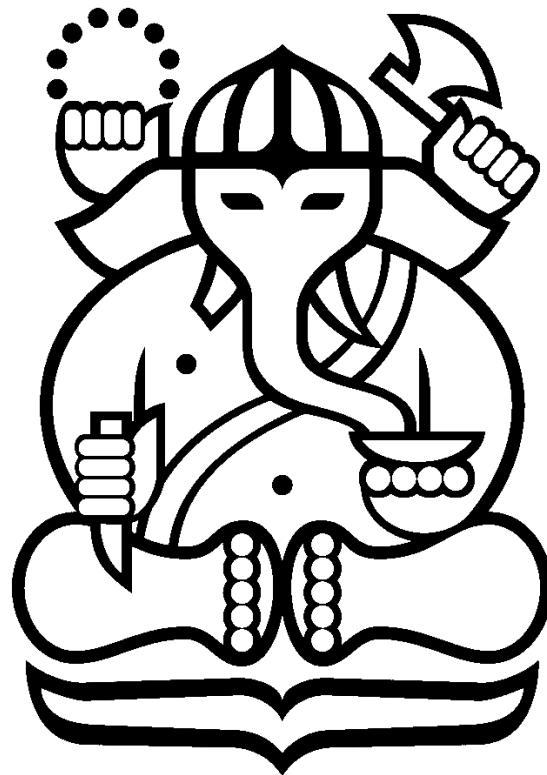


LAPORAN TUGAS INDIVIDU
IF5152 - Visi Komputer



Oleh:

Marvin Scifo Y. Hutahaean (13522110)

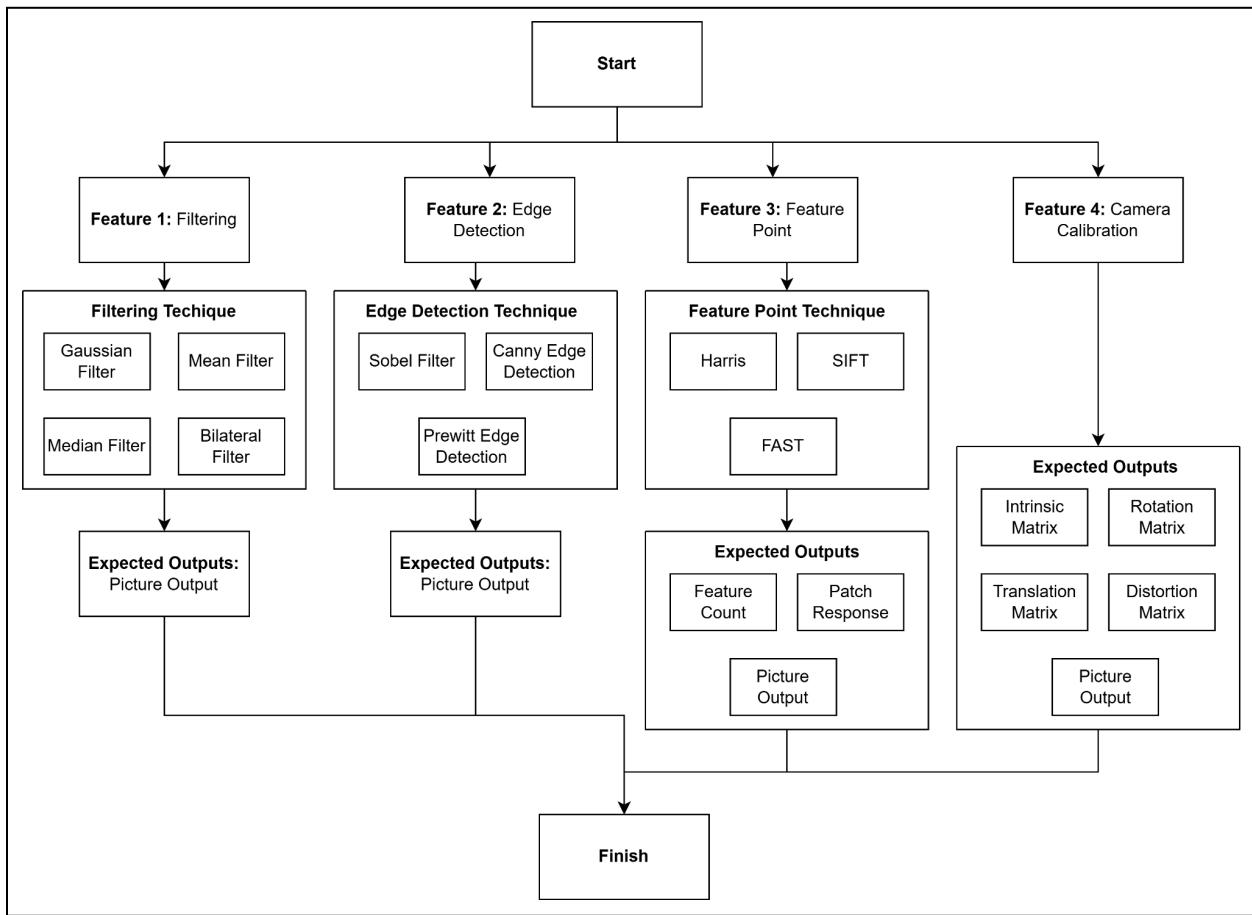
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2025

