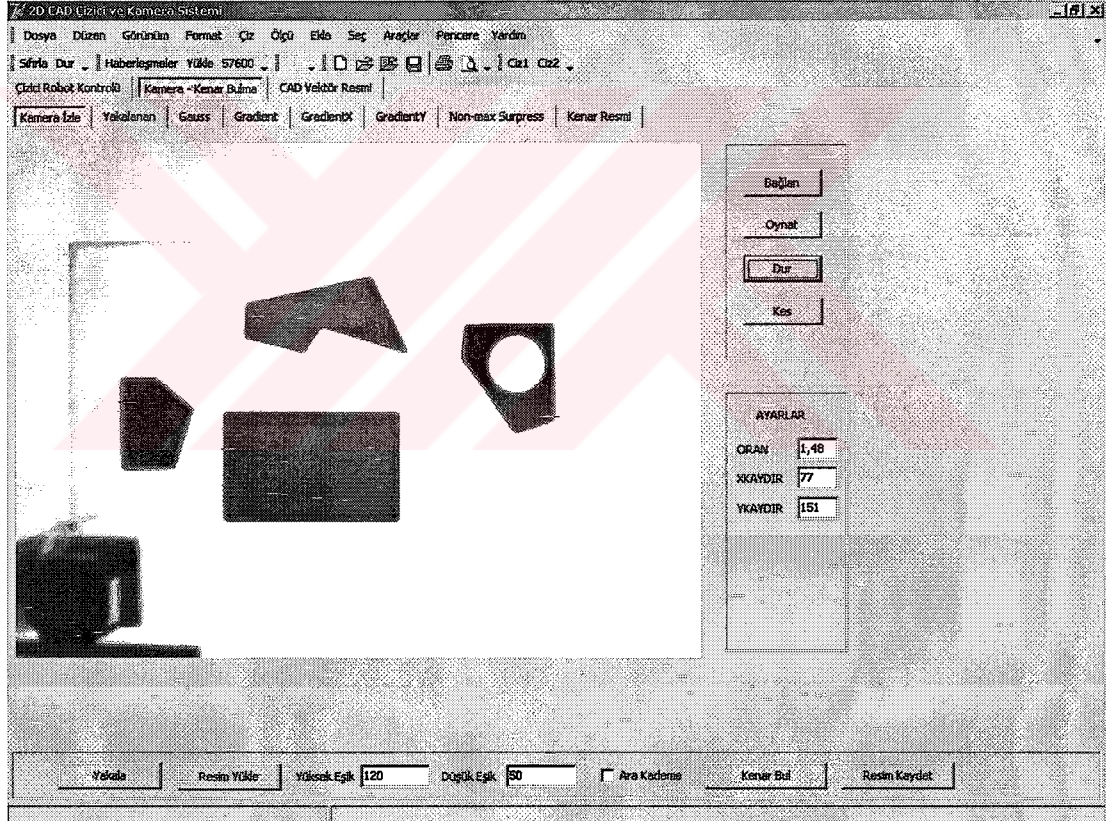
Şekil 4.1. Kartezyen robot kontrol arayüzü

Kartezyen robot kontrol arayüzü, hareket kartının sıfırlanması, servo motorların devreye alınması, eksenlerin sıfırlanması, sınır sensörü bilgilerinin ekrandan izlenmesi ve tezgahın konum bilgisinin izlenmesi fonksiyonlarını sağlar. Ayrıca robota basit seviyede çizgi, yay parçası ve daire komutlarının iletilmesini temin eder.

Jog fonksiyonları ile, her bir eksen için, kartezyen robotun kafasının bulunduğu pozisyon ince ayar yapılabilir.

4.2. Kamera Arayüzü

Kamera arayüzü ile kullanıcı sisteme bağlı kamerayı başlatıp çalışma tezgahı üzerinden görüntü alınmasını sağlar. Yüksek eşik ve düşük eşik parametreleri, kenar bulma algoritmasında kenar süreklilik sağlaması yapılırken kullanılır. Ayarlar kısmındaki Oran, XKaydır ve YKaydır parametreleri, kenar resminden vektör resmine geçişte, ölçeklendirme yapmak üzere yaklaşma oranı ile X ve Y boyunda kayma miktarlarını belirler.



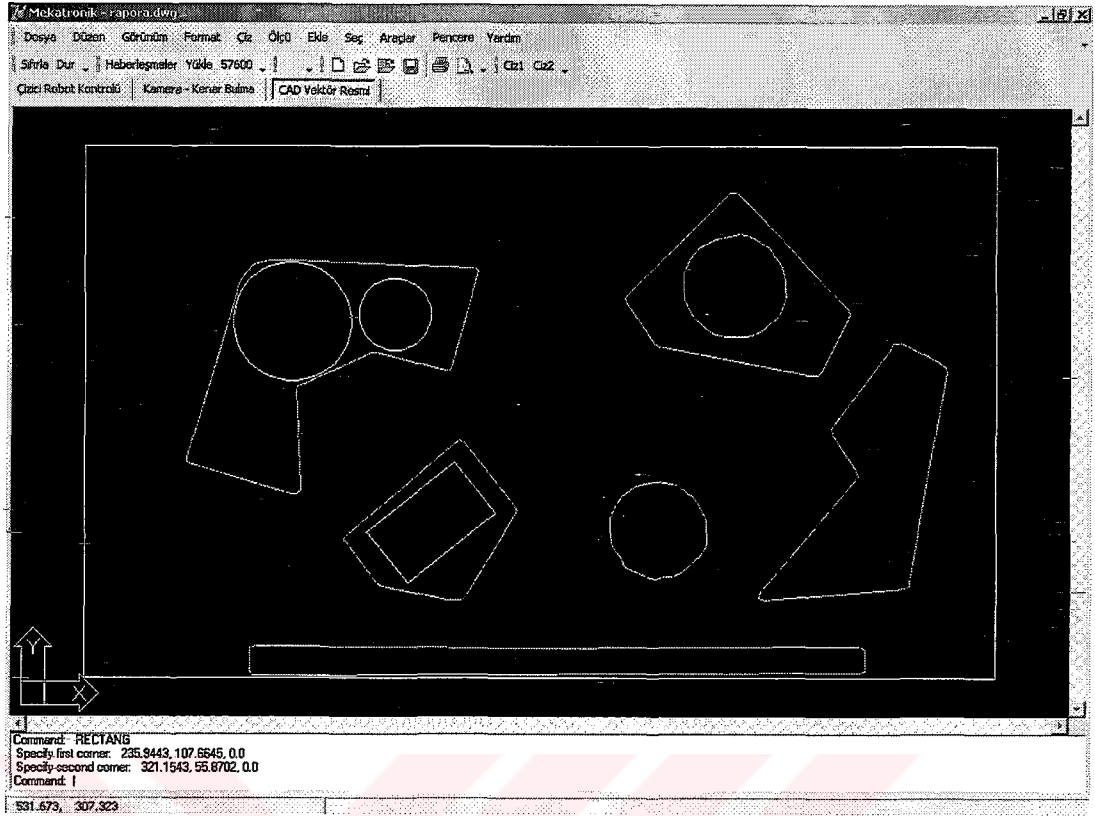
Şekil 4.2. Kamera izleme arayüzü

4.3. CAD-CAM Tasarım Arayüzü

CAD ifadesi, Computer Aided Design / Drafting, bilgisayar destekli tasarım veya çizim kelimelerinin baş harflerinden türetilmiştir. Mühendislik amaçlı çizimlerin bilgisayar ortamında bir veritabanında saklanarak modellenmesini ifade eder. Çizimler, çizgi, açılar, koordinat bilgileri ve ölçeklendirmeleri içerir. Mimari, mekanik tasarım, haritacılık, şehir planlama ve benzeri uygulamalarda standart hale gelmiş CAD vektör resim formatları kullanılır. Vektör resimler, nesnelere koordinat bilgilerini taşıdıkları için, raster - bitmap resimlere göre daha az yer kaplarlar ve dinamik olarak ölçeklendirilebilirler. Diğer taraftan vektör resimler, üç boyutlu uzayı modelleme imkanına sahiptir. Bilgisayar endüstrisinde kullanılan OpenGL ve DirectX grafik kütüphaneleri, bilgisayarlı üç boyutlu simulasyon amaçları için ideal olmakla birlikte, mühendislik çizimlerinde gerekli unsurlara ancak temel seviyede altyapı oluşturabilirler. Open Design Alliance (opendwg.org) isimli kuruluş, mühendisliğe yönelik CAD formatlarının standartlaştırılması konusunda hizmet vermekte ve OpenDWG ismi verilen CAD dosya formatını, mühendislik yazılımı sektörünün önde gelen firmalarına sağlamaktadır. Bu projede, Open Design Alliance kütüphanelerini kullanan VectorDraw (vectordraw.com) ve Kolbasoft (kolbasoft.com) firmalarının CAD-kütüphanelerinin mimari yapıları incelenmiştir.

CAM (Computer Aided Manufacturing, Bilgisayarlı Üretim) ifadesi ise, bilgisayar ortamında çizilen modellerin, çeşitli malzemelerden gerçek boyutlarında üretilmesini ifade eder.

Kenar bulma ve vektörizasyon neticesi elde edilen ana parçanın vektör resmi, CAD-CAM tasarım ve imalat ortamına aktarılır. Kullanıcı, kameradan ölçekli olarak tasarım ortamına aktarılan ana parça üzerinde, üretilecek parçaları tasarlar. Sonrasında, kullanıcı tarafından çizilen yeni parçalar, kartezyen robota hareket komutları şeklinde aktarılarak üretimi gerçekleştirilir.



Şekil 4.3 CAD tasarım ortamı, vektörizasyon uygulaması

4.4. CAD Nesneleri

CAD çizimleri çeşitli basit nesnelere birlikte kullanılması ile oluşur. CAD çizimlerinde kullanılan başlıca nesnelere şunlardır:

- Çizgi /Line
- Çoklu Çizgi / Polyline
- Daire / Circle
- Yay / Arc
- Elips
- Metin / Text
- Ölçülendirmeler / Dimensions

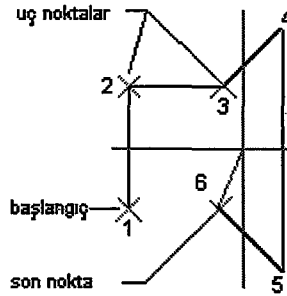
Her bir nesne kendisine ait özelliklere sahiptir. Nesnelere tipleri ve özellikleri bir veritabanında saklanarak gerektiğinde resim olarak bilgisayar ekranında tekrar üretilir. Nesnelere, layer olarak adlandırılan farklı katmanlarda bulunabilirler. Her bir

katman istendiği zaman gizlenerek, katman içindeki nesnelerin görünmez olması sağlanabilir. Nesnelerin çeşitli katmanlara dağıtılması ile, tasarım aşamasında sadeleştirme kolaylığı sağlanır.

