

Tugas 4 Pengolahan Citra Video

Filter Domain Frekuensi

Kenanya Keandra Adriel Prasetyo
5024211004

October 22, 2023

1 Pendahuluan

Pada Tugas 4 mata kuliah Pengolahan Citra Video, penulis diminta untuk membuat program yang akan melakukan filter pada citra pada domain frekuensi. Filter yang dilakukan adalah filter bandreject, bandpass, dan notch.

2 Pembahasan Program

Kode keseluruhan pada Tugas 4 Pengolahan Citra Video ini dapat diakses pada https://github.com/KenanyaKAP/pcv_4

2.1 Fungsi Filter

```
1 def ideal_filter(img, low_cut, high_cut, inverted):
2     M, N = img.shape
3     dft = cv2.dft(np.float32(img), flags=cv2.DFT_COMPLEX_OUTPUT)
4     dft_shift = np.fft.fftshift(dft)
5
6     u = np.arange(M)
7     v = np.arange(N)
8     u = u - M//2
9     v = v - N//2
10    U, V = np.meshgrid(v, u)
11    D = np.sqrt(U**2 + V**2)
12    mask = ~((D < high_cut) & (D >= low_cut)) if inverted else ((D <
13    high_cut) & (D >= low_cut))
```

```

14     dft_shift_filtered = dft_shift * mask[:, :, np.newaxis]
15
16     idft = np.fft.ifftshift(dft_shift_filtered)
17     img_filtered = cv2.idft(idft)
18     img_filtered = cv2.magnitude(img_filtered[:, :, 0], img_filtered[:,
19     :, 1])
19
20     img_filtered = cv2.normalize(img_filtered, None, alpha=0, beta=255,
21     norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8U)
22
23     return img_filtered, dft_shift, dft_shift_filtered, mask

```

Program di atas adalah fungsi utama dari filter domain frekuensi. Caranya adalah dengan mengubah gambar yang diberikan kedalam domain frekuensi, dengan fungsi dft dari opencv.

Fungsi tersebut memiliki parameter yakni `low_cut`, `high_cut`, dan `inverted`. `Low_cut` disini adalah frekuensi rendah yang digunakan, sehingga frekuensi dibawah nilai yang diberikan akan diubah nilainya menjadi 0.

Parameter selanjutnya adalah `high_cut`. Parameter ini berkebalikan dari `low_cut`. `High_cut` disini adalah frekuensi tinggi yang digunakan, sehingga frekuensi diatas nilai yang diberikan akan diubah nilainya menjadi 0.

`Low_cut` dan `High_cut` akan berperan untuk membuat mask dari domain frekuensi, sehingga pengguna dapat dengan mudah mengatur filter sesuai kebutuhan. Dan parameter terakhir adalah `inverted`. Parameter ini akan membalikkan nilai masking yang ada. Sehingga terbentuklah masking yang berkebalikan. Ini dapat berguna untuk membuat filter bandpass dan bandreject.

Dengan fungsi diatas, pengguna dapat membuat filter dengan mudah sesuai dengan apa yang diperlukan. Pengguna juga dapat mengubah nilai frekuensi cut off dari setiap filter sehingga terlihat hasil citra secara realtime dengan nilai frekuensi yang berbeda-beda

2.2 Fungsi Spectrum

```

1 def plot_spectrum(dft_shift):
2     magnitude_spectrum = 20 * np.log(cv2.magnitude(dft_shift[:, :, 0],
3     dft_shift[:, :, 1] + 1e-5))
4     return magnitude_spectrum

```

Fungsi `plot_spectrum(dft_shift)` adalah sebuah fungsi yang digunakan untuk menghasilkan spektrum magnitudo dari hasil Transformasi Fourier Diskrit (DFT) yang telah digeser (shifted). Fungsi ini membutuhkan masukan berupa citra DFT yang sudah digeser, dan kemudian mengembalikan spektrum magnitudo dari citra tersebut.