Daftar Isi

Daftar Isi.....	2
Diagram Pipeline.....	3
Proses & Hasil Tiap Fitur.....	4
Filtering.....	4
Edge Detection.....	4
Feature Point.....	4
Camera Calibration.....	4
Komparasi & Refleksi Pribadi.....	5
Komparasi.....	5
Preferensi Desain Pribadi.....	5

Diagram Pipeline



Proses & Hasil Tiap Fitur

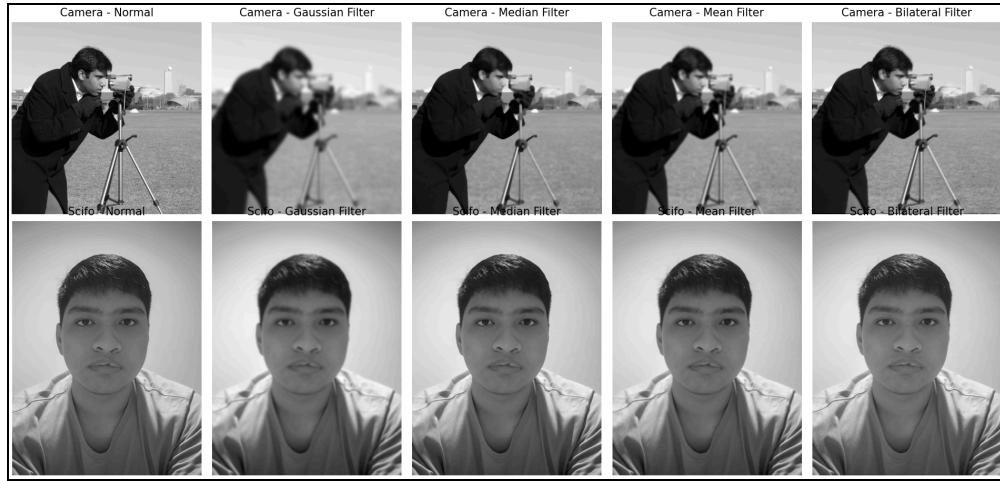
A. Filtering

Filtering merupakan teknik yang berfokus pada pemrosesan intensitas piksel untuk mengurangi gangguan (noise) atau menonjolkan detail tertentu pada citra. Proses ini dilakukan melalui operasi konvolusi antara citra dan kernel tertentu. Jenis filter yang sering digunakan meliputi *low-pass filter* seperti Gaussian dan median filter untuk menghasilkan efek perataan atau penghalusan, serta *high-pass filter* seperti Laplacian untuk menonjolkan tepi dan detail halus. Filtering menjadi dasar penting dalam berbagai aplikasi karena citra yang bersih dan stabil secara visual akan meningkatkan akurasi proses analisis berikutnya.

