



## Yüzey Doldurma Teknikleri

### 1. Giriş

Bu deneyde dolu alan tarama dönüşümünün nasıl yapıldığı anlatılacaktır. Dolu alan tarama dönüşümü poligon içindeki piksellerin bulunup, bu piksellere karşılık gelen doğru parlaklık değerlerinin atanması anlamına gelir. Tarama dönüşümü, katı cisim üretimi için yüzeylerin boyanmasında kullanılmaktadır.

### 2. Sıralı Kenar Liste Yöntemi (Scan Line Algorithm)

Bu yöntem, poligonun kenarları ile tarama satırlarının kesişim noktalarının kullanılmasına dayanır. Kesişim noktaları bulunup bu noktalar üzerinde sıralama işlemleri yapılarak, belli ölçütler ışığında doldurulması gereken pikseller belirlenir. Yöntemin etkinliği sıralama yönteminin etkinliğiyle doğru orantılıdır.

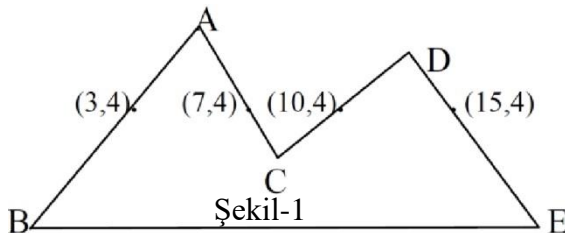
Basit şekilde bu yöntem aşağıdaki adımlardan oluşur:

1. Tüm poligon-tarama satırı kesişimleri bulunur. Poligon, bilinen DDA (Digital Differential Analyzer) veya Bresenham çizgi çizme yöntemleriyle oluşturulacağı için kesişim noktaları, çizgiler çizilirken kolaylıkla elde edilebilir. Noktalar bir listede tutulur. Her bir liste elemanı  $(x,y)$  şeklinde bir noktayı işaret edecektir.

2. Bütün  $(x,y)$  noktaları,  $y$  değerlerinin azalış veya artışına göre sıralandıktan sonra, aynı  $y$  değerine sahip noktalar da  $x$ 'in artan sırasına göre sıralanır.

3. Sıralamadan sonra, listeden nokta çiftleri seçilerek doldurulacak pikseller aşağıdaki kurala göre belirlenir.

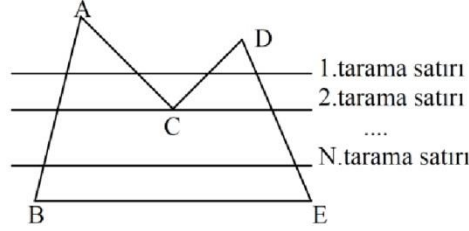
**Kural:**  $(x_1,y_1)$  ve  $(x_2,y_2)$  nokta çifti için,  $x$  tamsayı olmak üzere,  $x$  birer artırılarak  $x_1 \leq x + 1/2 \leq x_2$  koşulunu sağlayan  $(x,y_1)$  noktaları doldurulur. ( $y_1=y_2$ )



Yandaki şekilde  $(3,4)-(7,4)$  ve  $(10,4)-(15,4)$  çiftleri alındığında, yukarıdaki kurala göre doldurulacak noktaların ilk çift için  $(3,4)$ ,  $(4,4)$ ,  $(5,4)$ ,  $(6,4)$  ikinci çift için ise  $(10,4)$ ,  $(11,4)$ ,  $(12,4)$ ,  $(13,4)$ ,  $(14,4)$  olduğu görülmektedir.

Buna benzer bir yöntem de poligonun bulunduğu düzlem üzerinde, poligon sınırları dışından başlayıp soldan sağa doğru tarama satırları geçirilerek yüzeyin doldurulmasıdır. Soldan sağa doğru ilerlerken tek sayıda olan kesişimlerden sonraki pikseller doldurulur, kesişim sayısı çift olduğu anda doldurma işlemi kesilip, sağa doğru ilerlemeye devam edilir.

Şekil-2’de her bir tarama satırı için doldurulacak pikseller belirlenirken bazı problemlerle karşılaşılabilir. Örneğin ikinci tarama satırında soldan sağa doğru ilerlerken C noktasına rastlandığında doldurma işlemi durdurulacak ve bir sonraki DE kenarıyla kesişme noktasından itibaren yüzey tekrar doldurulmaya başlanacaktır. Yani C noktasından DE kenarına kadar olan pikseller doldurulmamış olacaktır.



