

# Görme Destekli Kartezyen Robot İçin Kenar Resmi Vektörizasyon Uygulaması

Cengiz Balta<sup>1</sup>

Sıtkı Öztürk<sup>2</sup>

Cüneyt Oysu<sup>3</sup>

<sup>1,2</sup>Elektronik ve Haberleşme Mühendisliği Bölümü, Kocaeli Üniversitesi

<sup>3</sup>Mekatronik Mühendisliği Bölümü, Kocaeli Üniversitesi

<sup>1</sup>e-posta: cengizbalta@kocaeli.edu.tr

<sup>2</sup>e-posta: sozturk@kocaeli.edu.tr

<sup>3</sup>e-posta: coysu@kocaeli.edu.tr

## Özetçe

Bu çalışmada, iki boyutlu parçaları görebilen ve bunlar için CAD-CAM parça programlama ve imalat ortamı fonksiyonlarını sağlayan genel amaçlı bir çizici-kescici kartezyen robot uygulaması ve bilgisayar yazılımı gerçekleştirilmiştir. Üç doğrusal eksenenden 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 birbir kopyasını görebilen, genel amaçlı bir çizici sistem üretilmiştir.

## 1. Giriş

Bu projede, imalat endüstrisinde yaygın olarak kullanılan iki boyutlu kesme makinalarına kamera desteği ile parçaları görme ve CAD-CAM tasarım-imalat ortamına ölçekli olarak aktarılabilme özelliği kazandırılması hedeflenmiştir. İmalata yönelik CAM uygulamalarında, işlenmemiş, ham haldeki ana parçanın, boyutları ve konumunun üretimi gerçekleştirecek olan makinaya 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 [3] 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[5] tarafından kullanılmış olan bazı kenar resminden gürültü eleme filtreleri kenar resmine uygulanmış, vektörlerin başlangıç ve bitiş noktalarındaki piksel çakışmalarının ve kaymalarının engellenmesi için Parker[7] tarafından geliştirilen vektörizasyon algoritmasının bazı

noktalarından yararlanılmıştır. Ayrıca Susan[6] 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, çember ve benzeri CAD nesnelerini, tasarım ekranından tarayıp, kartezyen robota hareket komutları şeklinde aktarır.

Projede gerçekleştirilen aşamalar şu şekilde özetlenebilir:

- XYZ kartezyen robot ve servo motorların kurulması
- Servo sürücüler ve motor bağlantılarının yapılması
- Hareket kartı ve servo sürücü haberleşmelerinin temin edilmesi
- Görüntü işleme, kenar bulma, vektörizasyon algoritmalarının geliştirilmesi
- Elde edilen vektör resminin CAD-CAM tasarım-imalat ortamına aktarılması
- Geliştirilen iki boyutlu CAD-CAM arayüzü ile kullanıcıya parça programlama imkanı sağlanması
- Kullanıcı tarafından ana parça üzerinde çizilen iki boyutlu parçaların tasarım ortamından XYZ kartezyen robota gönderilmesi

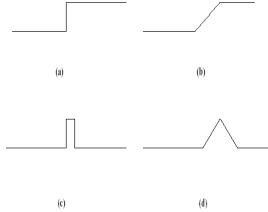
## 2. Kenar Bulma ve Vektörizasyon

### 2.1. Giriş

Bilgisayarlı görme problemlerinde kenar bulma algoritmaları sürekli olarak önemli bir problemi teşkil etmiştir. Bu çalışmada temel problem, iki boyutlu cisimlerin kamera desteği ile CAD ortamına aktarılması ve üç eksenli robot tarafından düzlemde çizdirilmesidir. İlk aşamada kameradan alınan resim, kenar bulma algoritmaları ile kenar resmine dönüştürülür ve tezgahtaki parçalar elde edilir. Sonrasında ise, kenar resmi CAD formatlı vektör resmine dönüştürülerek robota iletilebilir bir formata taşınır.

## 2.2. Bitmap Resimde Kenar Tipleri

Gri tonlu bir resimde, nesnelerin kenarlarının diki boyunca gri tonunun seviyesi hatırı sayılır oranda değişir. İdeal bir kenar, adım – step fonksiyonu ile modellenilebilir. Gerçek bir resimde ise, kenarın olduğu istikamette piksel değeri bir anda değil, kademeli olarak değişir. Dolayısıyla ancak bir rampa şeklinde modellenilebilir. Bazen de iki adım veya rampa kenar yan yana gelerek darbe veya çatı şeklinde bir sinyal üretebilirler[2]. Şekil 1’de kenar tipleri verilmiştir.



