Fundamentos Matemáticos de la Visión Robótica
Trabajo Práctico № 3: Sensor

Vázquez Lareu, Román. Facultad de Ingenieria de la UBA 19/015/2020

Introducción

En el siguiente informe, a partir de un conjunto de imágenes tomadas tapando el sensor en su totalidad, se realizará un análisis estadístico del ruido indicando media, desvío y comentando las posibles fuentes del mismo.

Desarrollo

En primer lugar, se importarán las librerías necesarias para el análisis de la imagen: **numpy**, **cv2**, **matplotlib.pyplot**.

A continuación se leerá la lista de imágenes a procesar, mediante la instrucción **glob**, la cual permite crear una lista con nombres de archivos en un directorio específico Se obtendrá una lista con todas las direcciones de los archivos buscados. Ya se puede notar que las imágenes, a pesar de haber sido tomadas tapando el sensor en su totalidad, varían en su tamaño(kb).

Dado que se busca trabajar las imágenes, se las lee en memoria, se las pasa al espacio conocido RGB y, de ser necesario, se las configura de manera de quedar apaisadas.

Captura de Notebook № 1: Paso a RGB y cambio de orientación de ser necesario

```
imgs = []
for fname in img_fnames:
    img = cv2.imread(fname)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    if img.shape[0] > img.shape[1]:
        img = cv2.transpose(img)
    imgs.append(img)
```

En la siguiente celda, a partir de un promedio de las imágenes, se busca ver el ruido en el campo del sensor. Para verlo con claridad, se normaliza la imagen promedio final.

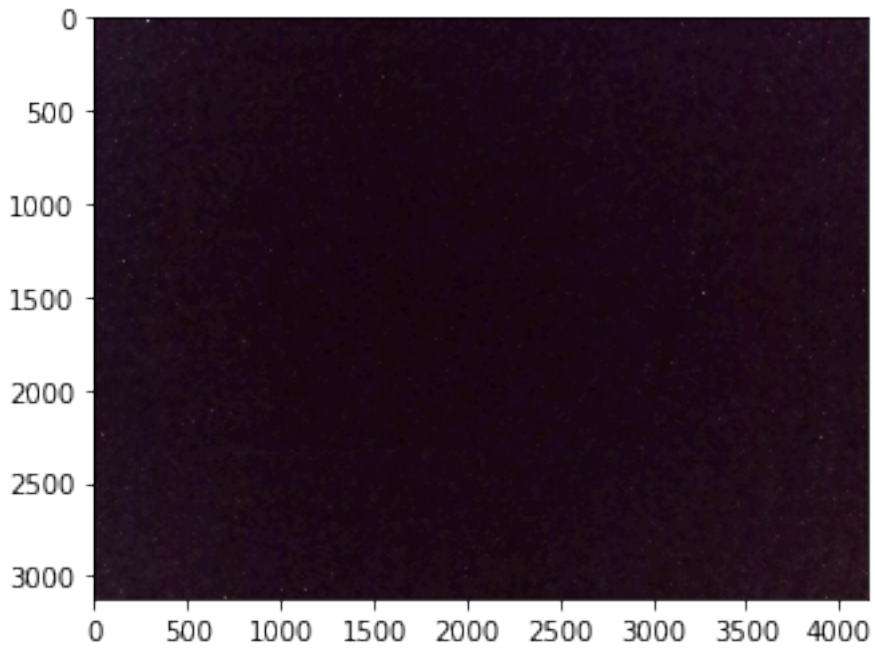
Captura de Notebook № 2: Muestras de color

```
1 img_promedio = np.zeros_like(imgs[0])
2 n = len(imgs)
3 for img in imgs:
4
5     img_promedio = img_promedio + img/n
6 img_promedio = img_promedio.astype(np.uint8)
7
8 img_promedio_norm = cv2.normalize(img_promedio, 0, 255, norm_type=cv2.↩
    NORM_INF)
9 plt.imshow(img_promedio_norm)
```

Al normalizarla, se llevan los pixeles distintos de cero a 255, de esta manera se acentúan aquellos donde a pesar de encontrarse el sensor tapado, perciben cierta intensidad. Se obtiene la siguiente imagen:

Captura de Notebook № 3: Imagen promedio normalizada

1



2

A diferencia de la imagen del video de la cátedra, el ruido a simple vista se ve parejo a lo largo de toda la lente. Aunque si se sube mucho el brillo, se observa una mayor intensidad hacia los laterales del lente en comparacion con el centro.

Con la idea de ver el ruido por color por pixel, se apilan las imágenes y se calcula la media y el desvio. Para visualizarlo, se dibujan contornos.

Captura de Notebook № 4: Apilamiento de imágenes. Cálculo de media y desvío

```
imgs_np = np.stack(imgs)
img_media = np.mean(imgs_np, axis=0)
img_std = np.std(imgs_np, axis=0)

print(imgs_np.shape)
print(img_media.shape)
print(img_std.shape)
```

En el output se observa cómo a las imágenes apiladas se les agrega una dimensión que indica la cantidad de imágenes con las que se cuenta. Al contrario, la media y el desvío no poseen esa posicion extra.

