

Görüntü İşleme ve Kenar Tespiti Projesi Raporu

Öğrenci: Mehmet Akkan

Numara: 210404050

Bölüm: Bilgisayar Mühendisliği

Sınıf: 3. Sınıf

Bu ödevde, bir görüntünün kenarlarını tespit etmek için Canny, Prewitt ve Sobel operatörleri kullanılmıştır. Bu operatörler görüntü işleme alanında kenarları belirleme amacı taşıyan farklı yöntemlerdir. Projenin amacı, bu operatörlerin performansını karşılaştırmaktır.

Canny operatörü, iki eşik değeri kullanarak çalışır. Bu eşik değerleri, kenarların tespitini ve görüntünün azaltılmasını sağlar. Prewitt ve Sobel operatörleri ise yalnızca bir eşik değeri kullanır. Bu nedenle, Canny operatörü daha temiz ve belirgin kenar görüntüleri elde etmek için daha etkilidir.

Canny Kenar Tespit

Canny kenar tespiti algoritması, bir görüntünün kenarlarını tespit etmek için kullanılan en yaygın algoritmalarından biridir. Bu algoritma, iki eşik değeri kullanarak çalışır. Alt eşik, bir pikselin bir kenar olduğu kabul edilmesi için gereken minimum parlaklığı temsil eder. Üst eşik, bir pikselin bir kenar olmadığı kabul edilmesi için gereken maksimum parlaklığı temsil eder.

Bu projede, alt eşik değeri 50 ve üst eşik değeri 150 olarak kullanılmıştır. Bu değerler, görüntüdeki kenarların belirgin bir şekilde görünmesi için seçilmiştir.

Prewitt Operatörü

Prewitt operatörü, bir görüntünün kenarlarını tespit etmek için kullanılan bir filtreleme operatörüdür. Bu operatör, bir görüntünün x ve y yönlerindeki türevlerini hesaplar.

Prewitt operatörünün x ve y yönlerindeki filtreleri şu şekilde tanımlanmıştır:

```
kernel_x = np.array([[ -1,  0,  1], [ -1,  0,  1], [ -1,  0,  1]])
kernel_y = np.array([[ -1, -1, -1], [ 0,  0,  0], [ 1,  1,  1]])
```

Bir görüntünün x ve y yönlerindeki türevlerini aşağıdaki gibi hesaplar:

```
prewitt_edges_x = cv2.filter2D(image, -1, kernel_x)
prewitt_edges_y = cv2.filter2D(image, -1, kernel_y)
```

Sobel Operatörü

Sobel operatörü, Prewitt operatörüne benzer şekilde çalışan bir başka filtreleme operatörüdür. Ancak, Sobel operatörü Prewitt operatörüne göre daha hassastır.

Sobel operatörünün x ve y yönlerindeki filtreleri şu şekilde tanımlanmıştır:

```
sobel_edges_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)
sobel_edges_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3)
```

Bu filtreler, bir görüntünün x ve y yönlerindeki türevlerini aşağıdaki gibi hesaplar:

```
sobel_edges_x = np.sqrt(sobel_edges_x**2 + sobel_edges_y**2)
```

Sonuçlar

Canny kenar görüntüsü, diğer iki kenar görüntüsüne göre daha temiz ve belirgindir. Bunun nedeni, Canny operatörünün iki eşik değeri kullanarak çalışmasıdır. Alt eşik, kenarların tespit edilmesini sağlarken, üst eşik, görüntünün azaltılmasını sağlar.

Prewitt kenar görüntüsü, Canny kenar görüntüsüne göre daha az temiz ve belirgindir. Bunun nedeni, Prewitt operatörünün yalnızca bir eşik değeri kullanmasıdır. Bu, görüntünün kenarlarla birlikte tespit edilmesine neden olabilir.

Sobel kenar görüntüsü, Prewitt kenar görüntüsüne benzerdir. Bunun nedeni, Sobel operatörünün Prewitt operatörüne benzer şekilde çalışmasıdır.

Sonuç ve Değerlendirme

Bu projede, Canny, Prewitt ve Sobel operatörleri kullanılarak bir görüntünün kenarlarının tespit edilmesi işlemi gerçekleştirilmiştir.

Canny operatörü, diğer iki operatöre göre daha temiz ve belirgin kenar görüntüleri elde etmek için daha etkilidir. Bunun nedeni, Canny operatörünün iki eşik değeri kullanarak çalışmasıdır.

Prewitt ve Sobel operatörleri, Canny operatörü kadar temiz ve belirgin kenar görüntüleri elde etmek için etkili değildir. Bunun nedeni, bu operatörlerin yalnızca bir eşik değeri kullanmasıdır. Bu, görüntünün kenarlarla birlikte tespit edilmesine neden olabilir.

Canny operatörü, alt ve üst eşik değerlerini kullanarak gürültüyü azaltır. Bu, daha temiz ve belirgin kenar görüntüleri elde edilmesini sağlar.

Prewitt ve Sobel operatörleri, daha hızlı ve daha az kaynak gerektirir. Bu nedenle, gerçek zamanlı uygulamalarda kullanılabilirler.

Sonuç olarak, Canny operatörü, temiz ve belirgin kenar görüntüleri elde etmek için en iyi seçenektir. Ancak, Prewitt ve Sobel operatörleri, daha hızlı ve daha az kaynak gerektiren uygulamalar için kullanılabilir.