Bu projede, robotun tezgahı üzerinde bulunan parçalar, çizgilerden oluşan iki boyutlu CAD vektör resmine dönüştürülüp ölçekli vektör resmi haline getiriliyor. Uygulamadaki CAD dosyası, “taranan”, “kullanıcı” ve “robot” isimli katmanları içerir. Taranan katmanında, tezgahtan alınıp vektörize edilen parçaların vektör resimleri bulunur. Kullanıcı katmanında ise, sistemi kullanan kişi tarafından çizilen nesnelere tutulur. Robot katmanında ise, kullanıcı katmanındaki kompleks nesnelere çizgi nesnelere indirgenerek çizici XYZ robota gönderilir.

4.4.1 Çizgi / line

Çizgi nesnesi, başlangıç ve bitiş noktasının koordinatları ile tanımlanır. Veritabanında toplam altı adet noktalı sayı parametresi ile saklanır. Kullanılan CAD kütüphanesinde CadAddLine (handle, x1, y1, z1, x2, y2, z2) fonksiyonu ile mevcut katmana bir çizgi eklenir. (x1,y1,z1) çizginin başlangıç noktasını, (x2,y2,z2) noktası ise bitiş noktasını ifade eder.



Şekil 4.4 Çizgi nesnesi. Resimde toplam beş adet ayrı çizgi nesnesi mevcut

Aşağıda bitmap kenar resminden elde edilen vektör resmin parçalarını, çizgi nesneleri olarak CAD ortamına aktaran kod parçası görülüyor:

```
StrPcopy(szLayerName, 'Taranan');  
hLayer := CadAddLayer( hDwg, szLayerName, CAD_COLOR_CYAN, 0, CAD_LWEIGHT_DEFAULT );  
CadSetCurLayer( hDwg, hLayer );
```

```

for i := 1 to VectorList[0].vxl do
begin
  x1 := VectorList[i].vx1;
  y1 := VectorList[i].vy1;
  x2 := VectorList[i].vx2;
  y2 := VectorList[i].vy2;
  if not ((x1=x2) and (y1=y2)) then
  begin
    xal := (x1*oran - xkaydir);
    yal := (y1*oran - ykaydir);
    xa2 := (x2*oran - xkaydir);
    ya2 := (y2*oran - ykaydir);
    if (xal>0) and (xal<cerx) and (xa2>0) and (xa2<cerx) and
      (yal>0) and (yal<cery) and (ya2>0) and (ya2<cery) then
      CadAddLine( hDwg, xa1, ya1, 0, xa2, ya2 , 0);
  end;
end;

```

Aşağıdaki kod parçasında ise, CAD ortamındaki çizgi vektörlerin koordinatlarının alınıp XYZ robota iletilmesi işlevi yerine getiriliyor. CadLineGetPoint fonksiyonu ile, aktif katmandaki çizgi nesnelerinin koordinatları alınıp, çalışma tezgahının fiziki boyut limitleri içinde kalıyor ise, XYZ robota gönderilir. Çizgi için sınır testi, çizginin her bir noktasının tezgah içinde kalması ile sağlanır:

```

//cizgiler
if EntType = CAD_ENT_LINE then
begin
  //x1,y1,z1,x2,y2,z2 degerlerini al...
  CadLineGetPoint1(hEnt, @x1, @y1, @z1);
  CadLineGetPoint2(hEnt, @x2, @y2, @z2);
  //Robota gönder...
  if TezgahLimitKontrolLine(x1, y1, SonZ, x2, y2, SonZ) then
  begin
    //z -> kafa asagi hale gelir, SonZ set edilir
    BaslangicaGit(x1,y1);
    NOP := EP_Line(x2, y2, SonZ, 0, 0, 0, 0);
    if (NOP >= 0) then
    begin
      SonX := x2;
      SonY := y2;
      //SonZ := SonZ;
      MemoRPR.Lines.Add('Çizgi başarılı...');
    end;
  end
else
begin

```

```
    MemoRPR.Lines.Add('Çizgi hedefi sınır dışında...');  
end;
```

4.4.2 Çoklu çizgi / polyline

Çoklu çizgi nesnesi, uç uca ekli çizgi ve yaylardan oluşur. Polyline açık veya kapalı olabilir. Kapalı olması halinde, başlangıç ve son noktası birleştirilir.



Şekil 4.5 Polyline nesnesi, çizgi ve yayların birleşmesinden oluşur

Polyline'in CAD veritabanından her bir kenarının koordinatları ile birlikte, bir kenardan diğerine hangi metodla bağlanacağı (Linear, Round, Quad, Qubic, Bezier, Spline) ve çoklu çizginin kapalı veya açık olacağı bilgisi saklanır.

Aşağıdaki kod parçasında, CAD ortamındaki çoklu çizgi nesnesinin robota hareket olarak iletilmesi örneği gösteriliyor. Uygulama çizgi ve yay parçalarını destekliyor:

```
if EntType = CAD_ENT_POLYLINE then  
begin  
    //kenar sayısını al  
    nVers := CadPLineGetNumVers(hEnt);  
  
    //kenarin bağlantı cesidi. Bu uygulama CAD_PLINE_LINEAR (line+arc) destekler  
    Fit := CadPLineGetFit(hEnt);  
  
    for i:=0 to nVers-2 do  
    begin  
        //birinci kenarı al  
        CadPLineGetVer(hEnt, i, @x1, @y1, @z1);  
        //ikinci kenarı al  
        CadPLineGetVer(hEnt, i+1, @x2, @y2, @z2);  
  
        //bulge coefficient: bulge geometride orta ucgenin eğimi  
        Prm := CadPLineGetPrm(hEnt, i);  
  
        //line ve arc'lardan oluşan polygon. Prm: bulge katsayısını içeriyor...  
        if Fit = CAD_PLINE_LINEAR then
```



```

begin
  //noktalar line ile bagli...
  if Prm = 0 then
    begin
      //robot çiz...
      //LineCiz(x1, y1, x2, y2);
    end
  else
    begin
      //Prm: bulge kaysayisina sahip. P1, P2 noktaları mevcut
      //arc'i tanımla ve robota çizdir...
    end;
  end;
end;
end;

```

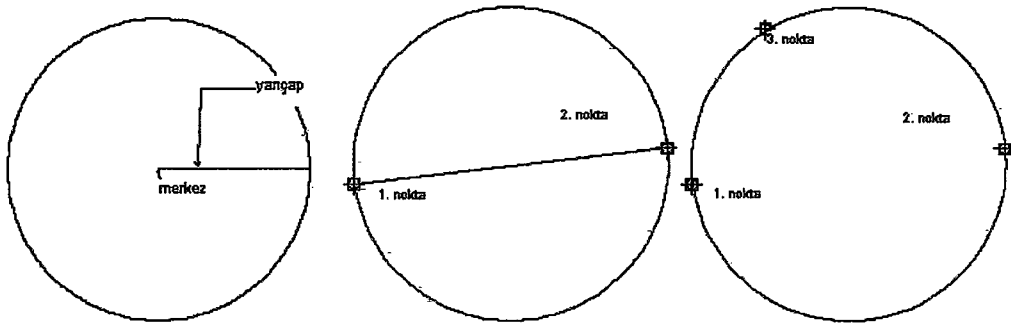
4.4.3 Daire

Daire, CAD ortamında merkez.noktası ve yarıçapı ile tanımlanır.



Şekil 4.6-Dairenin merkez noktası ve yarıçapı

CAD ortamında daire merkez/yarıçap, iki nokta veya üç nokta metodları ile çizilebilir.



Şekil 4.7 Daire merkez/yarıçap, iki nokta ve üç nokta ile tanımlanabilir

Dairenin tezgah sınırları içinde olup olmadığını test etmek için, merkez noktadan itibaren r yarıçap kadar X ve Y boylarındaki ileri koordinatların tezgah içinde olması test edilir. Burada sınır tolerans değeri, tezgahın çerçeve sınırlarında bırakılan fiziki toleransı milimetre cinsinden ifade eder. Aşağıda daire sınır kontrol kodu görülüyor:

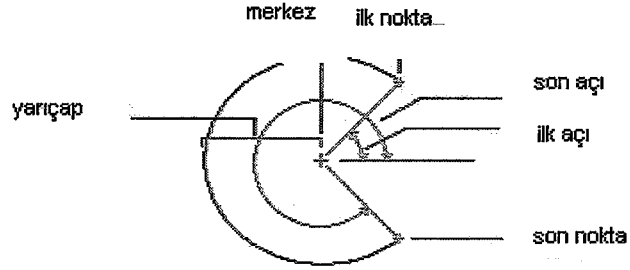
```
r := trunc (sqrt(sqr(X1-X0) + sqr(Y1-Y0))) + SınırTolerans;
//merkezden yarıçap kadar X boyunda ve Y boyunda sapmalara izin ver
a := ((X1+r) <= TezgahX);
b := ((X1-r) >= 0);
c := ((Y1+r) <= TezgahY);
d := ((Y1-r) >= 0);
TezgahLimitKontrolCircle := (a and b and c and d);
```

XYZ robota daire çizdirmek üzere, robot kafasının bulunduğu noktadan itibaren çizilecek çembere en yakın nokta hesaplanır. Sonrasında dairenin tezgah sınırları içinde olması sağlanması yapılır ve hareket kartına gerekli komut gönderilir. Aşağıda bu işlevi yerine getiren kod görülüyor:

```
if ArcType = CAD_CIRCLE then
begin
//merkez koordinatlarını ve yarıçapı cad ortamından al
CadArcGetCenter(hEnt, @x1, @y1, @z1);
r := CadArcGetRadius(hEnt);
//(SonX, SonY) noktasından çembere en yakın noktayı bul
m := sqrt(sqr(x1-SonX)+sqr(y1-SonY));
p := (m - r);
//kafanın çembere en yakın noktası (x2,y2)
x2 := SonX + ((p * (x1-SonX)) / m);
y2 := SonY + ((p * (y1-SonY)) / m);
if TezgahLimitKontrolCircle(x2, y2, x1, y1) then
begin
// (x2, y2) noktasına kafa yukarıda git
BaslangicaGit(x2,y2);
NOP := EP_CircleXY(x1, y1, 0, 0);
if NOP >= 0 then
begin
MemoRPR.Lines.Add('Daire başarılı...');
end;
end
else
begin
MemoRPR.Lines.Add('Daire kısmen sınır dışında...');
end;
end
end
```

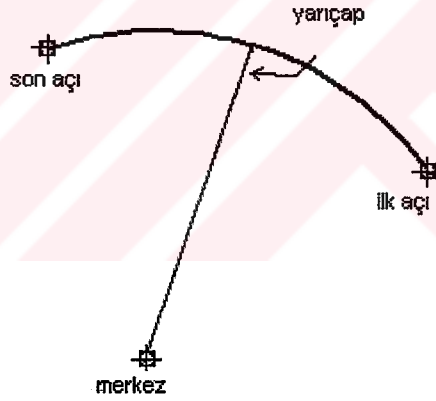
4.4.4 Yay parçaları

Dairesel yay parçası, merkez, yarıçap, başlangıç açısı ve bitiş açısı ile tanımlanır.