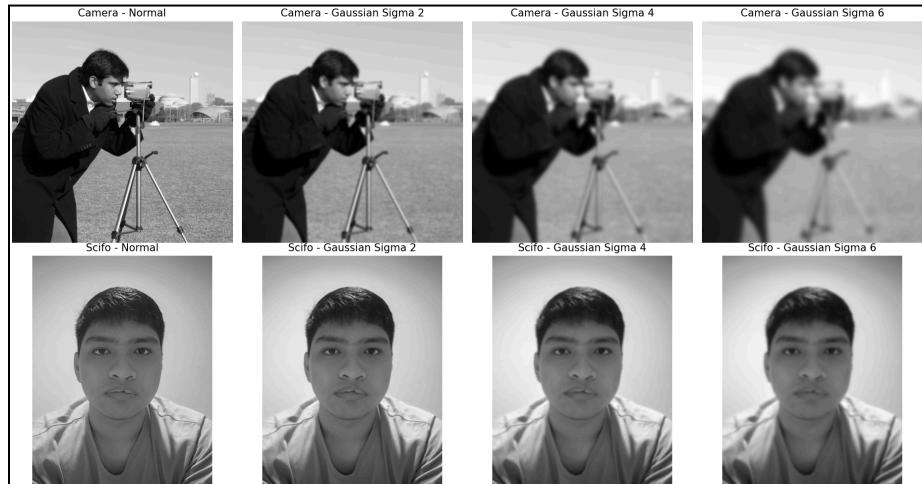
Sebelum menunjukkan hasil filtering, berikut adalah variabel-variabel yang diperhatikan ketika melakukan filtering.

Gambar:	data.camera(), Scifo.jpg		
Percobaan	Teknik Filtering	Sigma	Footprint
Percobaan 1	Gaussian	4	-
Percobaan 2	Median	-	4
Percobaan 3	Mean	-	4
Percobaan 4	Bilateral	-	-

Berikut adalah hasil yang didapat ketika menjalankan keempat percobaan filtering yang dijelaskan sebelumnya.



Berdasarkan hasil, terlihat Gaussian Filter memberikan hasil yang lebih halus dibandingkan yang lainnya. Hal ini terjadi karena Gaussian Filter memiliki nilai sigma yang bisa menaikkan tingkat blur dari hasil gambar. Hasil blur dari gambar oleh Gaussian Filter dapat didukung lebih lanjut oleh gambar dibawah ini.



Terlihat dengan menaikkan nilai sigma, gambar akan menjadi semakin blur apalagi dengan gambar dengan hitam putih yang bervariasi.