2.3 Fungsi update image

```
1 def updateImage():
2     img_filtered, dft_before, dft_after, mask = ideal_filter(img, lowCut,
3         highCut, invert)
4
5     # Plot original image
6     axs[0, 0].cla(), axs[0, 0].set_title('Original')
7     axs[0, 0].imshow(img, cmap='gray'), axs[0, 0].axis('off')
8
9     # Plot frequency spectrum
10    axs[0, 1].cla(), axs[0, 1].set_title('Frequency Spectrum')
11    axs[0, 1].imshow(plot_spectrum(dft_before), cmap='gray'), axs[0, 1].
12    axis('off')
13
14    # Plot Mask
15    axs[1, 0].cla(), axs[1, 0].set_title('Mask')
16    axs[1, 0].imshow(mask, cmap='gray'), axs[1, 0].axis('off')
17
18    # Plot frequency spectrum after masked
19    axs[1, 1].cla(), axs[1, 1].set_title('Frequency Spectrum after masked
20    ')
21    axs[1, 1].imshow(plot_spectrum(dft_after), cmap='gray'), axs[1, 1].
22    axis('off')
23
24    # Convert histogram canvas to image and show
25    fig.canvas.draw()
26    histoImg = np.frombuffer(fig.canvas.tostring_rgb(), dtype=np.uint8)
27    histoImg = histoImg.reshape(fig.canvas.get_width_height()[::-1] +
28    (3,))
29    histoImg = cv2.cvtColor(histoImg, cv2.COLOR_RGB2BGR)
30    cv2.imshow("Histogram", histoImg)
31
32    # Show image
33    cv2.imshow('Filtered', img_filtered)
```

Fungsi diatas adalah fungsi updateImage. Fungsi inilah yang merupakan fungsi utama dari program. Fungsi ini akan memanggil fungsi ideal filter, dan menghasilkan output yakni img_filtered, dft_before, dft_after, dan mask. Nilai-nilai ini kemudian akan ditampilkan dengan menggunakan library matplotlib.

2.4 Hasil Citra Program

Berikut adalah citra yang dihasilkan dari program yang telah dibuat.