Şekil 4.8 Yay parçasının CAD ortamındaki unsurları

CAD ortamında yay parçası çizimi, merkez, yarıçap, başlangıç ve bitiş açısı ile veya üç nokta ile yapılır. Aşağıdaki şekillerde bu iki metod açıklanıyor:



Şekil 4.9 Merkez nokta, yarıçap, başlangıç ve bitiş açısı ile yay çizimi



Şekil 4.10 Üç nokta ile yay çizimi

Yay parçalarının tezgah limitleri içinde kaldığının test edilmesi için, başlangıç açısından bitiş açısına kadar olan bölge n adet parçaya bölünerek, yay üzerindeki her bir noktanın tezgah limitleri içinde kaldığı test edilir. Aşağıda bununla ilgili uygulamadaki kod parçası görülüyor:

```
//(X0, Y0) baslangic noktasi, (X1, Y1) merkez nokta, Theta: Aci degeri
//yaricapi hesapla, 5 mm ilave toleransla
r := (sqrt(sqr(X1-X0) + sqr(Y1-Y0)));

//Cember üzerindeki N adet noktanin calisma sahasi icinde oldugu test edilir
N := 10;
aciadim := (Theta/10) * 0.0174532925; //derece->radyan
ilkaci := arctan2(Y0-Y1, X0-X1); //radyan
limitcheck := True;
for i := 0 to N do
begin
aci := ilkaci + (i * aciadim);
X := (X1 + r * cos(aci));
Y := round(Y1 + r * sin(aci));
//(X, Y) noktasi ic bölgede olmalidir
a := (X>=0) and (X<=TezgahX-SinirTolerans);
b := (Y>=0) and (Y<=TezgahY-SinirTolerans);
c := (X>=0) and (X<=TezgahX-SinirTolerans);
d := (Y>=0) and (Y<=TezgahY-SinirTolerans);
limitcheck := limitcheck and (a and b and c and d);
if not limitcheck then break;
end;
```

Aşağıdaki kod parçası ise, yay parçasını, tezgah limit kontrolü sonrasında XYZ robota çizdirilmek üzere gönderir:

```
if ArcType = CAD_ARC then
begin
//merkez koordinatlarini ve yaricapi cad ortamindan al
CadArcGetCenter(hEnt, @x1, @y1, @z1);
CadArcGetStartPt(hEnt, @x2, @y2, @z2);
CadArcGetEndPt(hEnt, @MSX, @MSY, @MSZ);
r := CadArcGetRadius(hEnt);
aci := CadArcGetAngle(hEnt) * 57.2957795785522989; //radyan -> derece

//arc'in tezgah icinde kaldigini ispatlayarak ciz
if TezgahLimitKontrolArcTheta(x2, y2, x1, y1, aci, SX, SY) then
begin
//(x2, y2) noktasina kafa yukarda git
BaslangicaGit(x2,y2);
NOP := EP_ArcThetaXY(x1, y1, aci, 0);
```

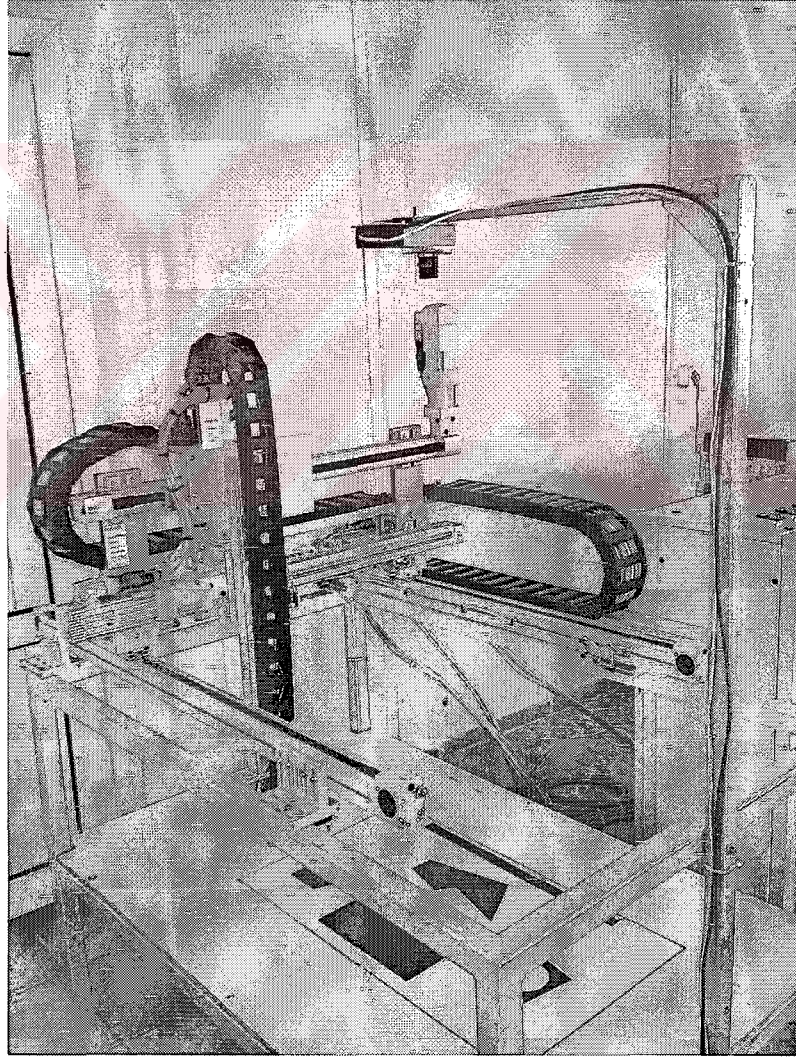
```
if NOP >= 0 then
begin
//Varis noktasi kontrol fonksiyonda donecegi gibi,
//CAD ortamından da alınabilir alternatif olarak
SonX := MSX;
SonY := MSY;
MemoRPR.Lines.Add('YayAciXY başarılı...');
end;
end
else
begin
MemoRPR.Lines.Add('YayAciXY kısmen sınır dışında...');
end;
```

4.5 Nesnelerin Alt Nesnelere Parçalanması

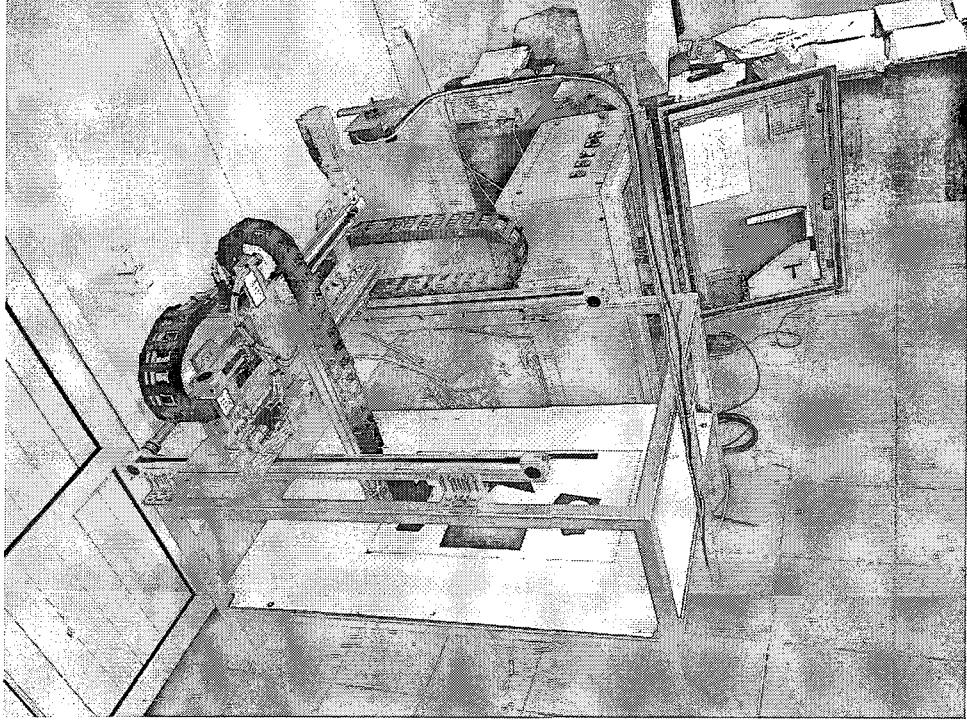
CAD ortamındaki kompleks nesnelere, kompleks enterpolasyonların yapılması yerine çizgi parçalarına kadar basit nesnelere indirgenerek XYZ robota çizdirilebilir. Mevcut çizici robot uygulamasında, “kullanıcı” katmanında kullanıcı tarafından çizilen nesnelere, “robot” isimli ayrı bir katmana taşınarak çizgi seviyesine kadar alt nesnelere indirgenir. Sonrasında “robot” katmanındaki her bir çizgi nesnesi çizici robota gönderilir.