Şekil-2

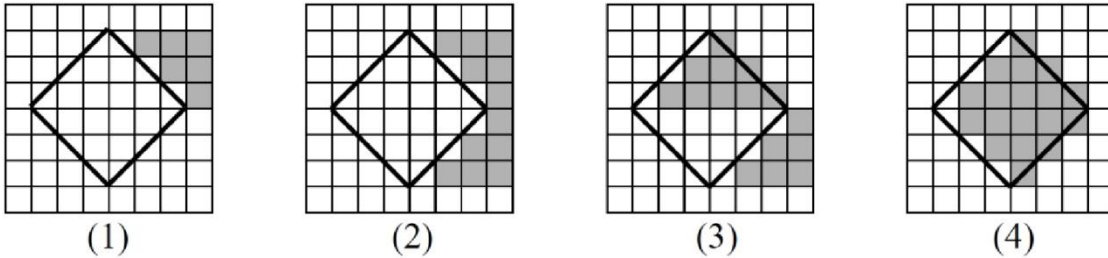
C noktası bir yerel minimum noktadır. Yerel minimum noktayı oluşturan kenarlar X-Y düzleminde Y eksenini boyunca artan değerlere sahiptir. Yukarıdaki probleme benzer bir problem D noktası için de vardır ve D noktası bir yerel maksimum noktadır. Bu durumda çözüm olarak tarama satırı boyunca yerel maksimum ve yerel minimum noktalar ihmal edilebilir. Kenar liste yönteminde de ikililer şeklinde değerlendirilme olacağından yerel minimum ve yerel maksimum noktaları listeye ikişer kere alınarak problem ortadan kaldırılmış olur.

Bu yöntem, her piksel bir kere adreslendiği için, etkin bir yöntemdir. Hızlı olduğu için gerçek zamanlı uygulamalara uygundur. Üstünlüklerine rağmen bu yöntem sıralama ön işlemine ek olarak yatay çizgiler bulunduran poligonların yatay kenarlarının ayrıca ele alınmasını gerektirir.

### 3. Kenar Doldurma Yöntemi

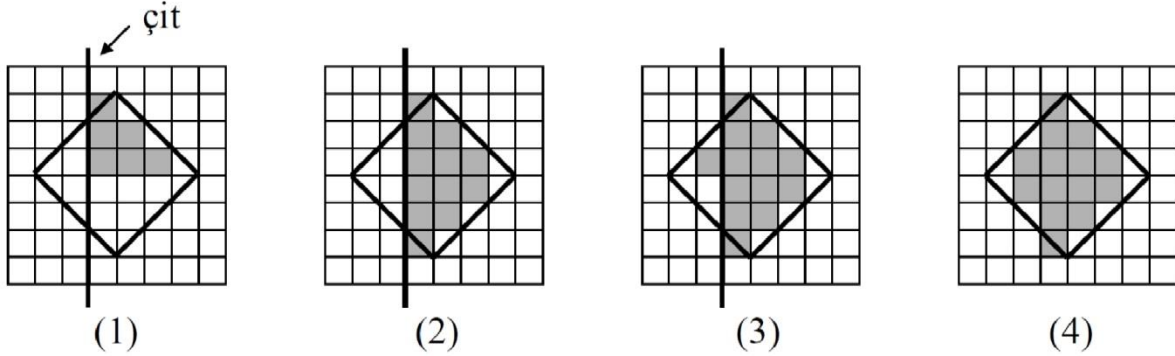
Kenar doldurma tekniğinde kesişim noktalarının tutulduğu listede sıralama ve listeyi düzenleme gibi işlemler yapmadan sadece kenarlar kullanılarak yüzey doldurma işlemi gerçekleştirilir.

Bir kenar seçilir ve o kenarın sağındaki tüm pikseller doldurulur. Eğer sağa doğru ilerlerken rastlanan piksel doldurulmuşsa, o piksel zemin rengine çevrilir. Tüm kenarlara bu işlem uygulanır ve sonuçta doldurulmuş yüzey elde edilir. Şekil-3’te bu uygulamanın adımları görülmektedir.



Şekil-3

Bu yöntemin sakıncası, poligon içindeki ve dışındaki piksellere birçok kere erişilmesidir. Büyük poligonlar için etkinliği azalan bir yöntemdir. Adreslenen piksel sayısını sadece poligon içindeki piksellerle sınırlamak için bir çit (fence) kullanılabilir. Poligonun herhangi bir noktasından bir çit seçilir. Kenarlardan çite doğru tarama işlemine başlanır. Yukarıdaki gibi, zemin renginde olan pikseller doldurulur, doldurulmuş pikseller ise zemin rengine çevrilir. Tüm kenarlar bitince poligon doldurulmuş olur. Aynı şekil için bu yöntemin adımları aşağıdaki gibi olacaktır.

