**Color Image Processing 2**

****

**SISTEM PENGOLAHAN CITRA**

**PROGRAM STUDI SISTEM KOMPUTER**

**SCHOOL OF INFORMATION SCIENCE & TECHNOLOGY**

**UNIVERSITAS PELITA HARAPAN**

**DISUSUN OLEH:**

**Ray Antonius**

**1 November 2018**

**Tujuan**: Memperlihatkan mahasiswa cara memanipulasi gambar 3 channel RGB, dan juga menunjukkan kepada mahasiswa gaussian filtering yang terjadi pada gambar dalam space YCbCr.

**Alat dan Bahan:**

* Octave
* Lena512color.tiff
* Ycbcr.mat

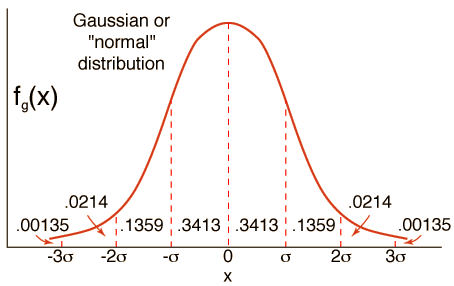
**Prosedur:**

1. Buka octave/matlab
2. Muatkan lena512color.tiff pada sebuah variabel dengan menggunakan imread(‘lena512color.tiff’);
3. Pecahlah gambar tersebut menjadi 3 channel dengan memilih index array yang berbeda (R=img(:,:,1); G=img(:,:,2); B=img(:,:,3);)
4. Gambarkan masing-masing channel dan gambar original dengan fungsi subplot(2,2,n) (n=1,2,3,4).
5. Buat channel R menjadi 0 dan bagi semua nilai channel hijau menjadi setengah.
6. Gambarkan lagi tiap channel yang telah dimanipulasi dan hasil gabungannya dengan fungsi subplot(2,2,n), catat hasilnya.
7. Muatkan ycbcr.mat dengan fungsi load(‘ycbcr.mat’), akan didaptkan h, dan Ycbcr
8. Pisah ycbcr menjadi variabel Y, Cb, dan Cr
9. Filter variabel Y dengan fungsi filt2(h, Y);
10. Ubah Y, Cb, Cr menjadi RGB dan tunjukkan hasil gabungan 3 channel RGB tersebut dengan imshow();
11. Ulang bagian 7-8, lalu sekarang filter Cb dan Cr dengan fungsi filt2(h, Cb); filt2(h, Cr);
12. Gunakan imshow untuk menunjukkan hasil manipulasi, dan berikan kesimpulannya.

**Hasil**:

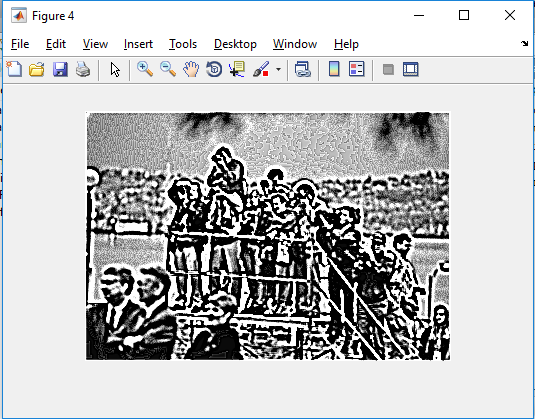
Pembahasan:

1. Gambar dari mesh yang ditunjukkan pada gambar 1 sesuai dengan hipotesa yang diambil. Gaussian distribution adalah sebuah distribusi yang menjelaskan mengenai distribusi probabilitas. Di mana bagian tengahnya akan lebih padat ketimbang yang lain. Hal ini diadopsi oleh mesh yang dilihat pada gambar 1. Namun, karena dalam 2D, maka gaussian filter tersebut memiliki bentuk seperti sebuah gunung. Konsepnya masih sama, di mana titik tengah akan memiliki dampak pengolahan lebih banyak daripada bagian yang lebih di pinggir.



1. Gambar2 dan 3 adalah hasil dari unsharp masking. Unsharp masking berfungsi dengan mengkomparasi 2 gambar, yaitu gambar yang asli dengan gambar yang telah di filter (konvolusi dengan filter). Setiap titik yang “unsharp” akan dibuang oleh mask tersebut. Pada kali ini, gambar 2 lebih tajam daripada gambar asli. Namun, jika dilihat dengan teliti, akan terlihat lebih kasar. Hal ini disebabkan oleh gambar hasil pengolahan akan kehilangan beberapa fitur dari gambar asli.

Apabila kita meningkatkan nilai alpha dan beta menuju ke titik ekstrim, maka kekasaran yang dimiliki oleh gambar olahan akan semakin terlihat.



Gambar 4. Unsharp Mask dengan alpha 110 dan beta 109

1. Gambar 3 memiliki hasil yang lebih tajam dari gambar 2, dengan meningkatkan alpha dan beta (dengan kondisi a-b = 1, gambar akan menjadi semakin tajam namun kasar (lihat gambar 4). Hal ini dikarenakan oleh intensitas yang diaplikasikan kepada gambar. Hal ini dapat dianalogikan dengan sebuah speaker. Speaker dengan peningkatan suara yang terlalu tinggi juga dapat menyebabkan noise ikut ditingkatkan. Hal ini menyebabkan output yang tidak sebaik data asli.
2. Properti yang dimiliki oleh gambar ini dapat dimanfaatkan untuk digunakan edge detection, dengan sengaja ‘mengkasarkan’ gambar, kita dapat melihat dengan baik titik titik yang dapat dianggap sebagai sebuah pinggiran.

**Kesimpulan:**

Gaussian Filtering adalah sebuah fungsi yang dapat digunakan untuk melakukan penajaman gambar. Gambar yang diolah akan terlihat lebih tajam, namun juga akan keluar lebih kasar ketimbang dari gambar asli. Unsharp masking membuang bagian-bagian gambar yang kurang tajam dengan melakukan komparasi antara gambar asli dengan gambar hasil filter.

|  |  |  |
| --- | --- | --- |
| **Nama** | **NIM** | **Tanda Tangan** |
| Ray Antonius | 00000021587 | [placeholder] |