BÖLÜM 5. KARTEZYEN ROBOT

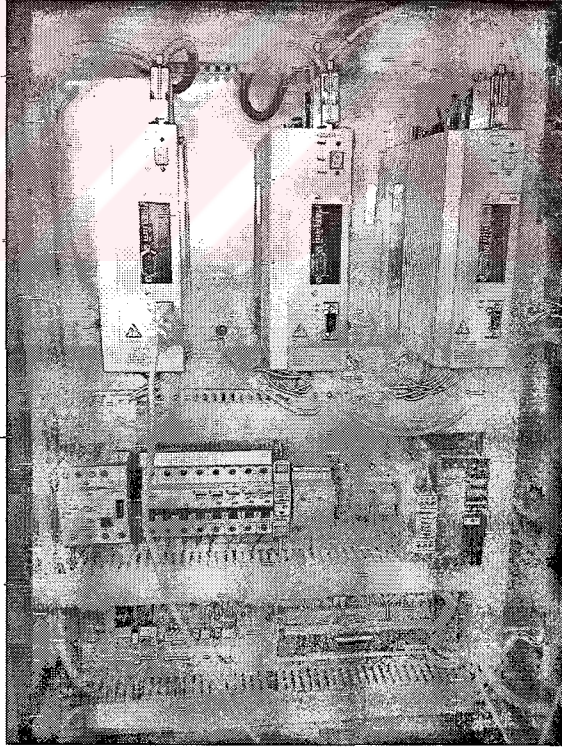
Kartezyen robot, doğrusal eksenler, servo motorlar, servo sürücüler ve hareket kartından oluşur. Doğrusal eksen olarak Festo DGE-25-750-ZR-RF ve FDG-25-450-ZR-RF kayışlı eksenleri, servo motor olarak, Festo MTR-AC-70-3S-GA ve MTR-AC-55-3S-GB motorları kullanılmıştır. Servo motorları sürmek üzere sistemde üç adet SEC-AC-305 servo sürücü bulunur.



Şekil 5.1. XYZ kartezyen robot yakın görünüm



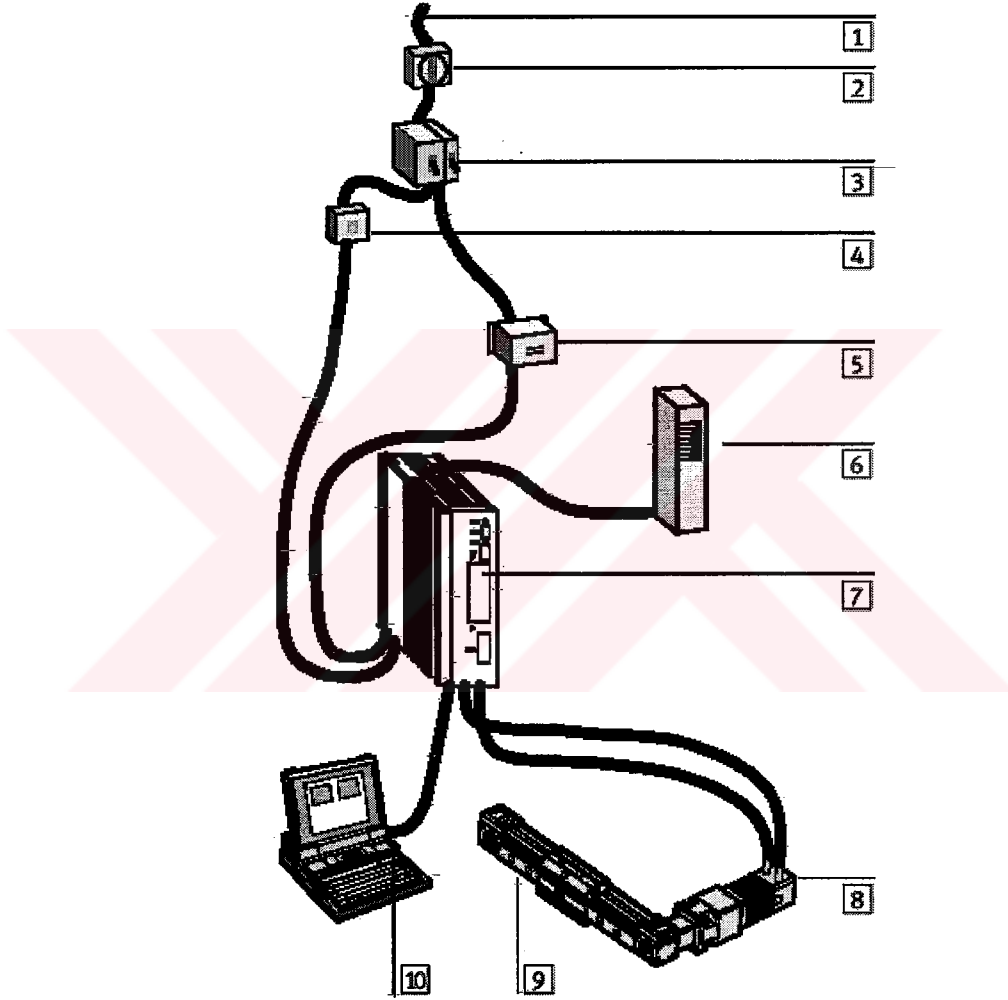
Şekil 5.2. XYZ kartezyen robot ve kontrol paneli



Şekil 5.3. Kontrol paneli yakın görünüm

5.1. Servo Sürücüler

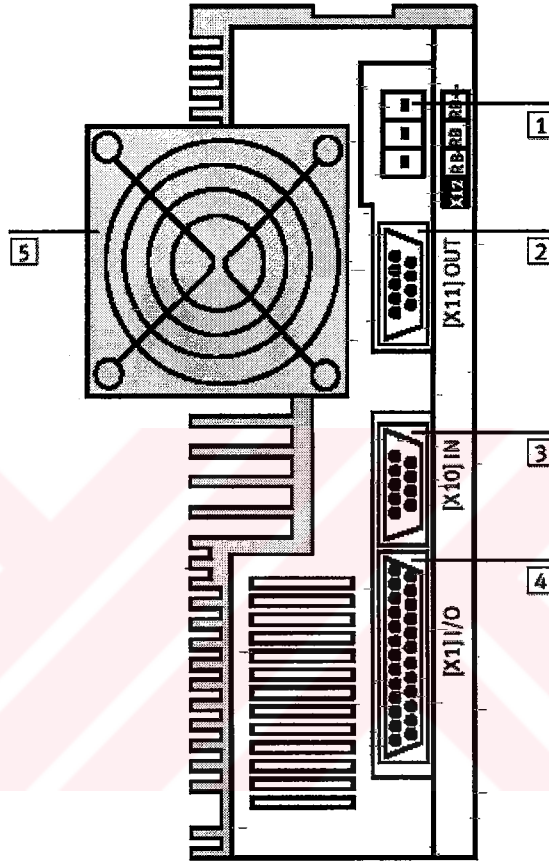
Projede kullanılan XYZ kartezyen robot sistemi, servo sürücüler, servo motorlar ve mekanik eksenlerden oluşmuştur. Aşağıdaki şekilde sistemin parçaları her bir eksen için numaralı olarak gösterilmiştir. (1) Güç bağlantısı (2) Anahtar (3) Sigortalar (4) Şase kontağı (5) 24 V DC kaynak (6) Harici direnç (7) SEC-AC 305 Servo sürücü (8) Motor ve resolver (9) Mekanik eksen (10) Parametre ayarları için PC



Şekil 5.4. Servo sistemin parçaları [9]

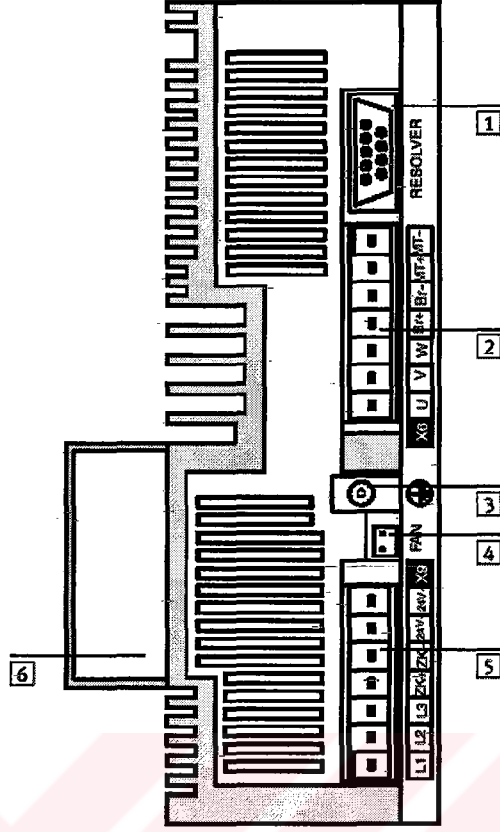
5.2. SEC AC 305 Servo Sürücüsü

Aşağıdaki şekilde SEC AC 305 servo sürücünün üstten görünümü verilmiştir. (1) X12 Harici direnç (2) X11 Artımlı enkoder çıkışı (3) X10 Artımlı enkoder girişi (4) Digital/analog giriş-çıkışlar (5) Soğutma fanı



Şekil 5.5. Servo sürücü, üstten görünüm [9]

Bir sonraki resimde servo sürücünün alttan görünümü veriliyor. (1) X2 Resolver girişi (2) X6 Motora güç besleme, sıcaklık sensörü, fren (3) Şase bağlantısı (4) Soğutma fanı bağlantısı (5) X9 Güç kaynağı bağlantısı (6) Soğutma amaçlı hava kanalı,

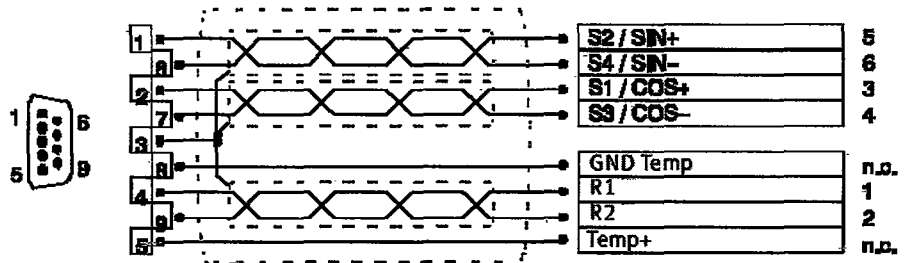


Şekil 5.6. Servo sürücü, alttan görünüm [9]