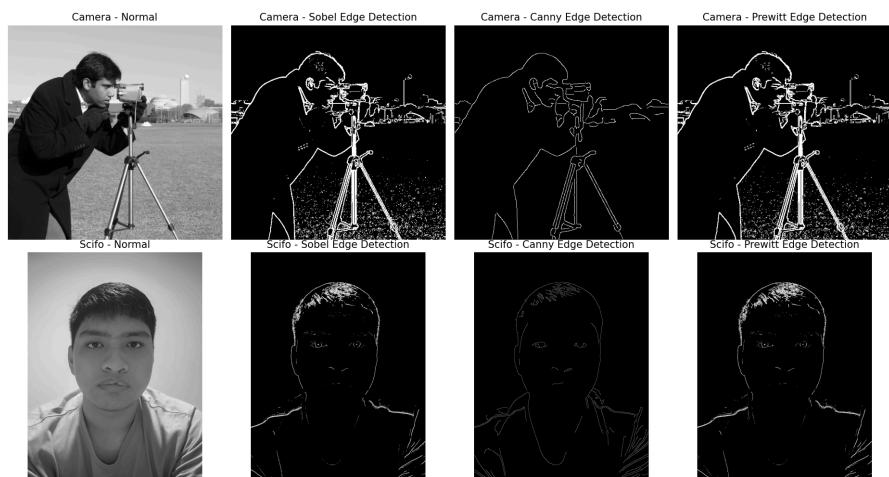
B. Edge Detection

Edge Detection bertujuan untuk menemukan batas-batas objek dalam citra dengan mendeteksi perubahan intensitas yang signifikan antara piksel. Tepi dianggap sebagai lokasi di mana terjadi peralihan tajam dari terang ke gelap atau sebaliknya. Metode deteksi tepi seperti *Sobel* dan *Prewitt* memanfaatkan perhitungan gradien untuk menentukan arah dan besarnya perubahan intensitas, sementara *Canny Edge Detector* menawarkan hasil yang lebih halus dengan melalui tahapan seperti perataan, non-maximum suppression, dan hysteresis thresholding. Informasi tepi yang diperoleh berperan penting dalam pengenalan bentuk, segmentasi objek, dan deteksi kontur.

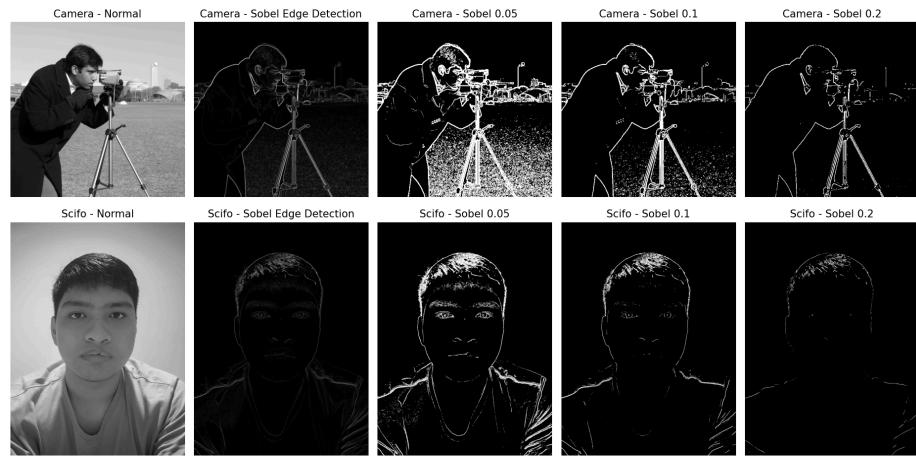
Sebelum menunjukkan hasil edge detection, berikut adalah variabel-variabel yang diperhatikan ketika melakukan edge detection.

Gambar:	data.camera(), Scifo.jpg			
Percobaan	Teknik Edge Detection	Threshold	Sigma	Sampling
Percobaan 1	Sobel	> 0.1	-	Default (1,1)
Percobaan 2	Canny	-	3	Default (1,1)
Percobaan 3	Prewitt	> 0.1	-	Default (1,1)

Berikut adalah hasil yang didapat ketika menjalankan keempat percobaan edge detection yang dijelaskan sebelumnya.



Jika dibandingkan, Canny lebih intens dalam membandingkan piksel yang merupakan edge dan piksel mana yang tidak sehingga noise sangat jarang ditemukan pada gambar. Sedangkan pada Sobel dan Prewitt, noise masih bisa terlihat pada bagian yang sebenarnya bukan edge. Jika threshold pada gambar dinaikkan, noise akan semakin berkurang namun bagian yang seharusnya merupakan edge pun kemungkinan bisa berkurang. Berikut adalah perbandingannya.



Berdasarkan gambar diatas, terlihat jika threshold dinaikkan, edge semakin sulit untuk dideteksi. Semakin sulit edge dideteksi, semakin sulit juga noise untuk dideteksi.