Captura de Notebook № 5: dimensión de imágenes apiladas. Media y desvío

```
(5, 3120, 4160, 3)
(3120, 4160, 3)
(3120, 4160, 3)
```

Con el objetivo de visualizarlo, se dibujan contornos para la media y desvío de cada color, mediante la siguiente función:

Captura de Notebook № 6: dibujo de contornos

```
def dibujar_contorno(mat):
    fig = plt.figure()
    X, Y = np.meshgrid(range(ancho), range(alto))
    Z = mat

    dec = 16

    fig = plt.figure(figsize=(16,12))
    cp = plt.contourf(X[::dec], Y[::dec], Z[::dec])
    fig.colorbar(cp)
    plt.show()
```

Se imprimen por pantalla los contornos de media y desvío para cada color.

Captura de Notebook № 7: impresión de contornos por pantalla

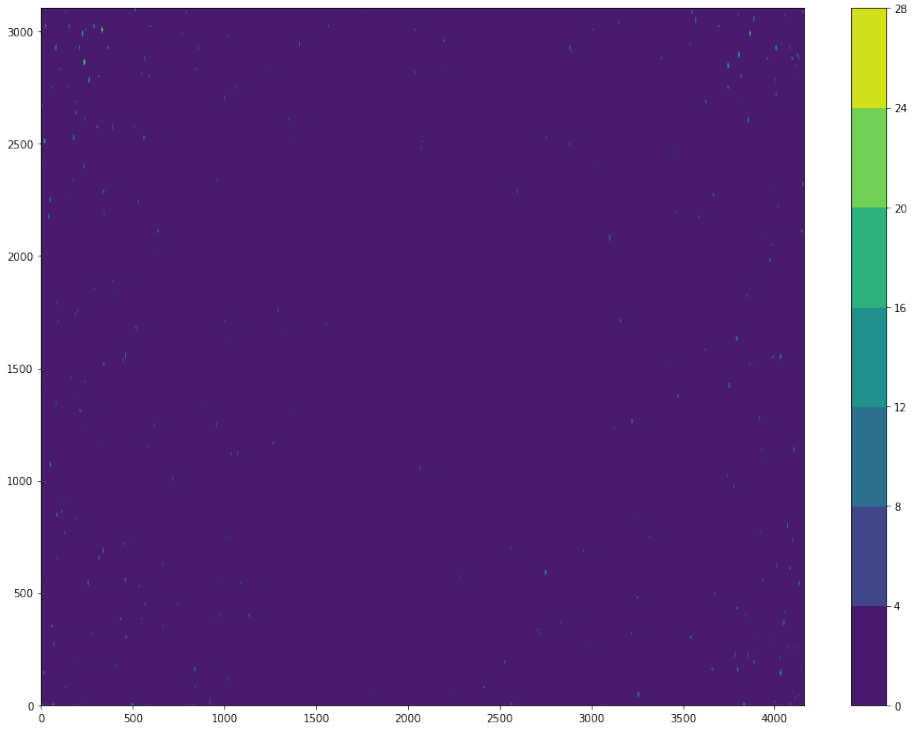
```
print('Std Rojo')
dibujar_contorno(img_std[:,:,0])
print('Std Verde')
dibujar_contorno(img_std[:,:,1])
```

```
print('Std Azul')
dibujar_contorno(img_std[:,:,2])
print('Media Rojo')
dibujar_contorno(img_media[:,:,0])
print('Media Verde')
dibujar_contorno(img_media[:,:,1])
print('Media Azul')
dibujar_contorno(img_media[:,:,2])
```

Output de los contornos de media y desvío:

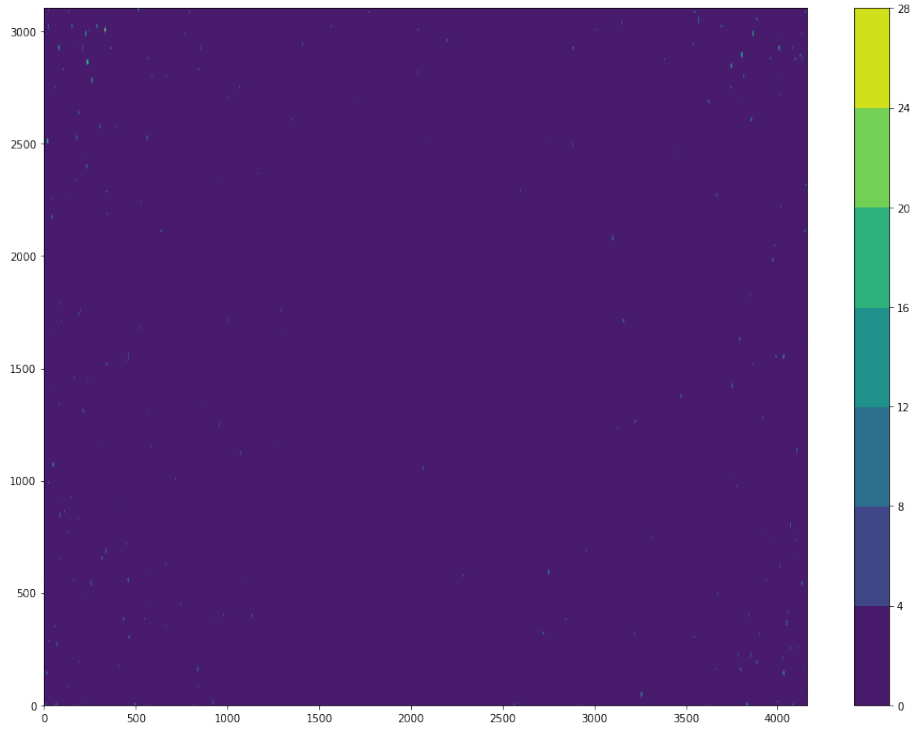
Captura de Notebook № 8: Desvío Rojo

```
1
```



