

Gaussian Filter

Kode ini menggunakan Gaussian Filtering untuk mengurangi noise dan menghaluskan gambar. Metode ini bekerja dengan mengonvolusi gambar menggunakan kernel Gaussian, yang membuat gambar tampak lebih halus. Noise berkurang, namun efek smoothing bisa mengorbankan beberapa detail halus. Meskipun begitu, Gaussian filter tetap efektif untuk membersihkan noise dengan sedikit penurunan ketajaman.

1. Import Libraries

```
import cv2
import numpy as np
```

cv2: Library OpenCV yang digunakan untuk pemrosesan gambar, termasuk membaca, menampilkan, dan memanipulasi gambar.

numpy: Library untuk operasi numerik pada array multidimensi, seperti manipulasi data citra yang direpresentasikan sebagai matriks piksel.

2. Load Image

```
image = cv2.imread('C:/Kuliah ya ges yak/Semester
3/PCD/Foto/example.jpg')
if image is None:
    print(" Error: Image not found. Check
the path.")
    exit()
```

cv2.imread(): Mengambil gambar dari file. Jika gambar tidak ditemukan, program akan berhenti sebagai bagian dari penanganan error.

3. Define Gaussian Kernel

```
kernel = np.array([[1, 2, 1],
                    [2, 4, 2],
                    [1, 2, 1]]) /
16
```

Gaussian Kernel: Matriks 3x3 yang digunakan untuk menghasilkan efek smoothing pada gambar.

Normalisasi: Nilai kernel dibagi 16 untuk menjaga total intensitas gambar tetap konsisten.

4. Add Padding

```
padded_image = cv2.copyMakeBorder(image, 1, 1, 1,
1, cv2.BORDER_REFLECT)
```

cv2.copyMakeBorder(): Menambahkan padding (dengan metode refleksi) di sekeliling gambar.

5. Create Output Image

```
gaussian_filtered_image = np.zeros_like(image)
```

np.zeros_like(): Membuat array dengan ukuran yang sama dengan

7. Save Filtered Image

```
cv2.imwrite('gaussian_filtered_image_manual.jpg',
gaussian_filtered_image)
```

cv2.imwrite(): Menyimpan gambar hasil konvolusi ke file dengan nama yang diberikan.

8. Display Images

```
cv2.imshow('Original Image', image)
cv2.imshow(' Gaussian Filtered Image (Manual)',
gaussian_filtered_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

cv2.imshow(): Menampilkan gambar dalam jendela OpenCV.

cv2.waitKey(0): Menunggu input dari pengguna, seperti menekan tombol, untuk menutup jendela.

cv2.destroyAllWindows(): Menutup semua jendela gambar yang sedang terbuka.

Perbandingan Input dan Output



Gambar Original (Kiri):

Tingkat noise sangat tinggi, tersebar merata dengan bintik-bintik terang dan gelap yang membuat detail sulit terlihat.

Gambar Gaussian Filtered (Kanan):

Setelah diterapkan filter Gaussian, noise berkurang secara signifikan. Gambar menjadi lebih halus, meskipun beberapa noise masih tersisa dan detail halus sedikit kabur akibat efek smoothing.

gambar asli untuk menyimpan hasil filter.

6. Apply Gaussian Filter (Manual Convolution)

```
for channel in range(image.shape[2]):
    for i in range(1, padded_image.shape[0] - 1):
        for j in range(1, padded_image.shape[1] - 1):
            patch = padded_image[i - 1:i + 2, j - 1:j + 2, channel]
            gaussian_filtered_image[i - 1, j - 1, channel] = np.sum(patch * kernel)
```

Iterasi Channel: Memproses masing-masing channel warna (R, G, B) secara independen.

Patch 3x3: Mengambil area 3x3 dari gambar yang telah dipadankan.

Konvolusi: Mengalikan area tersebut dengan kernel Gaussian, lalu menjumlahkan hasilnya.



By **Dody**
cheatography.com/dody/

Not published yet.
Last updated 21st October, 2024.
Page 1 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>