5.3. Servo Sürücü Bağlantıları

5.3.1 Resolver bağlantısı (X2)

Servo motorlar üzerinde bulunan resolver bağlantıları aşağıda şekilde gösterilmiştir:



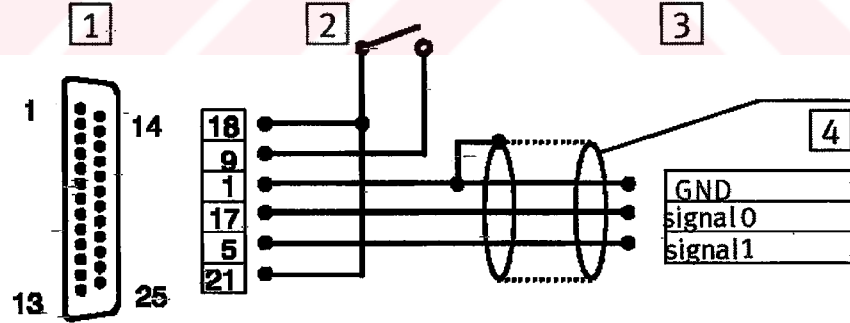
Şekil 5.7. Servo sürücü resolver bağlantıları

Tablo 5.1. Servo sürücü resolver bağlantı tablosu

Pin No	Sembol	Açıklama
1	SIN+	Resolver sinüs sinyali
6	SIN-	Sinüs sinyali tersi
2	COS+	Kosinüs sinyali
7	COS-	Kosinüs sinyali tersi
3	AGND	Analog şase
8	AGND	Analog şase
4	REF+	Resolver sinyal
9	REF-	Resolver sinyal tersi
5	MOT-TEMP	Motor ısı sensörü

5.3.2 Digital/analog giriş çıkışlar (X1)

Servo-sürücü üzerinde, dış dünya ile haberleşme amaçlı olarak, X1 soketi üzerinde çeşitli analog ve digital giriş/çıkışlar bulunur.



Şekil 5.8. Analog giriş/çıkış soketi (1)Soket (2) Sürücü hazır (3) BNC (4) Ekranlama

Tablo 5.2. Servo sürücü analog giriş/çıkış soketi bağlantıları

Pin	Sembol	Değer	Açıklama
1	Schirm	0V	Ekranlama
14	Schirm	0V	Ekranlama
2	Ain0	+/- 10V Ri=20K	Analog giriş 0
15	Ain0/		Analog giriş 0 referansı
3	Ain1		Analog giriş 1
16	Ain1/		Analog giriş 1 referansı
4	+VREF	+10V	Referans voltaj çıkışı
17	AMON0	+/- 10V	Analog monitör çıkışı 0
5	AMON1	+/- 10V	Analog monitör çıkışı 1
18	+24VDC	+24V	24V DC güç çıkışı, korumasız
6	GND	GND	Dijital şase
19	Din0	POS-Bit 0	Pozisyon seçme 0
7	Din1	POS-Bit 1	Pozisyon seçme 1
20	Din2	POS-Bit 2	Pozisyon seçme 2
8	Din3	POS-Bit 3	Pozisyon seçme 3
21	Din4	ENA-PWR	Son durum hazır
9	Din5	ENA-CL	Sürücü hazır (Servo On)
22	Din6	END-0	Sınır sensörü negatif
10	Din7	END-1	Sınır sensörü pozitif
23	Din8	START	Pozisyon başla
11	Din9	SAMPLE	Yüksek hız giriş
24	Dout 0	READY	İşleme hazır
12	Dout1	PRG-1	Programlanabilir çıkış
25	Dout2	PRG-2	Programlanabilir çıkış
13	Dout3	PRG-3	Programlanabilir çıkış

5.3.3 Artımlı enkoder giriři (X10)

Artımlı enkoder giriři, master/slave operasyon moduna alınan servo sürücülere, hareket kartı üzerinden enkoder formunda darbeler göndermek üzere kullanılmıřtır. Standart 9 pinli soket ile darbe/pulse giriřleri yapılmıřtır.

Tablo 5.3. Servo sürücü artımlı enkoder giriři

Pin No	Sembol	Açıklama
1	A	Sinyal giriři A
6	A/	Sinyal giriři A deęil
2	B	Sinyal giriři B
7	B/	Sinyal giriři B deęil
3	C	Sinyal giriři C
8	C/	Sinyal giriři C deęil
4	Schirm	Ekranlama
9	GND	Dijital řase
5	+5VDC	+5V DC kaynak, harici sensör için

5.3.4 Artımlı enkoder çıkıřı (X11)

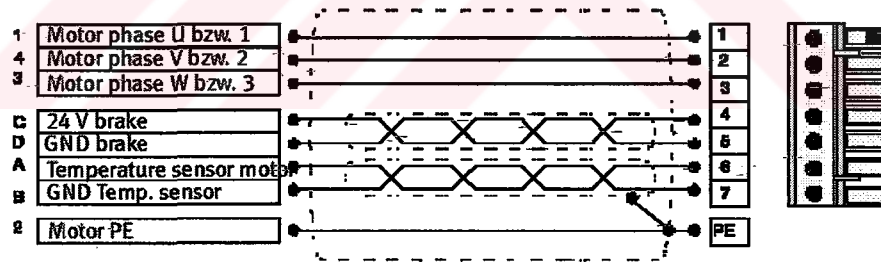
Artımlı enkoder çıkıřı, master/slave operasyon modunda çalıřan servo sürücüden, pozisyonlama bilgisini darbe/pulse olarak alıp hareket kartına geri besleme sağlamak üzere kullanılmıřtır. Darbe çıkıřı standart 9 pinli soket ile yapılmıřtır.

Tablo 5.4. Servo sürücü artımlı enkoder çıkışı

Pin No	Sembol	Açıklama
1	A	Sinyal girişi A
6	A/	Sinyal girişi A değil
2	B	Sinyal girişi B
7	B/	Sinyal girişi B değil
3	C	Sinyal girişi C
8	C/	Sinyal girişi C değil
4	Schirm	Ekranlama
9	GND	Dijital şase
5	+5VDC	+5V DC kaynak, harici sensör için

5.3.5 Motor güç bağlantısı (X6)

Servo sürücü üzerinden motora güç beslemek üzere (X6) soketi kullanılmıştır.



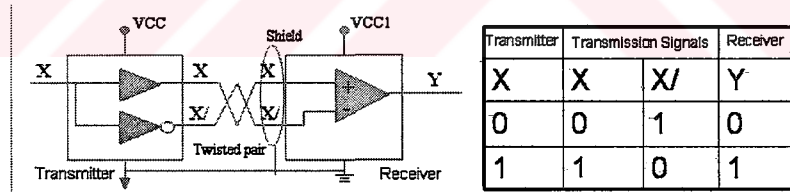
Şekil 5.9. Servo motor güç bağlantısı

Tablo 5.5. Servo sürücü motor güç bağlantısı

Pin No	Sembol	Açıklama
1	U	Motor bağlantısı, faz 1
2	V	Motor bağlantısı, faz 2
3	W	Motor bağlantısı, faz 3
4	Br+	24V Fren
5	Br-	24V Fren şase
6	MT+	Motor sıcaklık sensörü
7	MT-	Sıcaklık sensörü şase

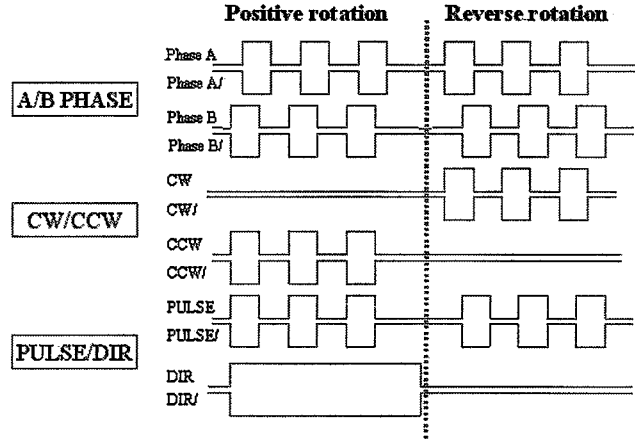
5.4. Diferansiyel Sinyal Aktarma ve Darbe Tipleri

Hareket komutlarının hareket kartından servo sürücülere aktarılması için darbeler kullanılmaktadır. Her bir hareket komutuna karşılık olarak üretilen darbe dizilerinin aktarım sırasında bozulmaması için diferansiyel aktarma metodu kullanılarak, sinyal ile birlikte sinyalin tersi de aynı anda servo sürücüye aktarılır.



Şekil 5.10 Diferansiyel sinyal aktarma metodu [10]

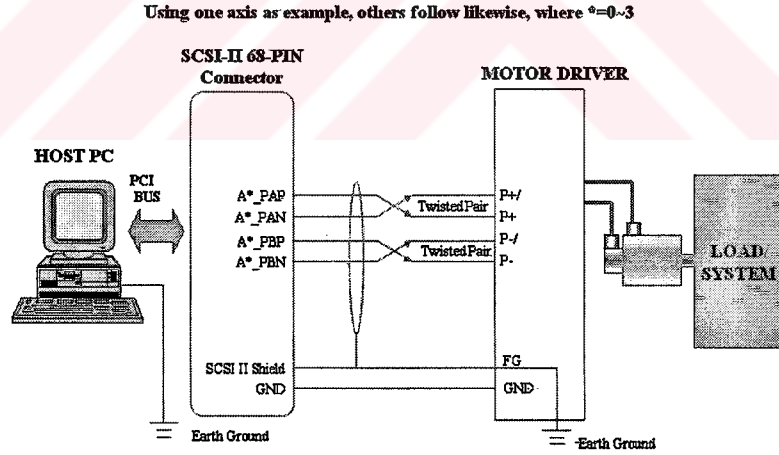
Bu projede, hareket komutlarını servo sürücülere iletmek üzere Advantech PCI 1242 modeli, darbe/pulse tipi hareket kartı kullanılmıştır. Bu hareket kartı, darbe giriş ve çıkışlarında üç tip darbe modunu desteklemektedir: pulse/direction (darbe/yön), ileri/geri pulsler (CW/CCW) veya A/B/Phase (enkoder formunda pulsler).