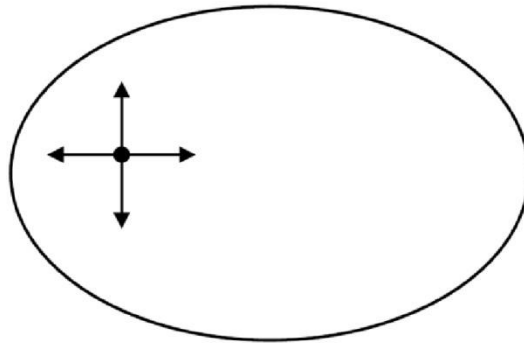


Şekil-4

#### 4. Çekirdek Doldurma Yöntemi (Flood Fill / Seed Fill Algorithm)

Sınırları tanımlı bir yüzey için geliştirilebilecek yöntemlerden biri de çekirdek doldurma yöntemidir. Bu yöntemde yığın, kuyruk veri yapıları veya özyinelemeli (recursive) fonksiyonlar kullanılabilir.

Örneğin, özyinelemeli fonksiyonlar kullanılarak uygulama geliştirilmek istenirse, öncelikle doldurulacak alan sınırları içerisinde bir piksel (seed point / boyama işleminin başlanacağı piksel) seçilir. Daha sonra, aşağıda sözde kodu verilen yapıdaki bir  $boya(x,y,renk)$  fonksiyonu piksel koordinat bilgileri ve boyanacak renk bilgisi parametreleri ile çağrılır. Bu fonksiyon, öncelikle pikselin kenar pikseli olup olmadığını kontrol eder (boyanacak şeklin sınırlarının aşılıp aşılmadığının kontrolü) ve kenar pikseli değilse o piksel ilgili renge boyar. Daha sonra, boyanan piksele komşu pikseller özyinelemeli boya fonksiyonuna parametre olarak gönderilir. İşlem yüzeyi doldurulana kadar özyinelemeli olarak devam eder.



Şekil-5

```
Fonksiyon boya(...)
  Eğer seçilen piksel sınır değeri değilse ve doldurulmamışsa
  Başla
    Piksel(x,y)=renk;
    boya(x+1,y,renk);
    boya(x,y+1,renk);
    boya(x-1,y,renk);
    boya(x,y-1,renk);
  Son
```

Özyinelemeli boya fonksiyonu sözde kodu

Çekirdek doldurma yöntemi yığın veri yapısı kullanılarak geliştirilmek istenirse, öncelikle başlangıç pikseli seçilir ve sonra aşağıdaki adımlarla yüzey doldurma işlemi gerçekleştirilir :

1. Seçilen piksel yığına itilir.
2. Yığından bir piksel çekilir. Piksel gereken renge boyanır.
3. Pikselin sağ, sol, üst ve alt komşularına bakılır. Herhangi bir komşu, sınır değeri değilse ve doldurulmuş değilse yığına itilir.
4. Yığındaki elemanlar bitinceye kadar 2. adıma gidilir.

Çekirdek doldurma yönteminin sakıncalarından biri, bir pikselin kendisini oluşturan pikseli de test etmesidir. Birkaç ek düzeltmeyle bu problem giderilebilse de, bu teknikte yığın boyutunun büyük olması kaçınılmazdır. Ayrıca bir piksele erişim sayısı 3-5 arasında olduğu için diğer yöntemlere göre daha yavaştır. Buna rağmen en büyük tercih sebebi, rasgele seçilen her yüzeyde iyi sonuç vermesidir.

