Kopuk Kenarları Bağlama

Edge Linking

Yöntemler

1. Hough kullanarak
2. Gradient büyüklüğü ve açıyı kullanarak

Hough ile

1. (Önceki yöntemlerden herhangi birini kullanarak) Binary imgeyi elde et.
2. $\rho\theta$ düzleminde alt bölümleri belirle.
3. En yüksek çakışmanın olduğu hücre sayısını incele.
4. Seçilen hücreye karşılık gelen pikseller arasındaki ilişkiyi (özellikle sürekliliği) incele.

Süreklilik: Pikseller arasındaki mesafe hesaplanıp belli bir eşik değerinin altındaysa bu pikseller süreklilik arz eder ve birleştirilir. Dolayısıyla kenar üzerindeki kesikler (gap) belirlenen eşikten küçüklerse kapanırlar.

Algoritmanın Avantajı: Sadece belli hücrelere karşılık gelen piksellerin incelenmesini sağladığı için önceki yöntemlerden üstünlüğe sahiptir.

Gradient ve Açı kullanarak

Yöntem 1: (x,y) merkezli bir kenar pikseline (s,t) komşusundaki diğer bir piksel aşağıdaki iki şartı sağlıyorsa bağlanır (M ve θ kenar büyüklüğü ve yönüdür)

$$|M(s,t) - M(x,y)| \leq E$$

$$|\theta(s,t) - \theta(x,y)| \leq A$$

Hesaplama maliyeti çok yüksek olduğundan gerçek zamanlı uygulamalar için uygun değildir.

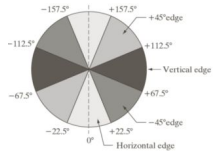
Gradient ve Açı kullanarak (devam)

Yöntem 2 (Gerçek zamanlı algoritmalara uygun)

Adım 1. Kenar büyüklüğü ve yönü hesapla.

Adım 2. Aşağıdaki eşitliğe göre ikili imgeyi oluştur.

$$g(x,y) = \begin{cases} 1 & \text{if } M(x,y) > T_M \; AND \; \theta(x,y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

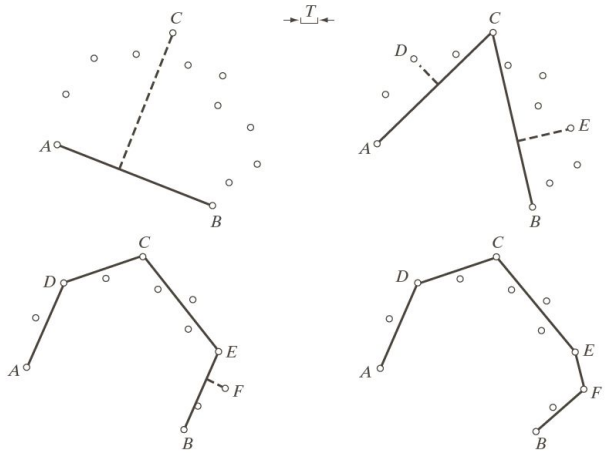


T_M kenar büyüklüğü için bir eşik, T_A, kenar açısı için bir eşiktir.

Adım 3. Satır doldurma: Belirlenmiş bir **K** uzunluğu aşmamak kaydıyla her bir satırdaki sıfır değerli boşlukları bir olarak doldur. Boşluk nasıl algılanır? Nesne ve zemin geçişlerine dikkat edilmeli. Nesneden(1) zemine(0) geçişte başlayıp tekrar nesneye ulaşıncaya kadar doldurma yapılmalıdır.

Adım 4. Sütun doldurma: Matrisin transpozu alınıp tekrar üçüncü adım çalıştırılmalı. İşlem bitince tekrar matris transpozu alınmalı.

Poligon Uydurma (fitting)



a b
c d

FIGURE 10.28
Illustration of the iterative polygonal fit algorithm.

Adım 1: İlk (A) ve son nokta (B) arasında bir çizgi çiz.
Adım 2: Diğer noktaların bu çizgiye olan maksimum dik uzaklığı bul.
Adım 3: Eşik değeri T den büyükse o zaman bu noktayı C olarak etiketle ve A-B çizgisini kaldırarak her iki noktadan C ye çizgi çiz.
Adım 4: Eşik değerinden büyük uzaklıkta nokta kalmayacak kadar devam et.