Şekil 5.11 Desteklenen darbe giriş/çıkış formatları [10]

5.5. Hareket Kartı ve Servo Sürücü Bağlantıları

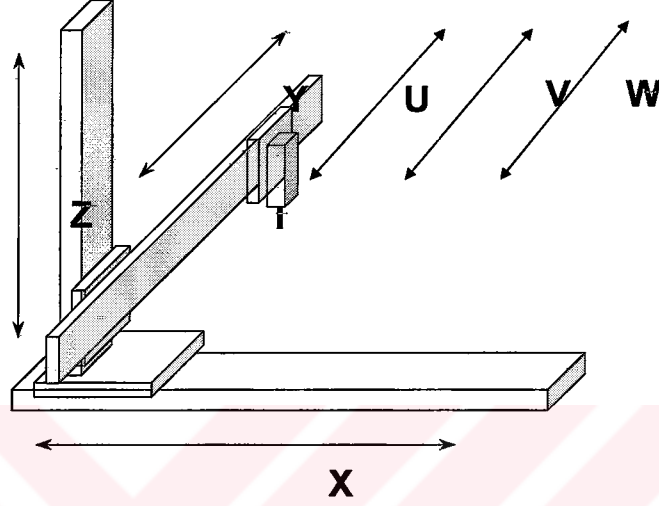
Projede kullanılan SEC-AC 305 servo sürücüleri enkoder formunda darbeler ile haberleştiğinden, hareket kartı ile olan darbe bağlantılarında A, A/, B, B/ darbe kanalları her bir eksen için ilgili servo sürüçülere bağlanmıştır.



Şekil 5.12 Hareket kartı - servo sürücü bağlantıları, bir eksen için [10]

5.6. Eksen Yapısı

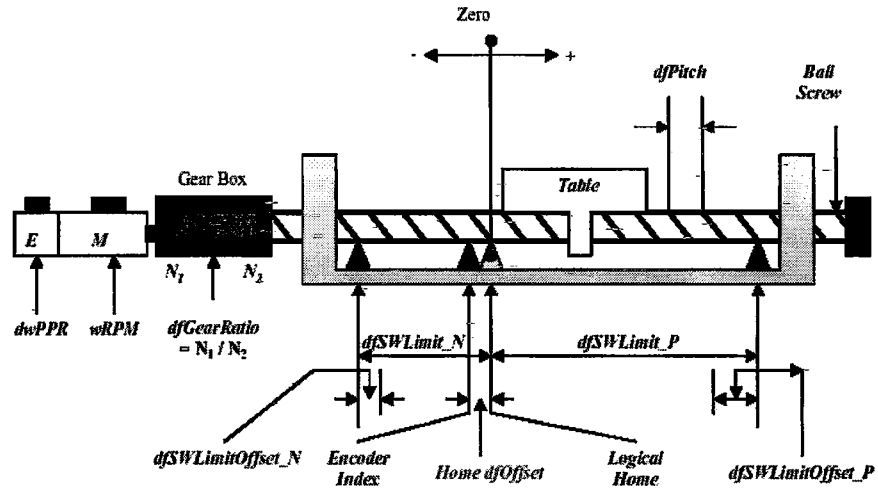
Projede kullanılan PCI 1242 tipi hareket kartı, üç adet ana eksen (X-Y-Z) ve üç adet yardımcı ekseni (U-V-W) kontrol etmek üzere tasarlanmıştır. Hareket komutları eksenlere mutlak veya göreceli koordinat sistemine göre gönderilebilir.



Şekil 5.13 Hareket kartının eksen yapısı

5.7. Mekanik Parametre Tanımları

Hareket kartının çalışacağı mekanik sisteme ait motor ve çalışma tezgahı parametrelerini sisteme girmek üzere EP_MAC_PARAM kayıt yapısı kullanılır.



Şekil 5.14 Mekanik parametre ayarları [10]

```

EP_MAC_PARAM = record
    wAxisDir : word
    wRPM : word;
    dwPPR : integer;
    dfPitch : double;
    dfGearRatio : double;
    dfSWLimit_P : double;
    dfSWLimit_N : double;
    dfSWLimitOffset_P : double;
    dfSWLimitOffset_N : double;
    wPulseMode : word;
    wPulseWidth : word;
    wCommandMode : word;
    wPaddle : word;
    wHWLimitMode_P : word;
    wHWLimitMode_N : word;
end;

```

wAxisDir: Eksen yönü. “1” değerine set edilirse, eksen ters yönde hareket eder

wRPM: Eksen hızı, dakikada tur sayısı cinsinden

dfPitch: Eksenin bir turda alacağı mesafe

dfGearRatio: Dişli sistemi ile motor tur sayısı azaltılıyor ise, kaldıraç oranı girilir

dfSWLimit_P: Pozitif yöndeki sınır. (Yazılım ile kontrol)

dfSWLimit_N: Negatif yöndeki sınır. (Yazılım ile kontrol)

dwSWLimitOffset_P: Pozitif yönde tezgah sınır bölgesi (Yazılım ile kontrol)

dwSWLimitOffset_N: Negatif yönde tezgah sınır bölgesi (Yazılım ile kontrol)

wPulseMode: 0: Pulse/Direction 1: CW/CCW 2: A/B Phase darbe modu

wPulseWidth: Serve sürücünün özelliğine göre ns cinsinden darbe genişliği

wHWLimitMode_P: 0: Normalde açık kontak 1: normalde kapalı kontak sınır sensörü, pozitif yönde

wHWLimitMode_N: 0: Normalde açık kontak 1: normalde kapalı kontak sınır sensörü, negatif yönde

5.8. Home Parametre Tanımları

XYZ robotun her bir ekseninin sıfır noktasına gitmesini sağlamak üzere eksenlerin uç noktalarında sensörler bulunur. Sıfır noktası parametre tanımları ile bu sensörlerin davranışları tanımlanır. Sensörler normalde açık veya normalde kapalı kontaklar şeklinde olabilir.

```
EP_HOME_CONFIG = record
  wMode : word;
  wDir : word;
  wSensorMode : word;
  wPaddle1 : word;
  nIndexCount : integer;
  nPaddle2 : integer;
  dfAccDecTime : double;
  dfHighSpeed : double;
  dfLowSpeed : double;
  dfOffset : double;
end;
```

wMode: Home pozisyon modu (0..13)

wDir: Hareket yönü 0: Pozitif yön 1: Negatif yön

wSensorMode: 0: Normalde açık kontak 1: Normalde kapalı kontak

nIndexCount: Enkoder indeks (Phase Z)

dfAccDecTime: ms cinsinden ivmelenme zamanı

dfHighSpeed: Hareket hızı mm/s

dfLowSpeed: Harekete başlama hızı mm/s

dfOffset: Tezgahta bırakılan tolerans mesafesi

5.9. Enkoder Parametre Tanımları

Hareket kartının darbe giriş ve çıkışlarında kullanılacak olan darbe şekillerinin tanımlanması için EP_ENCODER_CONFIG yapısı kullanılır.

```
EP_ENCODER_CONFIG = record
  wType : word;
  wAInverse : word;
  wBInverse : word;
  wCInverse : word;
  wABSwap : word;
  wInputRate : word;
  paddle : array [0..1] of word;
end;
```

wType: Enkoder formatı 0: A/B/Phase 1: CW/CCW 2: Pulse/Direction

wAInverse: 1: A darbe kanalı ters 0: A darbe kanalı düz

wBInverse: 1: B darbe kanalı ters 0: B darbe kanalı düz

wCInverse: 1: C darbe kanalı ters 0: C darbe kanalı düz

wABSwap: A ve B kanalları yer değiştirmesi

5.10. Koordinat Sistemi

Kullanılan hareket kartı mutlak veya artımlı koordinatlar ile çalışabilir. Hareket kartı arayüzü içinde, koordinat sistemi ile ilgili olarak şu fonksiyonlar sağlanmıştır:

EP_SetAbs(): Absolute (Mutlak) koordinat sistemi

EP_SetInc(): Incremental (Artımlı) koordinat sistemi

EP_GetCoordType(): Koordinat tipinin sorgulanması

EP_SetUnit(): Ölçüm birimi olarak mm veya inc seçilmesi

EP_GetUnit(): Mevcut ölçü biriminin sorgulanması

EP_GetUnitPos(): Eksenlerin pozisyonlarını mm cinsinden oku

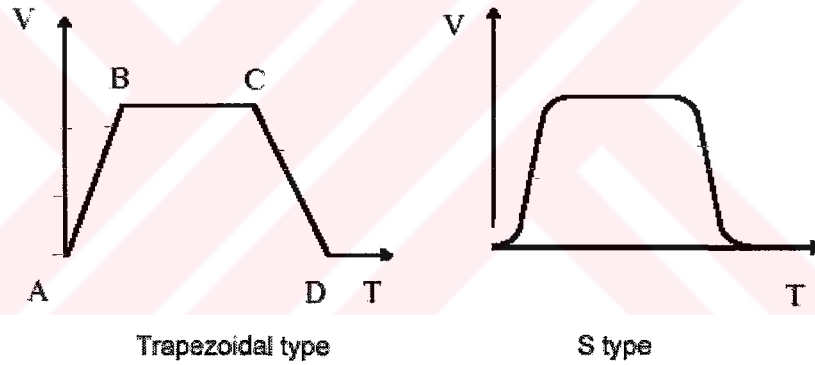
EP_GetPulsePos(): Eksenlerin pozisyonlarını darbe cinsinden oku

5.11. Yörünge Planlama

Hareket kartı ile gelen MCFL isimli fonksiyon kütüphanesi, doğrusal ve dairesel enterpolasyonları sağlar. Hareket fonksiyonları çağrılmadan önce kullanıcı, harekete ait tanımlama bilgilerini sağlamalıdır. Bu parametreler ivmelenme tipi (S veya Trapezoid), ivmelenme zamanları ve hareketin hızı ve hareketle ilgili diğer koordinat bilgileridir.