Şekil 1: Kenar tipleri, (a) Adım (b) Rampa (c) Darbe (d) Çatı

## 2.3. Kenar Bulmada Kullanılan Aşamalar

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

İlk olarak Sobel, türevleme aşamasından önce, resimdeki piksel değerlerini, komşu pikselleri ile ortalama olarak 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 eşığın 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.

### 2.3.1. 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 resmini,  $h$  çıktı resmini,  $g$  ise filtre maskesini göstermek üzere, ortalama – mean filtresi şu şekilde hesaplanır:

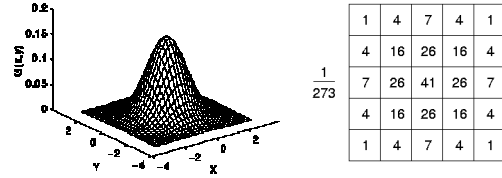
$$h(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j)g(i, j) \quad (1)$$

$$h(x, y) = f(x, y) * g(x, y) \quad (2)$$

Burada  $g$  filtre maskesini,  $*$  işareti ise konvolüsyon işlemini ifade ediyor

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:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$



Şekil 2:  $\sigma = 1$  değeri için gauss fonksiyonu ve tamsayı yaklaşımı

Görüntü işleme uygulamalarında aritmetik işlemlerin optimizasyonu açısından gauss maskesinin tamsayı yaklaşımı işlemlerde kullanılır.

### 2.3.2. Türevleme Aşaması

Sürekli bir fonksiyonun türevi şu şekilde tanımlanır:

$$f' = \frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} \quad (4)$$

Görüntü işlemede kullanılan resimler kesikli – discrete uzayda tanımlı oldukları için,  $\Delta x$  değeri en az 1 olabilir. Bu durumda türev formülü şu şekle gelir:

$$f' = \frac{df}{dx} = f(x) - f(x-1) \quad (5)$$

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

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:

$$\theta = \arctan(G_y / G_x) \quad (6)$$

$$G = \sqrt{G_x^2 + G_y^2} \text{ veya yaklaşım ile } |G| = |G_x| + |G_y| \quad (7)$$

$G$  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

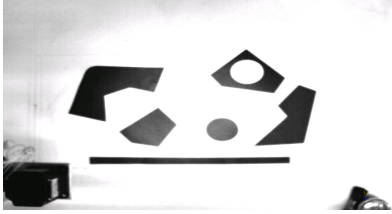
$G_x$

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

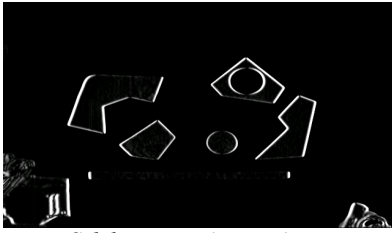
$G_y$

Şekil 3: Yatay ve düşey Sobel türev maskeleri

Şekil 4, 5, 6 ve 7'de işlemlerin çeşitli aşamalarında elde edilen imgeler gösterilmiştir.



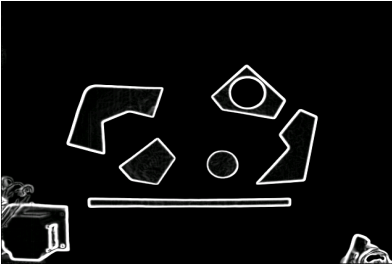
Şekil 4: Uygulamada kamera ile alınan görüntü



Şekil 5: Yatayda türevleme



Şekil 6: Düşeyde türevleme



Şekil 7: Gradient genlik resmi

### 2.3.3 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.

$$H(x, y) = \frac{G(x, y)}{\max_{i=1..n, j=1..n} G(i, j)} \times 100 \quad (8)$$

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

$$E(x, y) = \begin{cases} 1 & H(x, y) > T \\ 0 & \text{diğerleri} \end{cases} \quad (9)$$

### 2.3.4. Maksimum Olmayanların Bastırılması

Sobel 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:  $\theta = \arctan(G_y/G_x)$

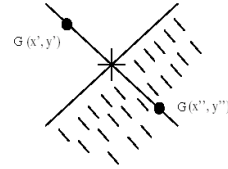
Kenara dik doğrultudaki gradient türev eğimi, piksel yapısından dolayı dört farklı yönde oluşabilir. Bu doğrultular şu şekilde bölgelere ayrılır:

- 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

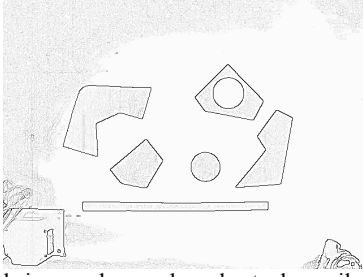
Her bir piksel için gradient yönü  $\arctan(G_y/G_x)$  ifadesi ile hesaplandıktan sonra, elde edilen açı değeri istikametindeki iki ilave komşu piksel seçilerek, ortanca pikselin bunlardan yüksek değerde olması istenir. Bu şart sağlanmaz ise, ortanca piksel sıfıra çekilerek kenar resminden elenir. Bu 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:

$$G(x, y) = \begin{cases} G(x, y) & \text{eger } G(x, y) > G(x', y') \text{ ve } G(x, y) > G(x'', y'') \\ 0 & \text{diğerleri} \end{cases} \quad (10)$$



Şekil 8: Gradient resminde, kenara dik doğrultudaki komşularından yüksek olmayan gradient genlik değerlerinin olduğu noktalar kenar resminden elenir



Şekil 9: Maksimum olmayanların bastırılması ile elde edilen kenar resmi. Bir piksel genişliğinde kenar resmi elde ediliyor.

### 2.3.5. 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 genişliği (piksel değeri) yüksek eşikten yüksek ise, kenar olarak bırakılır
  - Eğer kenara dik maksimum gradient genişliği (piksel değeri) düşük eşikten düşük ise, sıfıra çekilir
  - Eğer, piksel değeri, yüksek ve düşük eşik arasında ise, bu pikselin yüksek eşik aşan bir komşusu var ise kenar olarak kalmasına izin verilir.
  - Şayet eşik aşan bir komşusu yok ise sıfıra çekilerek kenar resminden elenir.

### 2.4. 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	

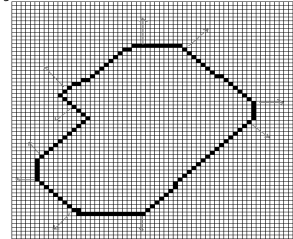
Şekil 10: Gürültü filtresi için çekirdek yapılar

### 2.5. Vektörizasyon Algoritması

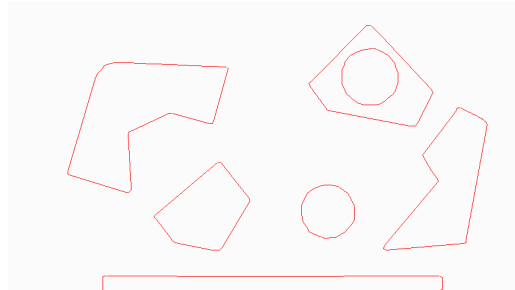
XYZ robotun çalışma tezgahından alınan parçaların kenar resmi bulunduktan sonra, geliştirilmiş olan, kenar resmi üzerinde gradient - türev yönüne göre piksel izleme temelli vektörizasyon algoritması ile tezgah üzerindeki parçalar CAD ortamına aktarılır.

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 artır, 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 11: Kenar pikselleri üzerinde gradient açı değeri izlenmesi metodu ile bitmap resimden vektör resim çıkarımı



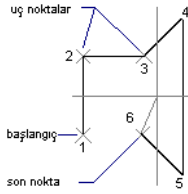
Şekil 12: Vektörizasyon sonrası CAD vektör resmi

### 3. Çizici/Plotter Yazılımı: CAD Nesneleri

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. CAD çizimleri çeşitli basit nesnelere birlikte kullanılması ile oluşur.

#### 3.1. 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 13: Çizgi nesnesi. Resimde toplam beş adet ayrı çizgi nesnesi mevcut

#### 3.2. 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.

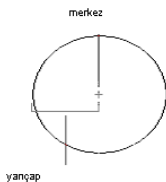


Şekil 14: Polyline nesnesi

Çoklu çizgi CAD veritabanında 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 de saklanır.

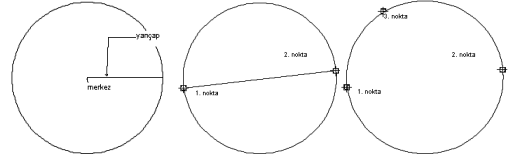
#### 3.3. Daire

Daire, CAD ortamında merkez noktası ve yarıçapı ile tanımlanır.



Şekil 15: Dairenin merkez noktası ve yarıçapı

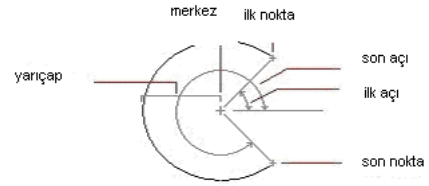
CAD ortamında daire merkez/yarıçap, iki nokta veya üç nokta metodları ile çizilebilir.