C. Feature Point

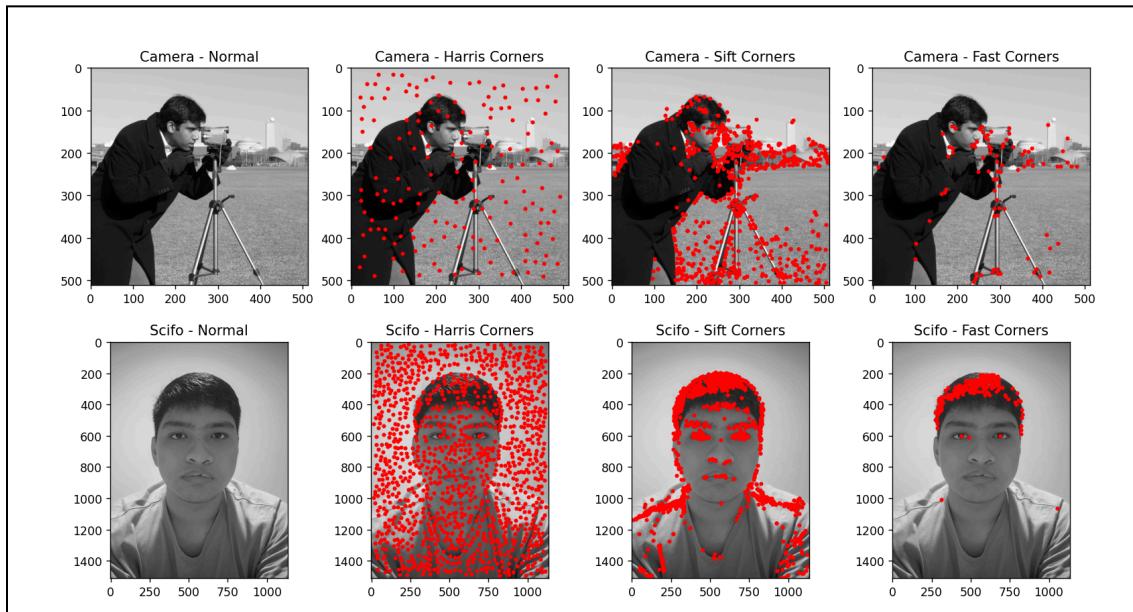
Feature Point Detection berfokus pada pencarian titik-titik khas di dalam citra yang dapat digunakan sebagai penanda unik. Titik-titik ini memiliki karakteristik stabil terhadap perubahan rotasi, skala, dan pencahayaan. Beberapa algoritma populer antara lain **Harris Corner Detector** yang mendeteksi sudut berdasarkan perubahan intensitas di dua arah, **SIFT (Scale-Invariant Feature Transform)** yang mencari fitur stabil di berbagai skala dan menghasilkan deskriptor numerik untuk keperluan pencocokan, serta **FAST (Features from Accelerated Segment Test)** yang menekankan kecepatan deteksi dengan memeriksa intensitas piksel di sekitar lingkaran tertentu. Feature

point digunakan secara luas dalam berbagai aplikasi seperti pelacakan objek, pencocokan citra, dan rekonstruksi 3D.

Sebelum menunjukkan hasil feature point detection, berikut adalah variabel-variabel yang diperhatikan ketika melakukan feature point detection.

Gambar:	data.camera(), Scifo.jpg
Percobaan	Teknik Feature Point Detection
Percobaan 1	Harris
Percobaan 2	SIFT
Percobaan 3	FAST

Berikut adalah hasil yang didapat ketika menjalankan keempat percobaan feature point detection yang dijelaskan sebelumnya.



Berdasarkan hasil yang didapat. Harris memberikan titik *feature point* yang paling acak jika dibandingkan dengan kakas lainnya. Jika dibandingkan dengan Harris, SIFT dan FAST memberikan lokasi *feature point* yang lebih berfokus di daerah gambar tertentu seperti bagian wajah, kamera, gedung, dll. Meskipun Harris terlihat memiliki banyak titik, SIFT lah yang memiliki hasil *feature point*

terbanyak jika dibandingkan dengan model yang lain. Berikut adalah hasil total *feature point* dan *patch response*-nya (untuk SIFT saja).