## 5. Deney Hazırlığı

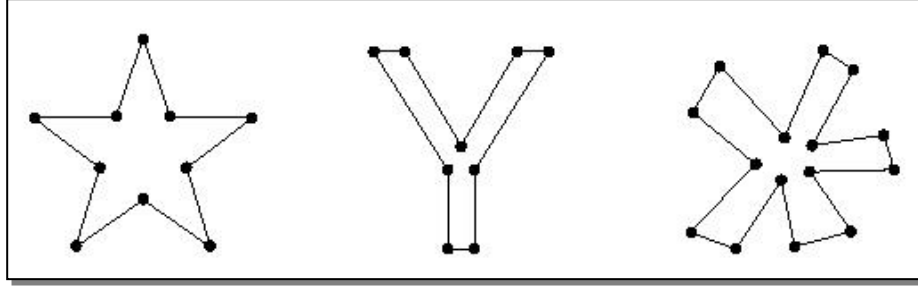
1. Deneyde DDA ve Bresenham doğru çizme yöntemlerinden de bahsedilecektir. Bu yöntemler hakkında bilgi sahibi olunuz.
2. Ekte verilen sıralı kenar liste yöntemini gerçekleyen uygulamayı (ScanLine klasörü içerisinde) çalıştırınız ve kaynak kodları inceleyiniz. (Uygulama OpenGL ile geliştirilmiştir, OpenGL'in yapılandırılması ile ilgili yönlendirmeler "IDE Kurulumu ve OpenGL Yapılandırması" adlı dökümanda verilmiştir.)
3. Ekte verilen ve çekirdek doldurma yöntemini kuyruk veri yapısı ile gerçekleyen FloodFill adlı uygulamayı çalıştırıp kodlarını inceleyiniz. Bu uygulamada, yukarıda belirtilen algoritmanın performansını yavaşlatan sakıncalarda kaçınma amacı ile farklı bir kuyruğa ekleme sıralaması kullanılmıştır. İnceleyiniz. (Uygulamayı çalıştırmak için gerekli bilgi BeniOku dosyası içerisinde verilmiştir.)
4. Çekirdek doldurma yönteminin etkinliğini arttırmak için neler yapılabilir, tartışınız.
5. Bildiğiniz farklı bir yöntem varsa o yöntemle, ya da buradaki yöntemlerin herhangi biriyle basit bir uygulama geliştiriniz.

## 6. Deney Tasarımı ve Uygulaması

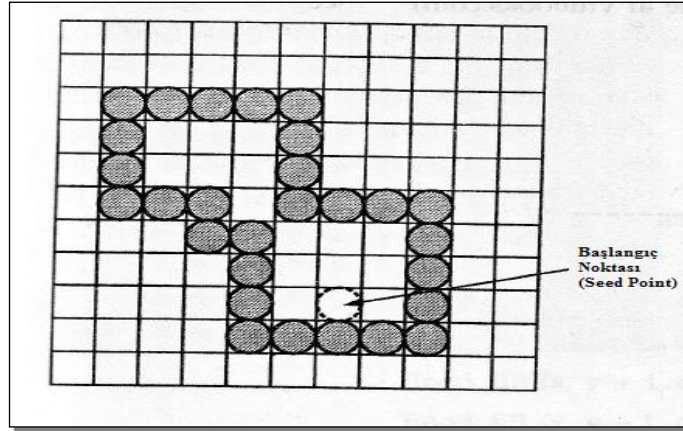
Deneyde, yüzey doldurma tekniklerinin işleyişleri tartışılacak, anlatılan yöntemlerle gerçekleştirilmiş uygulamalar incelenecektir. Bu yöntemlerde karşılaşılan ve karşılaşılabilecek problemler tartışılacaktır. Yöntemlerin hız ve bellek kullanımı yönünden üstünlükleri ve sakıncaları değerlendirilecektir.

## 7. Deney Soruları

1. Yüzey doldurma kavramını açıklayınız. Kullanım alanları nelerdir, araştırınız. Yukarıda bahsedilen 3 tekniğin işleyişini açıklayınız.
2. DDA ve Bresenham çizgi çizme yöntemlerini çizerek anlatınız ve bu iki algoritmayı performans ve doğruluk açısından karşılaştırınız.
3. Aşağıdaki konkav poligonlar sıralı kenar liste yöntemi ile doldurulmaya çalışılırsa nasıl sorunlarla karşılaşılır? Algoritma üzerinde nasıl düzeltmeler yaparak bu sorunların üzerinden gelirsiniz?



4. Aşağıdaki şekildeki kapalı alan 4 komşulu (four connected region) çekirdek doldurma yöntemi ile doldurulmaya çalışıldığında herhangi bir sorun ile karşılaşılır mı? Karşılaşırsa, bu sorun çekirdek doldurma algoritması üzerinde nasıl bir iyileştirme yapılarak aşılabilir? Açıklayınız.



5. Yüzey doldurma tekniklerini hız, bellek gereksinimi gibi performansı etkileyen kriterlere göre karşılaştırınız. Bu algoritmaların, işleyişleri sırasında hangi durumlarda hata verebileceğini açıklayınız.

## 8. Deney Raporu

Deney Raporunu sonraki hafta deneyine kadar (takım adına biriniz) dersin Moodle sayfasına yükleyiniz.