Şekil 16: Daire merkez/yarıçap, iki nokta ve üç nokta ile tanımlanabilir

#### 3.4. Yay Parçaları

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



Şekil 17: Yay parçasının CAD ortamındaki unsurları

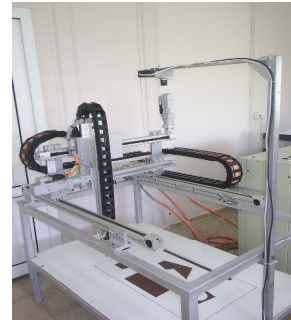
CAD ortamında yay parçası çizimi, merkez, yarıçap, başlangıç ve bitiş açısı ile veya üç nokta ile yapılır.

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.

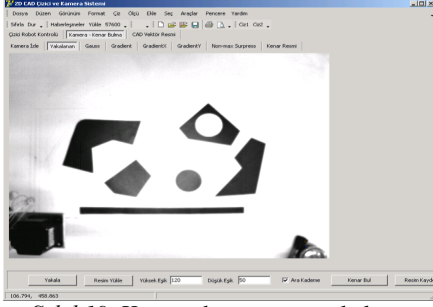
### 4. Uygulama Örnekleri

#### 4.1. Kenar Taraması

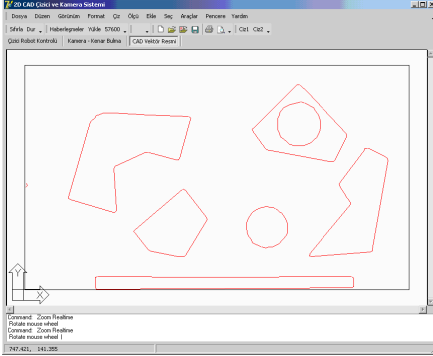
Bu uygulamada, tezgah üzerindeki parçalar kamera ile elde edilen resim kullanılarak vektör resmine dönüştürülür ve tasarım ortamına ölçekli olarak alınır. Kullanıcı, ana parçaların üzerinde çizilecek nesnelere tasarlayarak kartezyen robota çizdirir. Şekil 19, 20 ve 21'de bu uygulama gösterilmiştir.



Şekil 18: Kartezyen robot ve kamera sistemi



Şekil 19: Kameradan görüntü yakalama



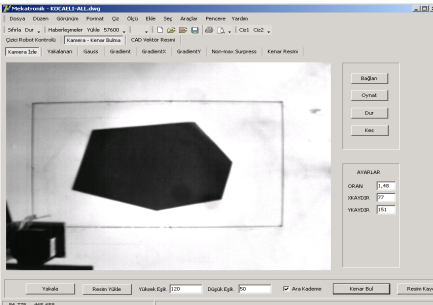
Şekil 20: Vektör resmi eldesi, parça tasarım ortamı



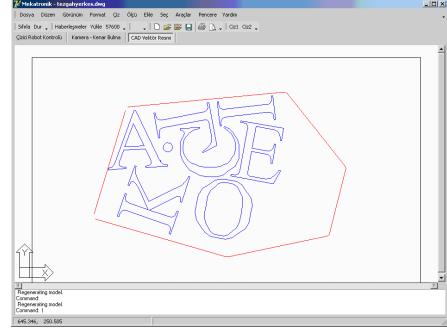
Şekil 21: Örnek kenar taraması

#### 4.2. Ana Parçadan Harf Kesme Uygulaması

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 22 ve 23'te bu uygulamanın aşamaları gösteriliyor.



Şekil 22: Ham parçanın kameradan okunması



Şekil 23: Harflerin ana parça üzerine yerleştirilmesi

### 5. Sonuçlar

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.

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 sağlaması ve bu nesnelere hareket komutları şeklinde hareket kartına aktarabilecek seviyede açık mimariye sahip olması gerekir.

### 6. Kaynakça

- [1] Fisher, R., Perkins, S., Walker, A., Wolfart, E., Hypermedia Image Processing Reference, (HIPR2)
- [2] Shah, Mubarak, Fundamentals of Computer Vision, University of Central Florida, 1997
- [3] Canny, J., A Computational Approach To Edge Detection, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol 8, No. 6, Nov 1986.
- [4] Green, Bill, Canny Edge Detection Tutorial, Drexel University, Philadelphia
- [5] Stagg, Malcolm, Development of an Embedded 3D Robot Visual System, 2004
- [6] S. M. Smith and J. M. Brady, 1997, "SUSAN - a new approach to low level image processing.". International Journal of Computer Vision 23 (1): 45-78
- [7] Parker, J.R, Extracting Vectors From Raster Images, University of Calgary, 1987