Feature Counts	
Camera - Harris	172 points
Camera - SIFT	791 points
Camera - FAST	89 points
Scifo - Harris	1220 points
Scifo - SIFT	1506 points
Scifo - FAST	190 points

Patch Response	
Camera	[0.0267 0.0399 0.0399 0.0334 0.0232 0.0232 0.0337 0.0438 0.0424 0.0183]
Scifo	[0.0495 0.034 0.0234 0.0238 0.0145 0.0239 0.0197 0.0267 0.0151 0.0142]

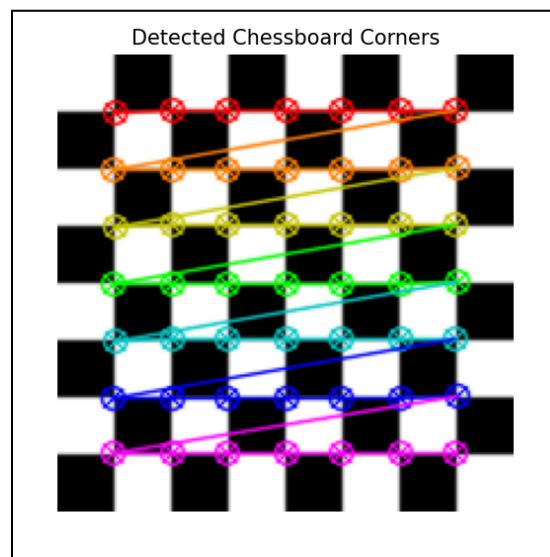
D. Camera Calibration

Camera Calibration merupakan proses untuk menentukan parameter yang menggambarkan hubungan antara koordinat tiga dimensi (dunia nyata) dengan proyeksi dua dimensi pada bidang gambar. Parameter ini terbagi menjadi tiga kategori: **intrinsic parameters** yang mendeskripsikan karakteristik internal kamera seperti panjang fokus dan titik pusat lensa, **distortion coefficients** yang memperbaiki penyimpangan akibat kelengkungan lensa, serta **extrinsic parameters** yang merepresentasikan posisi dan orientasi kamera terhadap dunia 3D melalui matriks rotasi dan vektor translasi. Melalui kalibrasi yang tepat, sistem kamera dapat menghasilkan representasi spasial yang akurat, yang menjadi dasar dalam aplikasi seperti fotogrametri, augmented reality, dan robotika.

Sebelum menunjukkan hasil camera calibration, berikut adalah variabel-variabel yang diperhatikan ketika melakukan camera calibration.

Gambar:	data.checkerboard()
Percobaan	Teknik Camera Calibration
Percobaan 1	Open CV Camera Calibration

Berikut adalah hasil yang didapat ketika menjalankan keempat percobaan camera calibration yang dijelaskan sebelumnya. Untuk gambar, camera calibration berhasil mendeteksi *corner* seperti tujuan yang ingin dicapai pada bagian C sebelumnya yaitu Feature Point. Berikut adalah hasil gambarnya.



