

NEVŞEHİR HACI BEKTAŞ VELİ ÜNİVERSİTESİ



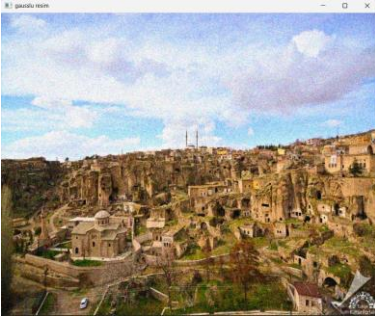
Ders: Görüntü İşleme

Konu: Gauss ve Tuz-Biber gürültülerini çift katmanlı hibrit filtreleme

Gauss Gürültüsü (Gaussian Noise):

Gauss gürültüsü, genellikle kamera sistemleri, görüntüleme sensörlerinin veya iletim hatalarındaki bozulmalardan kaynaklanan ve istatistiksel olarak normal dağılıma (çan eğrisi) uygulayan bir gürültü türüdür. Bu tür gürültü, görüntüdeki piksel değerlerine küçük, rastgele sapmalar ekleyerek, görüntünün gri ya da hafif bulanık bir şekilde bozulmasına neden olur. Matematiksel olarak her piksele, ortalaması sıfır ve varyansı σ^2 olan bir normal dağılımdan rastgele bir değer, yani $N(0, \sigma^2)$, eklenmektedir. Bu durum, düşük şiddette fakat yaygın bir bozulma etkisi oluşturur.

gauss



orijinal

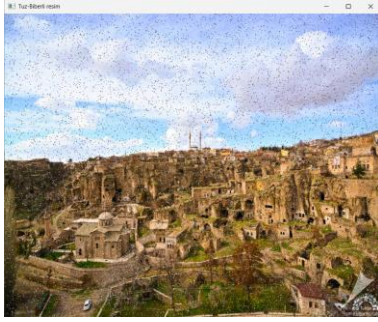


```
def gauss_gurultu_ekleme(image, mean=0, sigma=25):  
    gauss = np.random.normal(mean, sigma, image.shape).astype('float32')  
    gurultu = np.clip(image + gauss, 0, 255).astype('uint8')  
    return gurultu
```

Tuz-Biber Gürültüsü (Salt and Pepper Noise):

Tuz-biber gürültüsü, görüntü üzerinde belirli piksellerin rastgele ve tamamen siyah (0) ya da tamamen beyaz (255) olarak değer almasıyla oluşan bir bozulma türüdür. Genellikle kablosuz iletişimdeki bit kayıpları, kamera sensörü hataları ve görüntü sıkıştırma algoritmalarındaki bozulmalar neticesinde ortaya çıkmaktadır. Bu tür gürültü, eski televizyon yayınlarında gözlemlenen rastgele "karıncalanma" şeklindeki parazitlere benzer bir görsel bozulma üretir.

Tuz-Biber



Orijinal



```
def tuz_biber_gurultu_ekleme(image, salt_prob=0.02, pepper_prob=0.02):  
    gurultu = np.copy(image)  
    total_pixels = image.shape[0] * image.shape[1]  
  
    # Salt (beyaz) pikseller  
    num_salt = int(total_pixels * salt_prob)  
    coords = [np.random.randint(0, i, num_salt) for i in image.shape[:2]]  
    gurultu[coords[0], coords[1]] = 255  
  
    # Pepper (siyah) pikseller  
    num_pepper = int(total_pixels * pepper_prob)  
    coords = [np.random.randint(0, i, num_pepper) for i in image.shape[:2]]  
    gurultu[coords[0], coords[1]] = 0  
  
    return gurultu
```

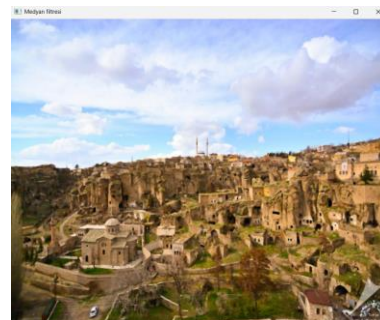
Medyan filtreleme

Tuz-biber gürültüsünün giderilmesinde **medyan filtreleme** yaygın olarak kullanılmaktadır. Bu yöntemde, belirli bir pencere (örneğin 3×3 boyutlarında) içerisinde yer alan piksellerin yoğunluk değerleri sıralanarak, ortanca (medyan) değer elde edilir ve merkez piksel bu değerle güncellenir. Bu sayede, aşırı uç (0 veya 255) değerlere sahip bozulmuş pikseller, komşu piksellerin tipik değerleri ile değiştirilerek etkisiz hale getirilir. Ancak, bu yöntem Gauss gürültüsüne karşı etkili değildir.