5.12. Yörünge Planlama: Çizgi, Yay, Çember

Hareket komutu karta gönderilmeden önce, EP_MOTION_CONFIG yapısı kullanılarak harekete ait parametreler belirlenir. Hareketin ivmelenme tipi (T/S), ivmelenme zamanları ve hareketin hızı, hareket konfigürasyon parametrelerini teşkil eder.



Şekil 5.15 İvme seçenekleri

Uygulama içerisinde, hareket konfigürasyonu işlevi, aşağıdaki SetMotionCfg fonksiyonuna atanmıştır:

```
procedure TForm1.SetMotionCfg(AccType, DecType, AccTime, DccTime, FeedSpd: integer) ;
var
  stMotionConfig : EP_MOTION_CONFIG;
begin
  stMotionConfig.cAccType := AccType;      // T/S
  stMotionConfig.cDecType := DecType;      // T/S
  stMotionConfig.dfAccTime := AccTime;     //acceleration time(ms)
  stMotionConfig.dfDecTime := DccTime;     //deceleration time(ms)
  stMotionConfig.dfFeedSpeed := FeedSpd;   //mm/sec, eksen hızı
  EP_SetMotionConfig (stMotionConfig, 0);
```

end;

Lineer Hareket Komutları: EP_Line(), EP_LineX(), EP_LineY(), EP_LineZ()

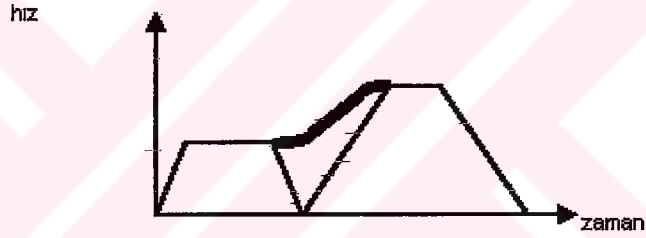
Yay Komutları: EP_ArcXYZ(), EP_ArcXY(), EP_ArcYZ()

Daire Komutları: EP_CircleXY(), EP_CircleYZ(), EP_CircleZX()

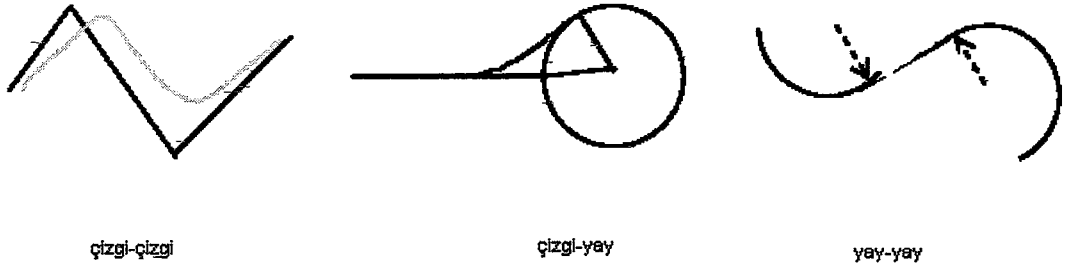
Yardımcı Eksen Lineer Hareket Komutları: EP_LineU, EP_LineV, EP_LineW

5.13. Gelişmiş Yörünge Planlama: Blending

Blending özelliği ile, hareket kartı, hareket komutları arasındaki geçişlerde yumuşatma uygular. Böylece keskin dönüş noktalarında, hareket kartı teğet hızı sabit tutmak şartıyla, köşe dönüşlerinde mekanizmanın hareketini yumuşatır. EP_EnableBlend() fonksiyonu ile bu özellik aktive edilir.



Şekil 5.16 Blending modunda hız grafiği



Şekil 5.17 Blending modunda geçişler

BÖLÜM 6. UYGULAMA ÖRNEKLERİ

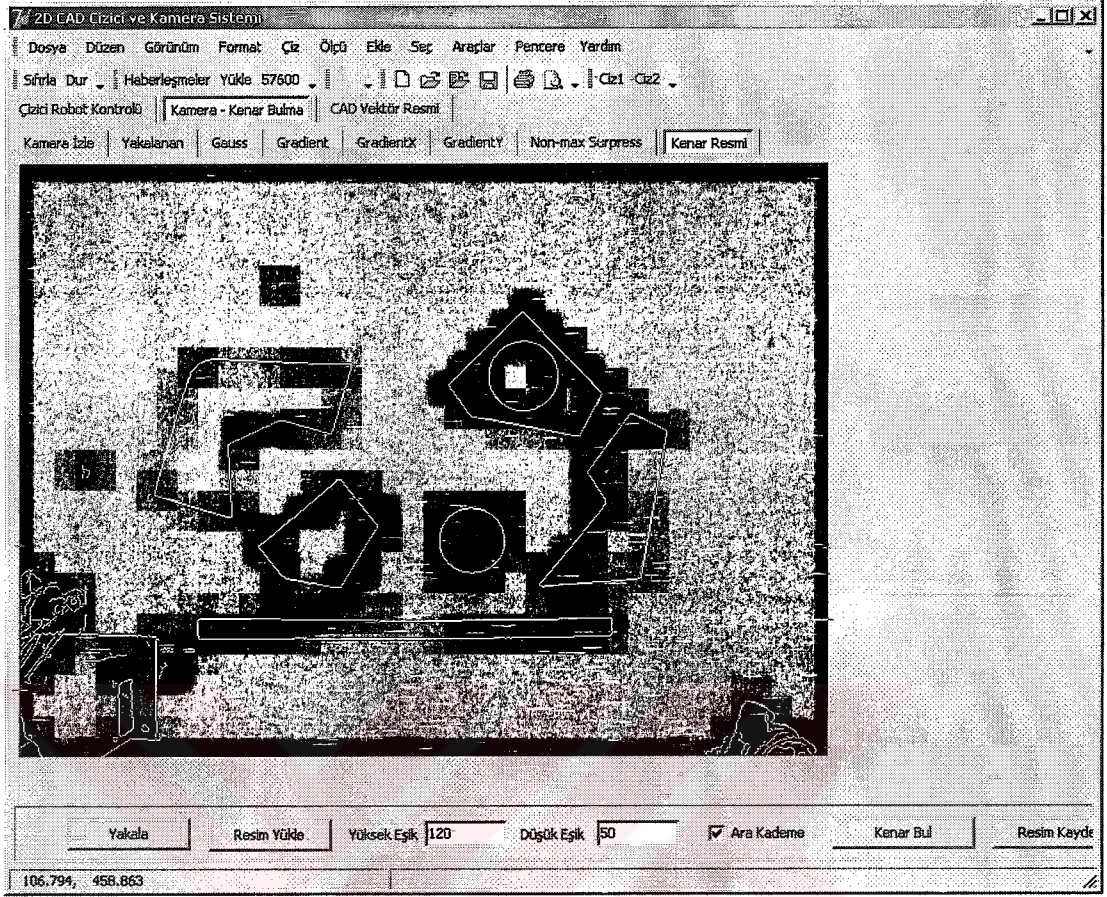
6.1. Kenar Taraması

Problem: Tezgah üzerindeki parçaların kenarlarının taranması

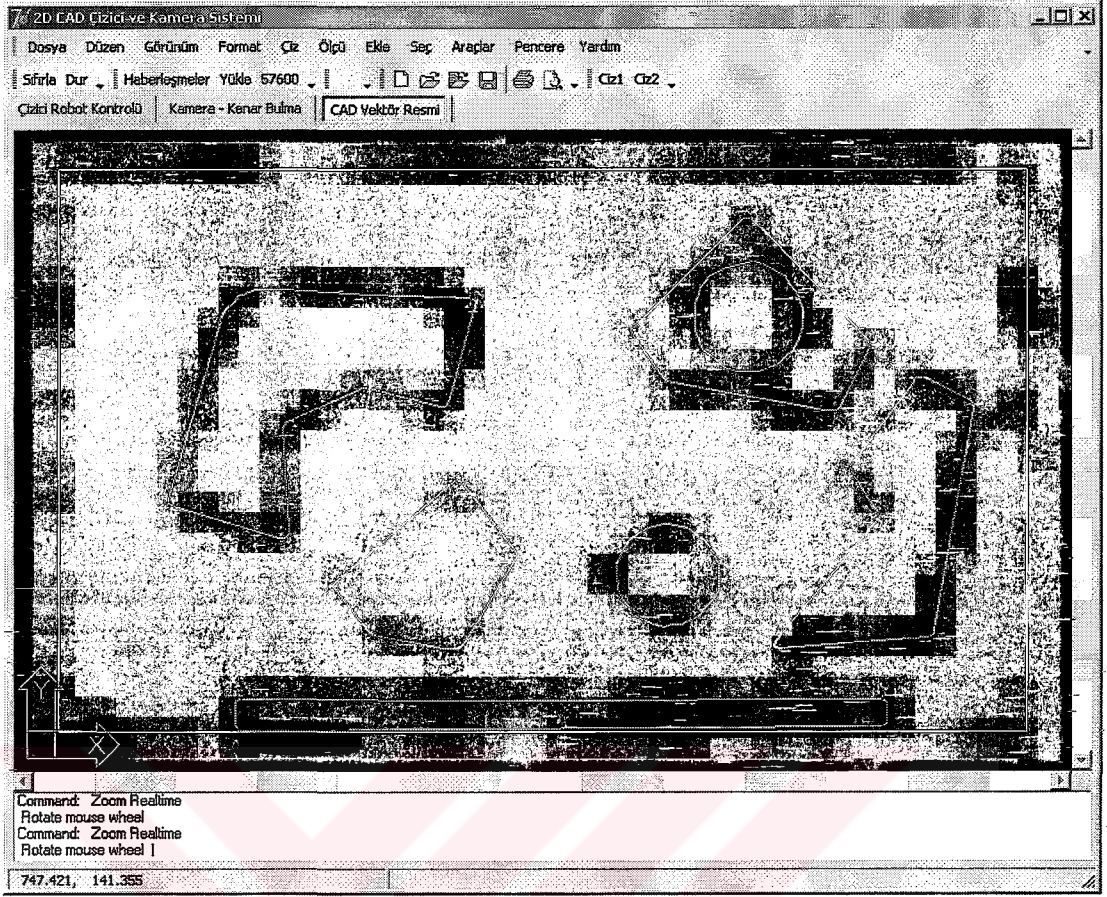
Bu uygulamada, tezgah üzerindeki parçalar kamera ile elde edilen resim kullanılarak vektör resmine dönüştürülür ve tasarım ortamına ölçekli olarak alınır. Kullanıcı, ana parçaların üzerinde çizilecek nesneleri tasarlayarak kartezyen robota çizdirmek üzere hareket kartına tasarımdan üretilen hareket komutlarını gönderir.