Berikut juga hasil dari matriks-matriks yang dibutuhkan untuk mengubah visualisasi 3D menjadi gambar 2D yaitu matriks intrinsik, rotasi, translasi, dan koordinat 3 dimensi itu sendiri (tidak didapat, ini ditentukan sendiri). Tidak hanya itu, didapat juga koefisien distribusi.

```
Camera Matrix (K):
[[9.11083132e+07 0 9.95035401e+01]
 [0 9.11182170e+07 9.95046017e+01]
 [0 0 1.00000000e+00]]
f_x = 91108313.21, f_y = 91118217.03
c_x = 99.50, c_y = 99.50

==== Distortion Coefficients ====
k1, k2, p1, p2, k3 = [ 9.11276424e-04 8.77593047e-16 5.45767782e-03 -3.47459583e-03
 | 9.30783685e-28]

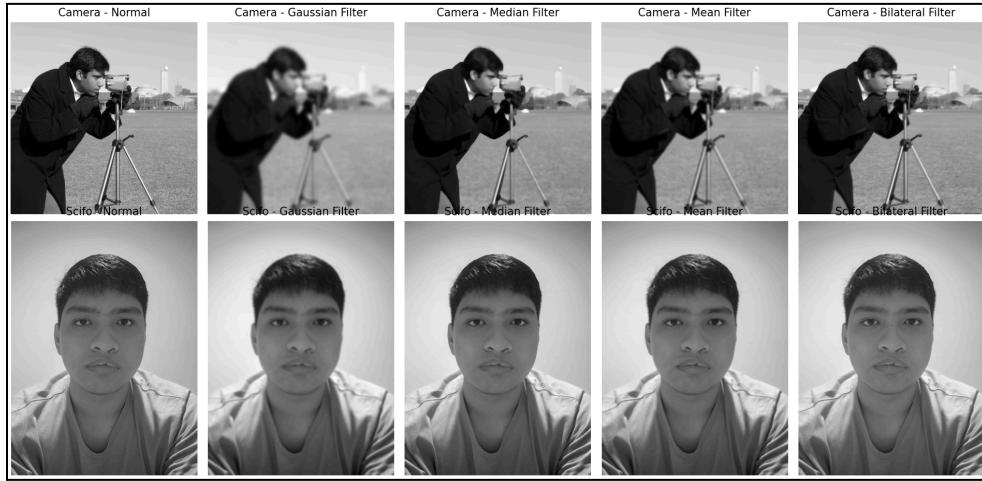
==== Extrinsic Parameters (for this image) ====
Rotation matrix (R):
[[ 9.99997205e-01 -5.75823115e-06 -2.36419117e-03]
 [-3.03363064e-05 9.99883454e-01 -1.52668626e-02]
 [ 2.36400354e-03 1.52668916e-02 9.99880660e-01]]

Translation vector (t):
[-3.00011593e+00 -2.99974341e+00 3.64432229e+06]
```

Komparasi & Refleksi Pribadi

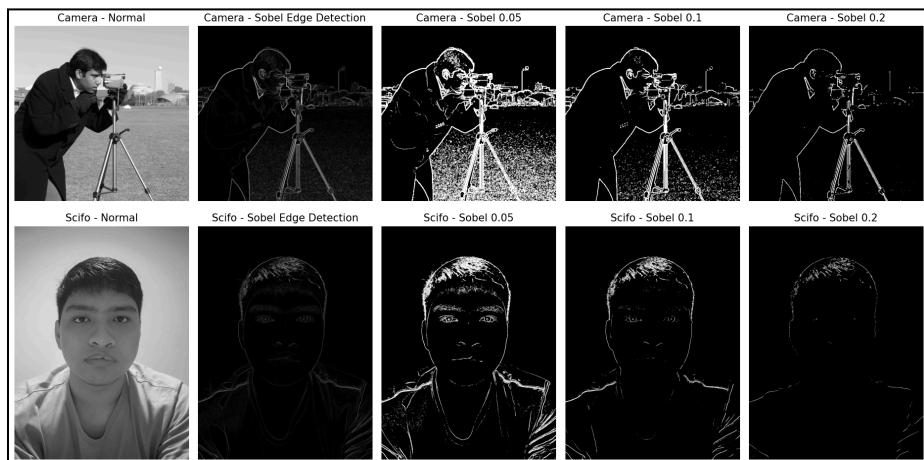
A. Komparasi

1. Filtering



Berdasarkan gambar, terlihat bahwa gambar dibawah cukup sulit untuk diidentifikasi bagian mana yang blur. Hal ini disebabkan karena komposisi gambar diatas lebih bervariasi dan banyaknya edge pada gambar yang pertama seperti yang disebutkan sebelumnya pada Feature Point Extraction.