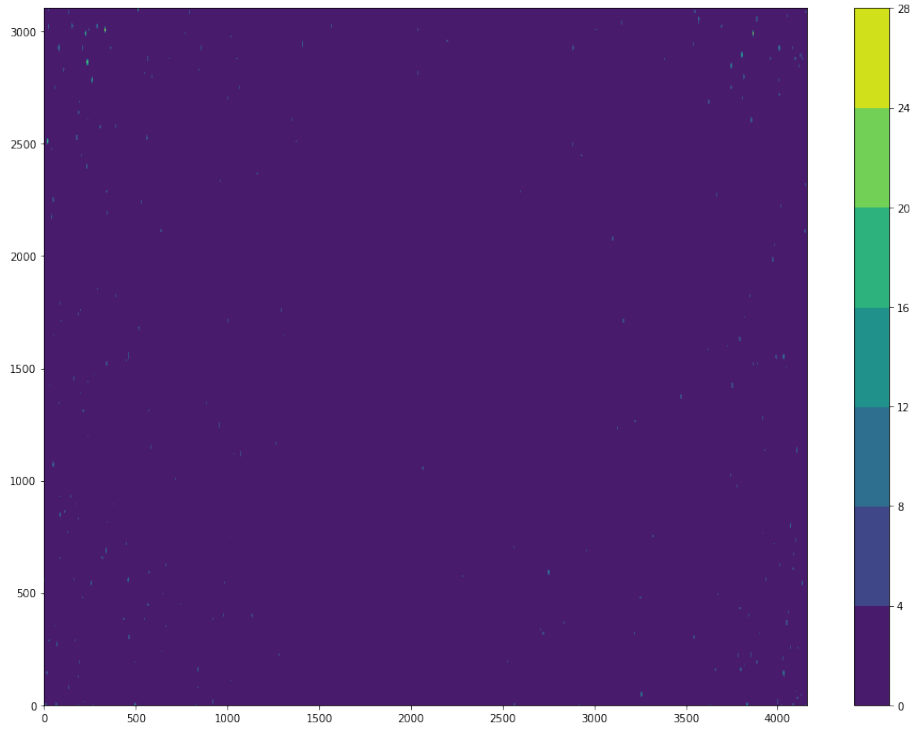
Captura de Notebook № 9: Desvío Verde

1



Captura de Notebook № 10: Desvío Azul

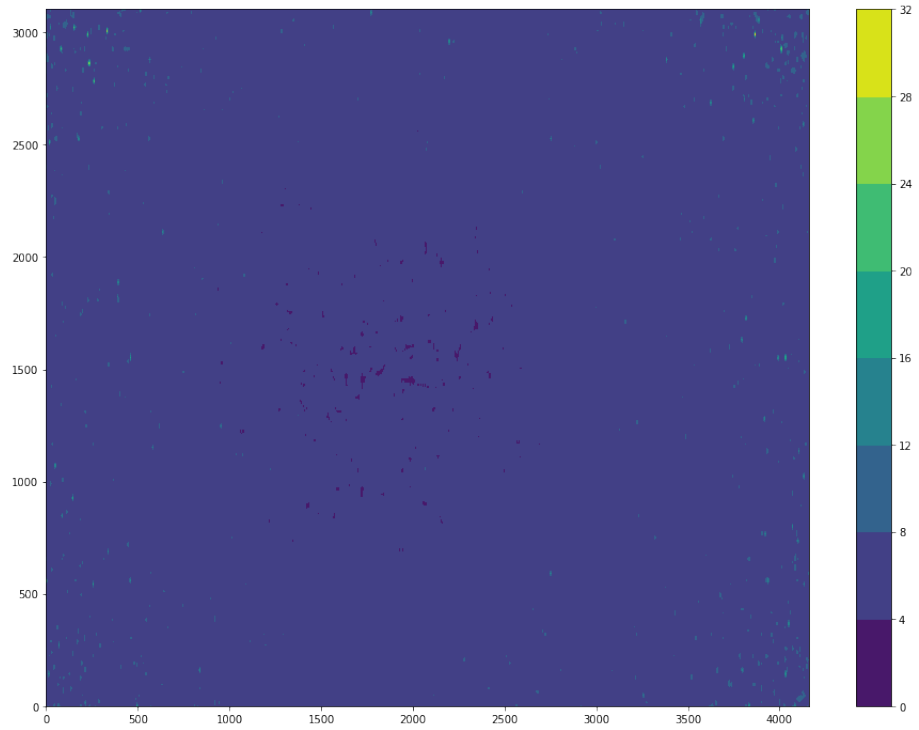
1



A pesar de verse bastante uniformes, con valores de intensidad entre 0 y 4, se observan destellos que alcanzan valores de 20.