Tuz-Biber



Medyan Filtreleme



Gauss Gürültüsünün Filtrelenmesi:

Gauss gürültüsünün giderilmesinde en yaygın yöntemlerden biri **Gaussian filtreleme**dir. Bu filtre, komşu piksellere verilen ağırlıkları Gaussian dağılımına göre belirleyerek, ağırlıklı ortalama alır ve merkez pikseli bu şekilde günceller. Böylece görüntüdeki düşük yoğunluklu, rastgele gürültüler etkin bir biçimde yumuşatılır. Bununla birlikte, Gaussian filtresi görüntüdeki kenar detaylarını da bulanıklaştırma eğiliminde olduğundan, özellikle kenar korumasının kritik olduğu uygulamalarda sınırlı başarı göstermektedir. Ayrıca bu yöntem, tuz-biber tipi gürültüye karşı etkisizdir; çünkü bu tür gürültü ani ve keskin yoğunluk değişimlerinden oluşur.

Gauss Gürültüsü



Gauss filtreleme



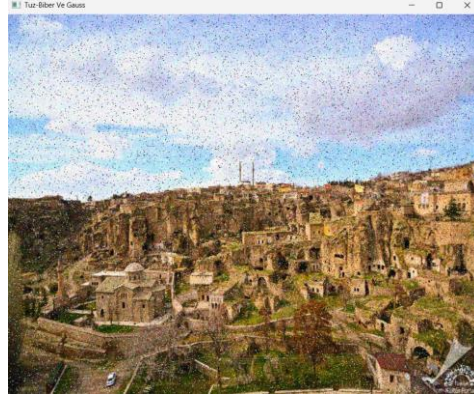
Çift Katmanlı Hibrit Filtreleme (Dual Layer Hybrid Filtering):

Gauss ve tuz-biber gürültüsünün aynı anda yer aldığı durumlarda, her iki gürültü türünü eş zamanlı olarak ortadan kaldırmak amacıyla **çift katmanlı hibrit filtreleme** yaklaşımı geliştirilmiştir. Bu yöntem, ardışık olarak uygulanan iki filtreden oluşur: birinci katman olarak **medyan filtresi**, ikinci katman olarak ise **Gaussian filtresi** kullanılır.

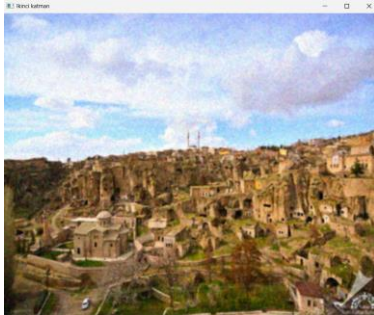
Filtreleme sıralaması bu yöntemde oldukça kritiktir. Öncelikle medyan filtresinin uygulanması, görüntüdeki ani değişimlere (siyah ve beyaz pikseller) müdahale ederek bu uç değerleri ortadan kaldırır. Eğer bunun yerine ilk olarak Gaussian filtresi uygulanırsa, siyah ve beyaz gibi uç değerler çevredeki piksellere yayılır, bu da medyan filtresinin etkinliğini azaltır ve görüntüdeki detayların kaybına yol açar. Bu nedenle, çift katmanlı hibrit filtrelemede doğru sıralama; önce **medyan filtresi** ile keskin gürültülerin giderilmesi, ardından **Gaussian filtresi** ile daha yumuşak ve yaygın bozulmaların azaltılması şeklindedir.

Bu yaklaşım, karma tipte gürültü içeren görüntülerde detay kaybını minimize ederken, hem ani hem de düşük şiddetli bozulmaların etkili bir biçimde giderilmesini sağlar.