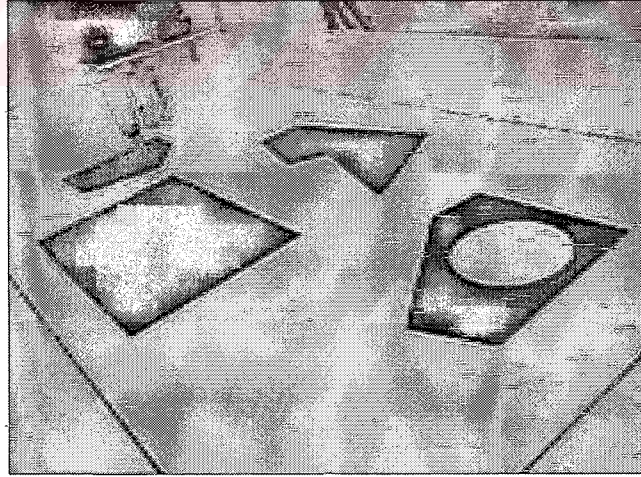
Şekil 6.1. Kameradan görüntü yakalama



Şekil 6.2: Kenar resmi eldesi



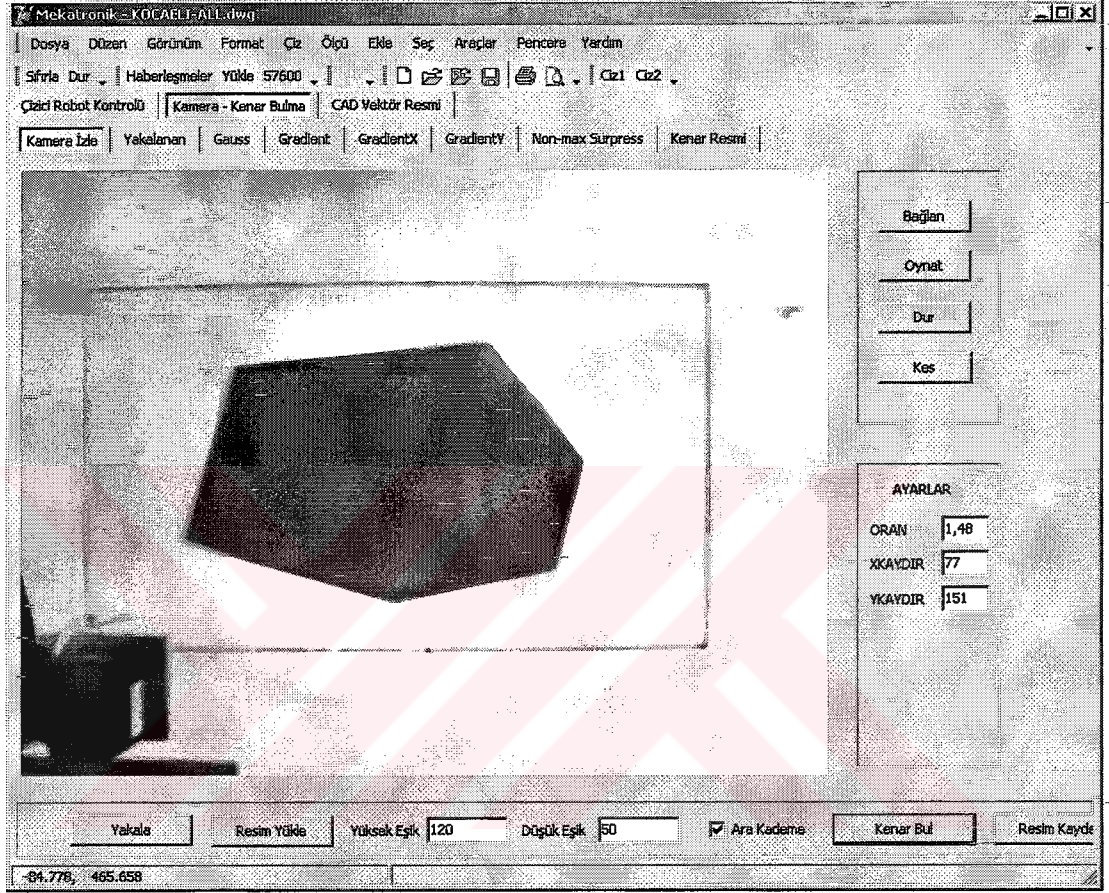
Şekil 6.3. Vektör resmi, CAD-CAM ortamı



Şekil 6.4. Örnek kenar taraması

6.2. Ana Parçadan Harf Kesme

Problem: Tezgah üzerindeki ham haldeki ana parça üzerine kesilecek harflerin yerleştirilmesi ve ana parçadan kesilmesi



Şekil 6.5. Ham parçanın kameradan okunması

İşlenecek ana parçanın kameradan taranıp tasarım ortamına vektör resmi olarak aktarılmasından sonra, kullanıcı tarafından ana parçadan çıkartılacak olan harfler gövde üzerine yerleştirilerek kesici robota gönderilir.



Şekil 6.6. Harflerin ana parça üzerine yerleştirilmesi

BÖLÜM 7. SONUÇ VE ÖNERİLER

Bu çalışmada, iki boyutlu kartezyen kesme sistemlerine kamera desteği ile işlenecek ham parçaları tasarım ortamına aktarma ve işlenmemiş parça üzerinde parça programlama amacı gerçekleştirilmiştir. Bu tip sistemlerin kurulmasında, ana parçadan elde edilecek vektör resminin kenar ve köşelerde gerekli incelik ve doğruluğu sağlaması için ön şart, kamera ve lens sisteminin, tezgah üzerindeki görüntüyü hatasız bir şekilde bilgisayar ortamına aktarabilmesidir. Kamera ve lens hatalarından kaynaklanabilecek bozulmaların en aza indirilmesi için, görüntünün mümkün olduğu kadar yakın mesafeden alınması, resimdeki doğrusal olmama hatalarını giderebilecek seviyede, robot görme amaçlı üretilmiş endüstriyel kalitede lenslerin kullanılması gereklidir. Görüntü alınacak bölgenin ışıklandırılması ve gölgelerin en aza indirilmesi de önemli bir fiziki etkidir. Görüntünün bilgisayar ortamına aktarılması sonrasında, Gauss ve benzeri gürültü eleme filtrelerinin kullanılması, kenar ve köşelerin yumuşatılmasını ve yerlerinin kaymasını netice vereceğinden, bu tip yumuşatma filtreleri, o anki hassasiyet gereksinimine göre opsiyon olarak bırakılmalıdır. Yumuşatma filtrelerinin devre dışı bırakılması durumunda elde edilecek kenar resmi oldukça parazitli olacağından, siyah-beyaz resim kalitesi ve zıtlığı, gerekli ışıklandırma şartları sağlanarak çok iyi seviyede elde edilmelidir.

Mekanik sistemin kurulması aşamasında kullanılacak olan doğrusal eksen, servo motorlar, servo sürücüler ve hareket kartının açık mimari yapıda olması, endüstriyel standartları sağlaması ve darbe tipi olarak mümkün olan tüm formatları (pulse/direction, CW/CCW, A/B/Phase) desteklemesi, istenilen eksenlerde çizgi, yay parçası, daire, elips enterpolasyonlarını sağlaması, gerekli sayıda yardımcı eksenleri desteklemesi, tezgahın mekanik yapısı ile uyumlu uç sensörleri desteklemesi ve mekanik aksamla ilgili güç gereksinimi şartlarını sağlaması gereklidir.

CAD-CAM tasarım-imalat ortamının ise, endüstriyel CAD standartlarını desteklemesi, DXF - DWG gibi yaygın olarak kullanılan-mimari tasarım formatlarına erişebilmesi, en alt seviyede çizgi, yay parçası, daire gibi basit nesnelere kullanıcıya sağlanması ve bu nesnelere hareket komutları şeklinde hareket kartına aktarabilecek seviyede açık mimariye sahip olması gerekir.



KAYNAKLAR

1. KOÜ - İşaret ve Görüntü İşleme Labratuarı, (<http://mf.kou.edu.tr/elohab/ipi>) Görüntü İşleme Ders Notları
2. FISHER, R., PERKINS, S., WALKER, A., WOLFART, E., Hypermedia Image Processing Reference, (HIPR2), <http://homepages.inf.ed.ac.uk/rbf/HIPR2> , 2000
3. SHAH, Mubarak, Fundamentals of Computer Vision, Computer Science Department, Universty of Central Florida
4. CANNY, J., A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986.
5. GREEN, Bill, Ph.D. Student, Mechanical Engineering and Mechanics, Drexel University, Philadelphia
6. STAGG, Malcolm, Robot Vision Algorithms, http://www.virtualsciencefair.org/2004/stag4m0/public_html/
7. SMITH, Stephen, SUSAN Edge-Corner Detector, Oxford Centre for Functional Magnetic Resonance Imaging of the Brain (fMRIB), www.fmrib.ox.ac.uk/~steve
8. PARKER, J.R, Extracting Vectors From Raster Images, The University of Calgary Digital Media Laboratory
9. FESTO SEC-AC-305 Servo Sürücü Dökümanları
10. ADVANTECH PCI 1241/1242 4 Eksen Darbe Tipi Hareket Kontrol Kartı Kitapçığı
11. YAĞMUR, Levent, Tasarım ve İmalatta CNC ve CAD/CAM sistemlerinin fonksiyonları, TÜBİTAK - UME, Gebze / KOCAELİ, Ağustos 2004

ÖZGEÇMİŞ

1973 yılında İzmit'te doğdu. İlk ve orta öğrenimini İzmit'te tamamladı. 1991 yılında İzmit Teknik Lisesi elektronik bölümünden mezun oldu. Aynı yıl Bilkent Üniversitesi Bilgisayar ve Enformatik Mühendisliği bölümünü kazanarak mesleki eğitimini tamamladı. Mezuniyeti sonrasında bankacılık ve finans sektöründe veritabanı ve yazılım uzmanı olarak çalıştı. 2004 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalında yüksek lisans öğrenimine başladı.