Captura de Notebook № 11: Media Rojo

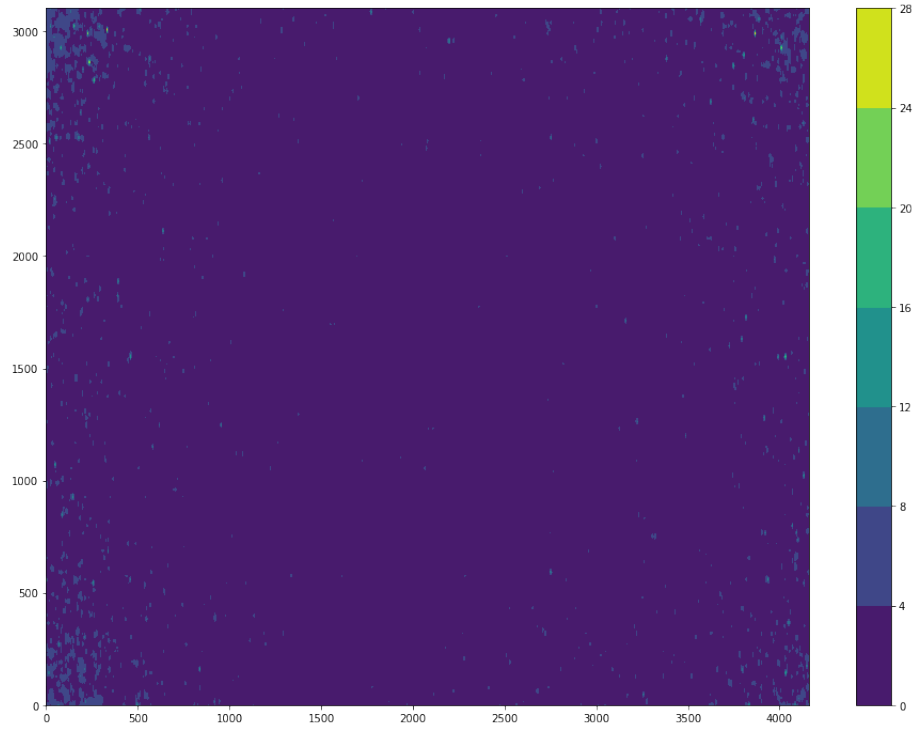
1



Se ve como, contrario a la lógica, la media del rojo general se encuentra entre 12 y 20.

Captura de Notebook № 12: Media Verde

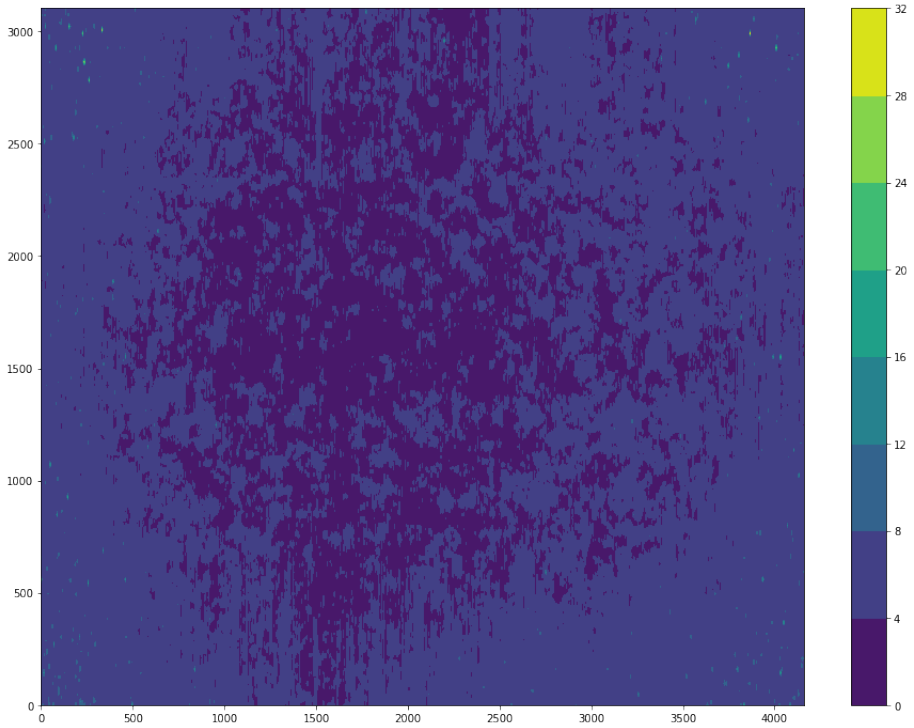
1



La media del verde, si bien no es tan uniforme como la del rojo, presenta una media general menor, entre 0 y 4.