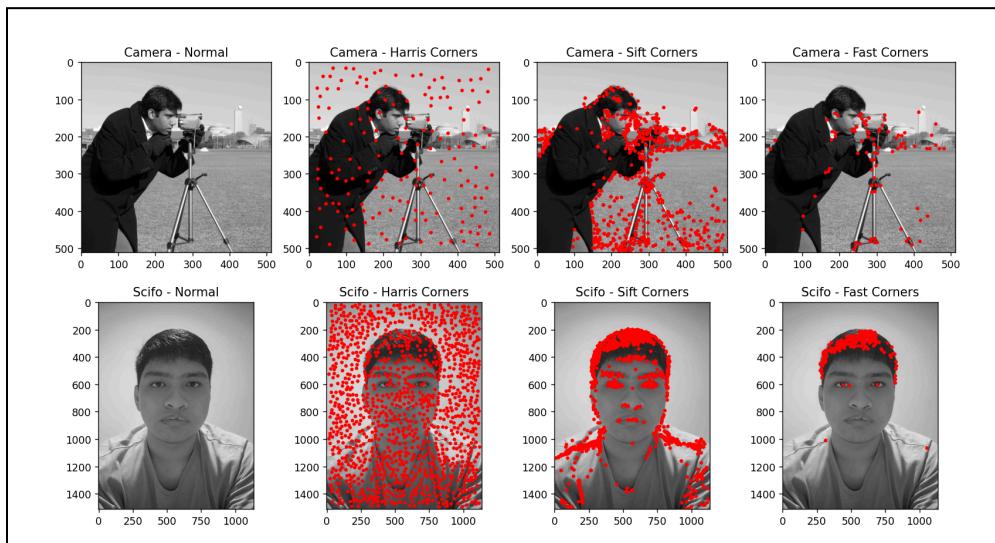
2. Edge Detection



Berdasarkan gambar, terlihat bahwa gambar diatas memiliki noise yang lebih banyak dibandingkan gambar yang dibawah. Hal ini

disebabkan karena gambar yang diatas memiliki edge dan brightness yang sangat beragam sehingga noise lebih banyak terdeteksi. Beda dengan gambar yang dibawah yang brightnessnya lebih monoton dibandingkan yang atas.

3. Feature Point Detection



Berdasarkan gambar, terlihat lokasi feature point relatif sama. Hanya saja gambar dibawah memiliki *feature point* yang lebih banyak dibandingkan gambar diatas. Hal ini bisa terjadi karena gambar di bawah lebih detil karena ketika ditangkap menghasilkan gambar dengan resolusi yang cukup tinggi sehingga feature point dideteksi dengan jumlah yang lebih banyak.

4. Camera Calibration

Bagian ini hanya menggunakan satu gambar saja sehingga tidak akan ada komparasi.

B. Refleksi Pribadi

Terkait pemilihan metode, terdapat metode tertentu yang bisa memberikan hasil terbaik tetapi dengan *tradeoff* yang lebih juga. Pengembang harus berhati-hati dalam melakukan *tuning* akan objek CV2 atau skimage supaya

ketika memproses gambar, banyak informasi yang bisa didapat tetapi sedikit noise yang didapat juga. Keseimbangan dalam melakukan *tuning* adalah hal yang cukup penting untuk dilatih sebagai pengembang fitur yang berhubungan dengan Computer Vision. Beberapa *error* bisa terjadi karena adanya konflik konfigurasi gambar atau kakas yang digunakan seperti ukuran gambar, koordinat gambar, dll sehingga perlu dikonfigurasi ulang sehingga memenuhi persyaratan kakas (OpenCV atau skimage). Dan hal itu pula yang membuat adanya gaya pengembangan yang cukup unik yaitu konversi gambar yang cocok dengan skimage dengan yang cocok dengan OpenCV ketika kakas OpenCV lebih diperlukan dibandingkan skimage. Selain itu, kendala-kendala lainnya yang dihadapi adalah kesulitan dalam eksplorasi kakas Computer Vision dan mencoba untuk mengaplikasikan konfigurasi yang sama semua kakas yang sama (Contoh: sampling terhadap Canny Edge Detection (Tidak bisa dilakukan)). Dengan itu, eksplorasi dari beberapa konsep Computer Vision ini mengajarkan banyak ilmu yang nantinya bisa digunakan ke konsep yang mungkin lebih kompleks.

Lampiran

[scifo04/13522110_IF5152_TugasIndividuCV](#)