2.4.1 LowPass Filter

2.4.2 HighPass Filter

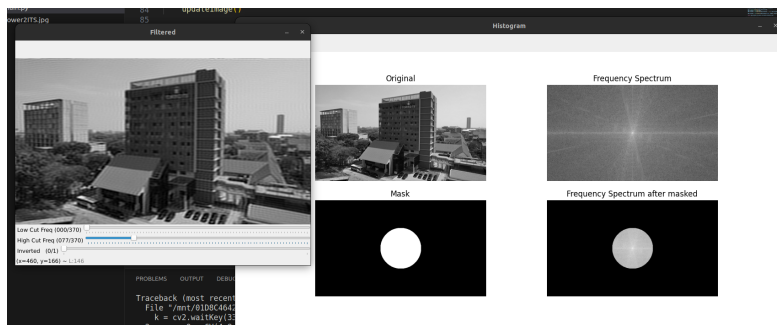


Figure 1: Fungsi LowPass Filter



Figure 2: Citra hasil LowPass Filter

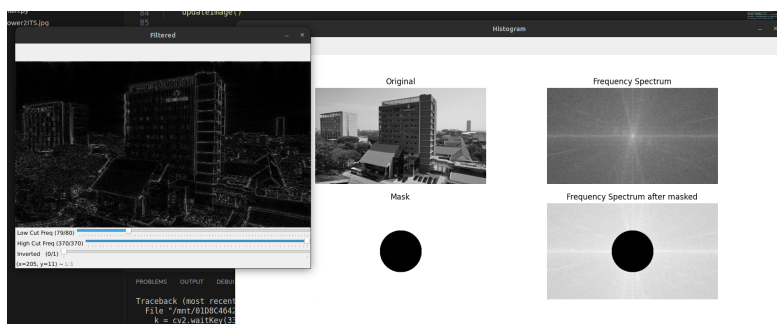


Figure 3: Fungsi HighPass Filter



Figure 4: Citra hasil HighPass Filter

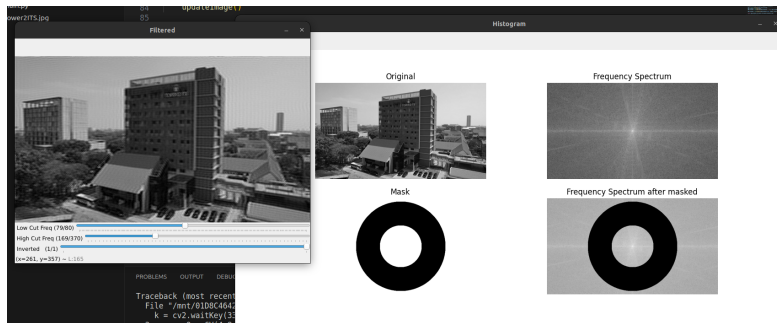


Figure 5: Fungsi BandReject Filter

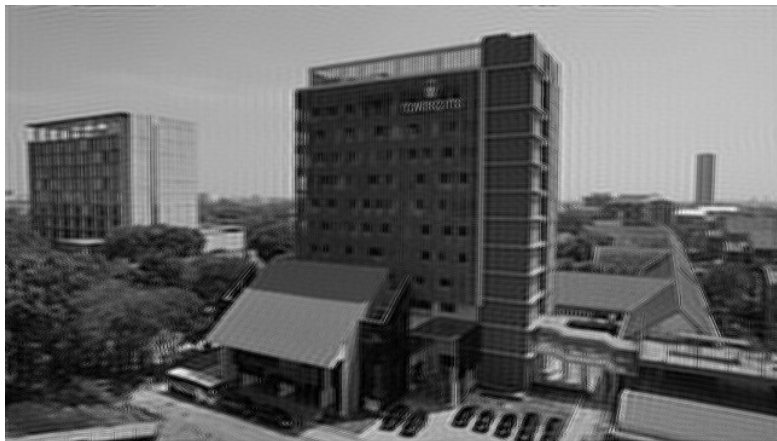


Figure 6: Citra hasil BandReject Filter

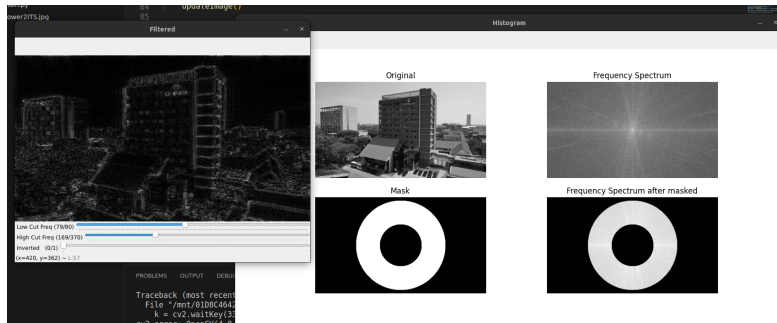


Figure 7: Fungsi BandPass Filter

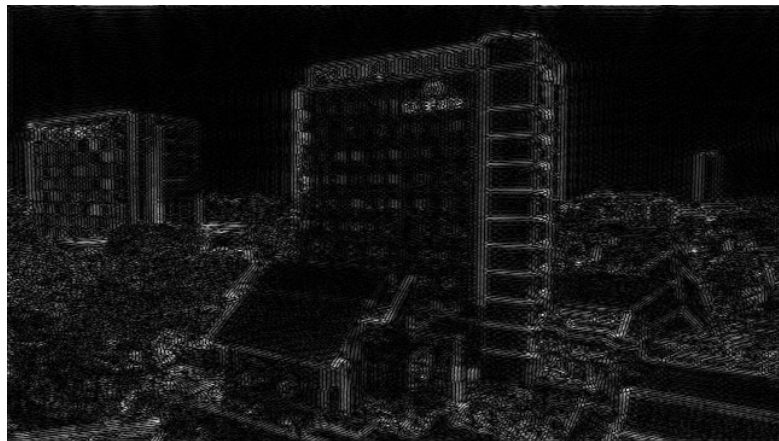


Figure 8: Citra hasil BandPass Filter

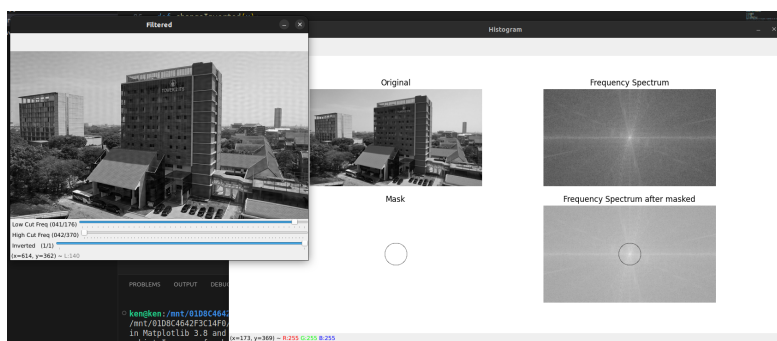


Figure 9: Fungsi Notch Filter



Figure 10: Citra hasil Notch Filter