Captura de Notebook № 13: Media Azul

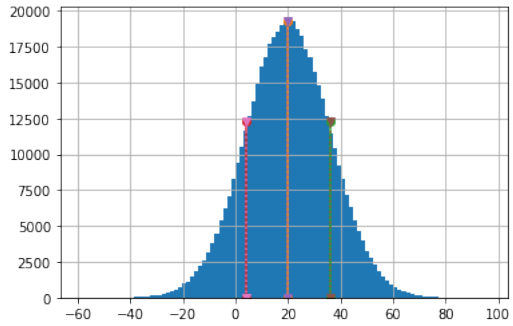
1



Se toma un punto cercano a la esquina con media 20 y desvio, en el peor de los casos, 16. Si se ve la campana de Gauss, hay valores por debajo y por encima del cero, cuando los esperado eran todos en el cero.

Captura de Notebook № 14: Campana de Gauss

1



Finalmente, la media general del azul es la menos uniforme de las tres, presentado una forma casi geométrica con valores entre 0 y 4 en el centro, y a medida que se aleja del mismo, los valores se encuentran entre 8 y 20. Si se analizan en conjunto la media y el desvío (en el caso del azul que es el más notorio), se ve que si bien es bastante uniforme, hacia los laterales de la lente el desvío es ampliamente mayor al del centro: 0 a 4 contra 12 a 24. Viendolo en el gráfico de la media, se puede intepretar que en el centro del lente el 68 % de los valor se econtrarán entre 0 y 4 con un $\pm 4 (setienecomopisoel0). Sisevahacialoslaterales, lamediadeentre8y20tendrasuvezundesvoque$

$24. Se podrá decir que el 68\,\% de los pixeles en los laterales tendrán un valor de entre 44 y 0 (piso). Como conclusión tempran$

Con la idea de ilustrar mejor lo que ocurre y tener idea mejor idea de una fuente posible de este fenómeno, a continuación se presentan las medias de cada color pero las imágenes originales tomadas fueron hechas con gran angular, a diferencia de la estándar.

Estándar:

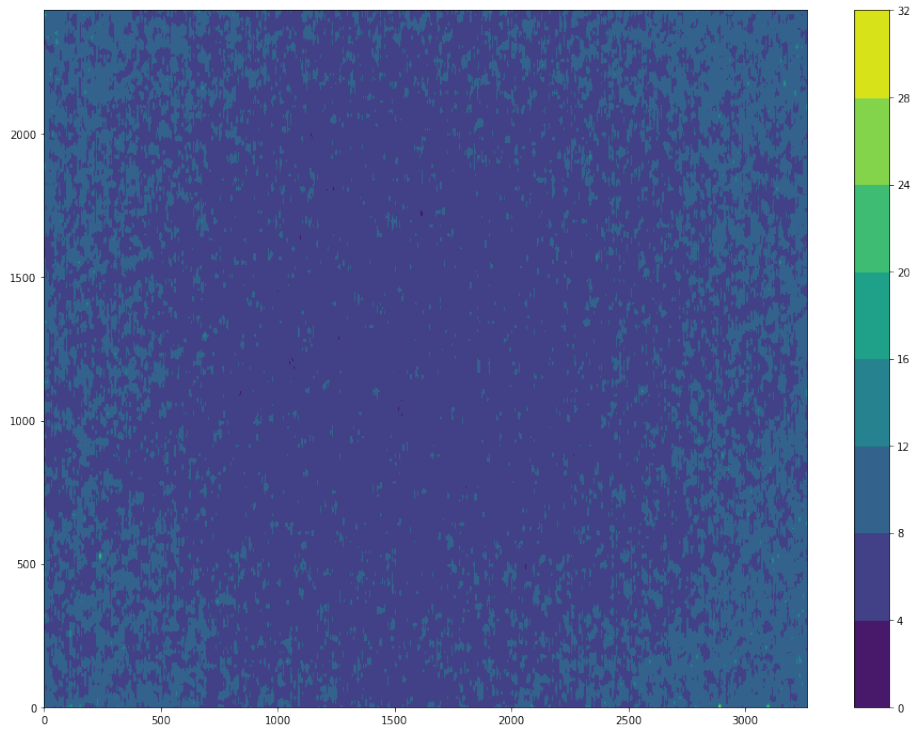
- 13.0MP 3120x4160 2.3 MB
- f/2.0 1/14 3.54mm ISO1050

Gran angular:

- 8.0MP 2448x3264 1.4 MB
- f/2.2 1/14 1.66mm ISO703

Se sabe que a mayor distancia focal, menor ángulo de visión. En este caso, la distancia focal del gran angular es menos de la mitad que la del estándar.