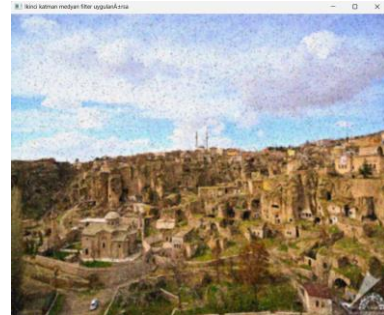
Hem gauss hem de tuz-biber



Katmanlar doğru uygulanırsa



Katmanlar yanlış uygulanırsa



Bu çalışmada kullanılan görseller, Aksaray iline ait görüntülerdir. Aşağıda yer alan ana kod satırlarında belirtildiği ve yukarıda tanımlanan fonksiyonların işlevlerinde gösterildiği üzere, görüntüler sırasıyla; orijinal hâliyle, Gauss gürültüsü eklenmiş, Tuz-Biber gürültüsü eklenmiş, yalnızca Medyan filtreleme uygulanmış, yalnızca Gauss filtreleme uygulanmış, her iki gürültü türü birlikte uygulanmış, çift katmanlı filtreleme doğru sırayla (önce medyan, ardından Gauss) ve yanlış sırayla (önce Gauss, ardından medyan) uygulanmış hâlleriyle işlenmiştir.

Main

```
image = cv2.imread('aksaray.jpg',1)

if image is not None:
    # görüntüyü yeniden boyutlandırmaya yarar
    image2 = cv2.resize(image, (800, 640))
    cv2.imshow("Orjinal", image2)

    #resime gauss gürültüsü ekler
    gausslu_resim = gauss_gurultu_ekleme(image2)
    cv2.imshow("Gausslu resim", gausslu_resim)

    #gauss filtresi sayesinde gauss gürültüsünü filtreler
    gauss_filtresi = cv2.GaussianBlur(gausslu_resim, (5,5),1)
    cv2.imshow("Gauss filtresi", gauss_filtresi)

    #resime tuz-biber eklemeye yarar
    tuz_biber_resim = tuz_biber_gurultu_ekleme(image2)
    cv2.imshow("Tuz-Biberli resim", tuz_biber_resim)

    #ortalama filtresi sayesinde tuz-biber gürültüsünü filtreler
    medyan_filtresi = cv2.medianBlur(tuz_biber_resim,3)
    cv2.imshow("Medyan filtresi", medyan_filtresi)

    #tuz-biber gürültüsü ekleyen fonksiyona gauss gürültülü resim göndermeye yarar
    gurultu_karismi = tuz_biber_gurultu_ekleme(gauss_gurultu_ekleme(image2))
    cv2.imshow("Tuz-Biber Ve Gauss", gurultu_karismi)

    gurultu_karismi_filtreleme=cift_katmanli_hibrid_filtreleme(gurultu_karismi)
    gurultu_karismi_filtreleme=cift_katmanli_hibrid_filtreleme_katmanlari_degistir(gurultu_karismi)
```

Avantajlar ve Dezavantajlar

Bu çalışmada uygulanan filtreleme yöntemlerinin her biri belirli türdeki gürültüleri filtrelemede etkili olup her birinin avantajları ve sınırlılıkları bulunmaktadır. Bunlar:

Medyan Filtresi

Avantajları:

Tuz ve biber gürültülerinde bulunan keskin değerlere sahip (siyah”0”, beyaz”255”)

Pikselleri yok ederek bozulmuş görseli orijinal haline daha çok benzettir. Gauss filtrelemeye karşın Non –lineer bir filtrelemeye sahip olmasından sebep kenar detaylarını koruma konusunda daha etkilidir.

Dezavantajları:

Bu filtreleme çeşidinin fazla kullanılması sonucu resim içerisindeki detaylar kaybolur.

Gauss ile bozulmuş görsellerde detayları daha fazla kaybedebileceği için bu amaçla kullanılamaz.

Gauss (Gaussian) Filtresi

Avantajları:

Lineer bir filtreleme yöntemi olmasından dolayı matematiksel hesaplaması daha basittir. Görüntüyü yumuşattığından dolayı gauss gürültüsünü filtrelemek için daha uygundur

Dezavantajları:

Tuz ve Biber gürültüsü için kullanılamaz. Görselde bulunan kenarları ve keskin değerlere sahip pikselleri dağıttığı için görseli düzeltemez.

Çift Katmanlı Hibrit Filtreleme (Dual Layer Hybrid Filtering)

Avantajları :

Hem Gauss hem de Tuz-Biber filtreleme için kullanılabilir ve aynı anda filtreleme işlemi yapılabilir. Katmanlar doğru sıra ile uygulandığında etkili bir şekilde görseli orijinal şekline benzetebilir.

Dezavantajları:

Katmanları yanlış sıra ile yapılırsa piksellerin keskin değerleri yayılacağından dolayı görüntü bozulur. Hesaplama maliyeti tek katmanlı filtreleme yöntemlerine göre daha pahalıdır.

Muhammet Ali Aslanhan

22260810029