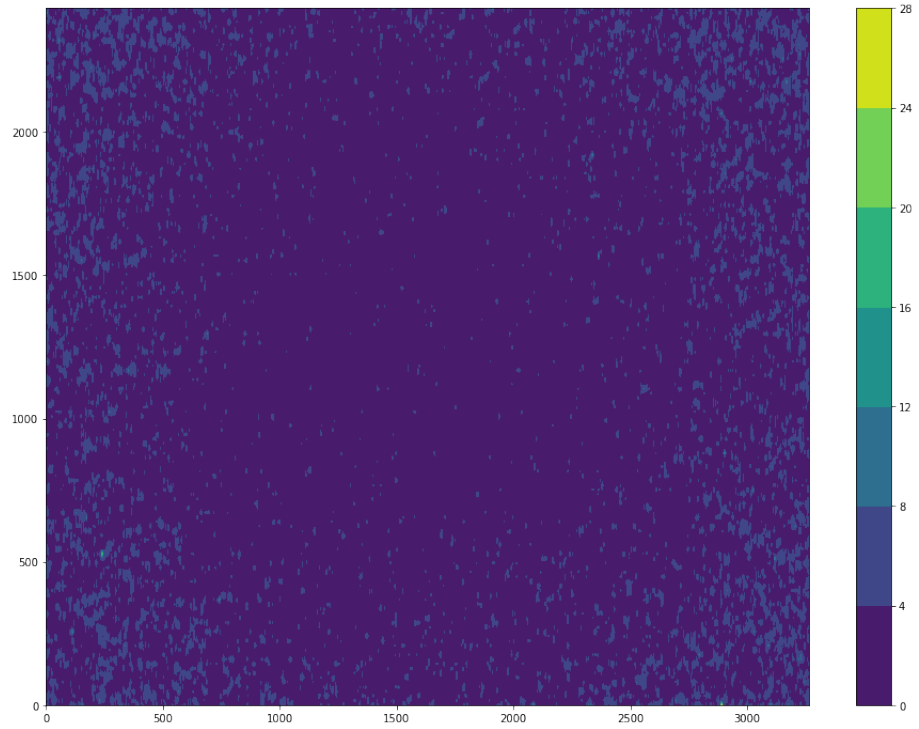
Captura de Notebook № 15: Media Rojo Gran Angular



1

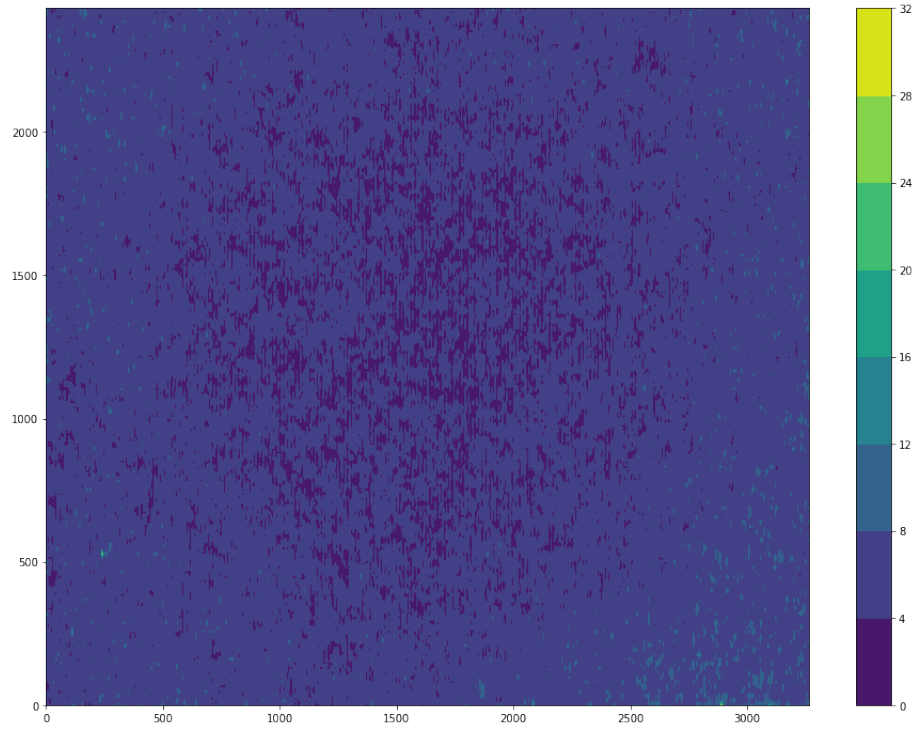
Captura de Notebook № 16: Media Verde Gran Angular

1



Captura de Notebook № 17: Media Azul Gran Angular

1



En las medias del gran angular se observa el mimso fenómeno que en las estándar pero amplificado.

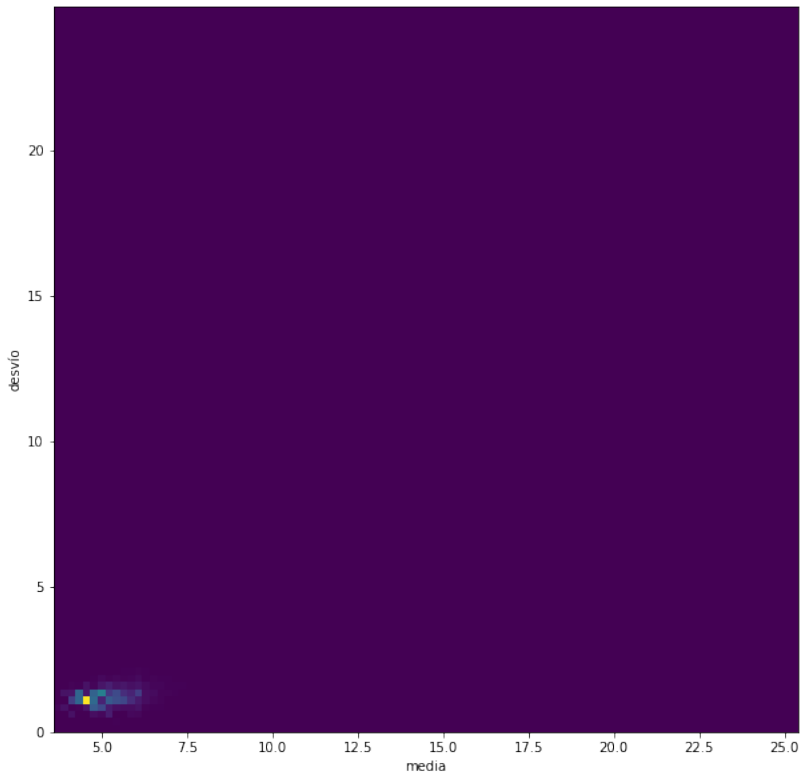
A continuación se ven estadísticas globales de ruido de todos los píxeles por color:

Captura de Notebook № 18: estadísticas globales de ruido

```
dec = 100

todos_los_rojos_std = np.ravel(img_std[:,:,0])
todos_los_rojos_media = np.ravel(img_media[:,:,0])

todos_los_verdes_std = np.ravel(img_std[:,:,1])
todos_los_verdes_media = np.ravel(img_media[:,:,1])

todos_los_azules_std = np.ravel(img_std[:,:,2])
todos_los_azules_media = np.ravel(img_media[:,:,2])

plt.figure(figsize=(10,10))
plt.xlabel('media')
plt.ylabel('desv o')
cb = plt.hist2d(todos_los_rojos_media[::dec], todos_los_rojos_std[::dec], bins=100)


plt.figure(figsize=(10,10))
plt.xlabel('media')
plt.ylabel('desv o')
cb = plt.hist2d(todos_los_verdes_media[::dec], todos_los_verdes_std[::dec], bins=100)


plt.figure(figsize=(10,10))
plt.xlabel('media')
plt.ylabel('desv o')
cb = plt.hist2d(todos_los_azules_media[::dec], todos_los_azules_std[::dec], bins=100)
```

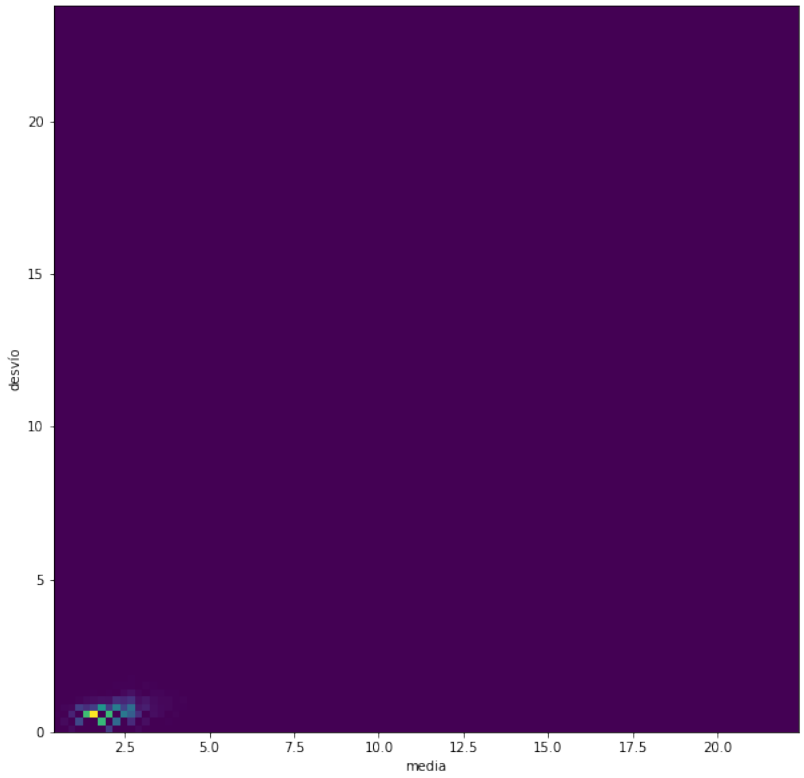
Obteniendo la siguiente salida por pantalla, donde se observa un histograma 2d, con la media y el desvío como ejes

Captura de Notebook № 19: histograma 2d rojo



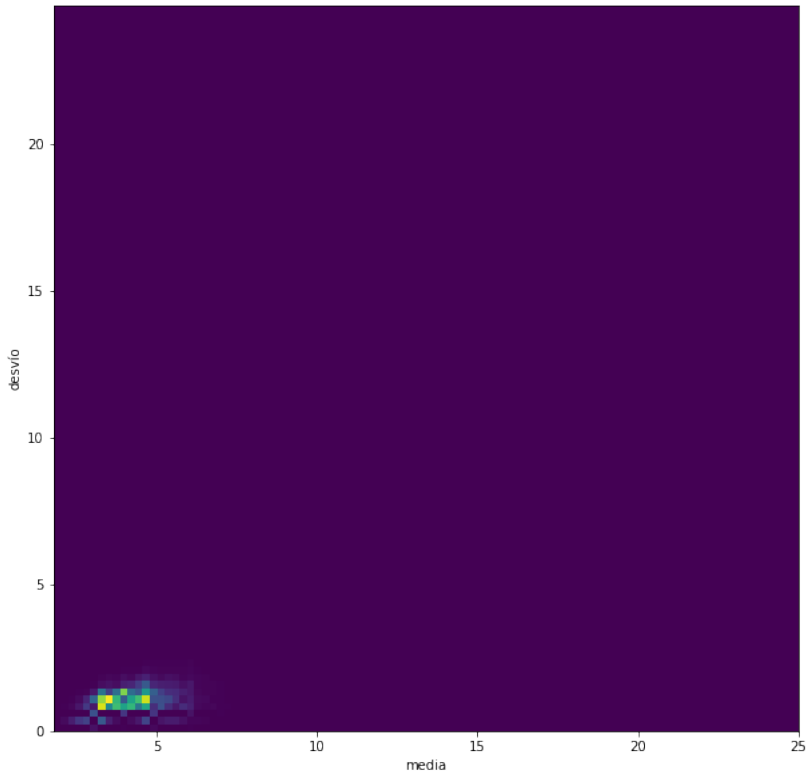
1

Captura de Notebook № 20: histograma 2d verde



1

Captura de Notebook № 21: histograma 2d azul



1

De estos gráfico se observa que hay pixeles con medias y desvíos distintos. A su vez, el tratamiento de los pixeles parece diferir según su posición, viendose en mayor mediad en el caso del azul y en menor medida en el rojo

Para terminar se verá un histograma que representará cuantos pixeles caen por rango de color.

Captura de Notebook № 22: realización de histograma

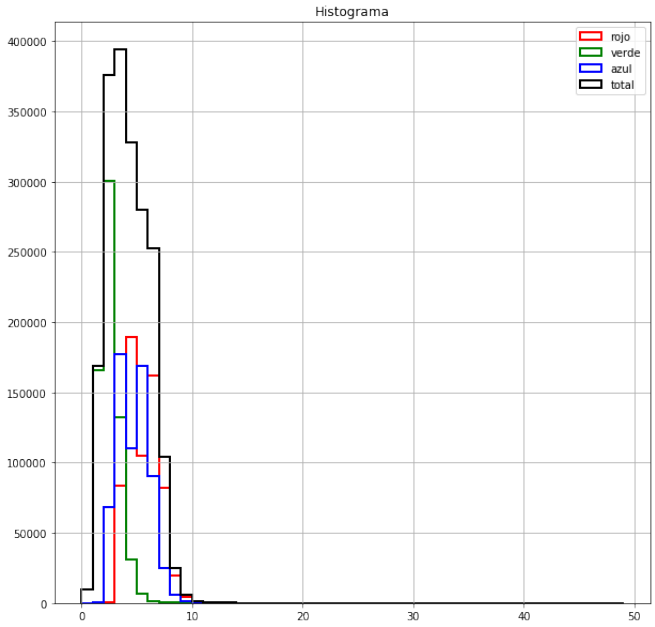
```
1  dec = 100
2  plt.figure(figsize=(10,10))
3  plt.title('Histograma')
4  todos_los_rojos = np.ravel(imgs_np[:,:,:,0])
5  todos_los_verdes = np.ravel(imgs_np[:,:,:,1])
6  todos_los_azules = np.ravel(imgs_np[:,:,:,2])
7  plt.grid()
8  i_max = 50
9  _ = plt.hist(todos_los_rojos[::dec], bins=range(i_max), color='red',↩
       histtype='step', linewidth=2.0)
10 _ = plt.hist(todos_los_verdes[::dec], bins=range(i_max), color='green', ↩
       histtype='step', linewidth=2.0)
11 _ = plt.hist(todos_los_azules[::dec], bins=range(i_max), color='blue', ↩
       histtype='step', linewidth=2.0)
12
13 _ = plt.hist(np.ravel(imgs_np)[::dec], bins=range(i_max), color='black', ↩
       histtype='step', linewidth=2.0)
```

```
14 plt.legend(['rojo', 'verde', 'azul', 'total'])
```

Se obtiene la siguiente salida:

Captura de Notebook № 23: histograma

```
1
```



Teóricamente se querría obtener todos en cero (ya que se tapó el sensor en su totalidad), sin embargo se obtienen valores distintos de cero y una distribución asimétrica, no Gaussiana. A su vez, la estádistica que siguen los colores es distinta entre si. Además se puede ver el piso de ruido a tener en cuenta al momento de capturar una imágen con muy baja iluminación.

Conclusión

En conclusión, a pesar de tapar el sensor en su totalidad al momento de capturar una imagen, aun se registran niveles de intensidad. Esto se puede deber tanto a un error de procedimiento (no tapar bien la lente) como al viñetado, donde aquellos elementos más alejados del eje óptico reciben menos luz debido a como fue realizado el sistema óptico. Esto varía de acuerdo a la apertura de la lente. Se puede ver al comparar gran angular con estándar.