

Procesamiento de Imagenes

Practica 2

Indice

Separación y alineación de los canales R, G, B(3,4)
Alineación con descomposición piramidal(5-7)

Definición de funciones desarrolladas:

def gaussiana(sigma):

función para la obtención del filtro gaussiano.

def lowFilter(img,filtro):

función para la convolución del filtro pasa bajos.

def getChannel(img):

función para separar los canales de la imagen.

def cropImage(img,point,extents):

función para la obtención del tamaño a escala del canal.

def cropFinalImage(img,point,extents):

función para recortar las filas y columnas sobrantes de la imagen final.

def correlationMatrix(img1,img2):

función de correlación entre canales.

def generateNewRGB(s,rgbCopia,diffila,difcolumna):

función para desplazar los canales según desplazamiento en fila y columna.

def alineacion(image,rgbCopia,filtro,escala):

función de alineación que hace uso de las funciones anteriormente mencionadas para la realización de la alineación según desplazamiento.

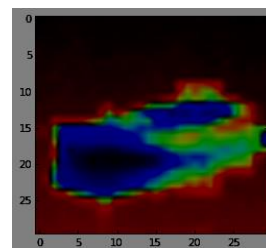
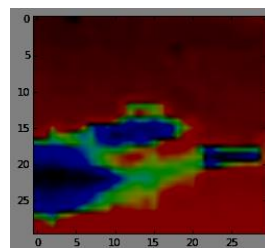
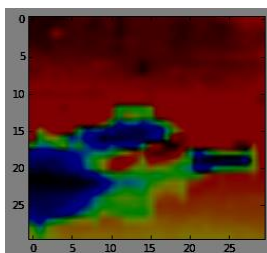
def piramide(imagen,rgbCopia,filtro,escala):

función para la alineación con respecto a una pirámide de imágenes.

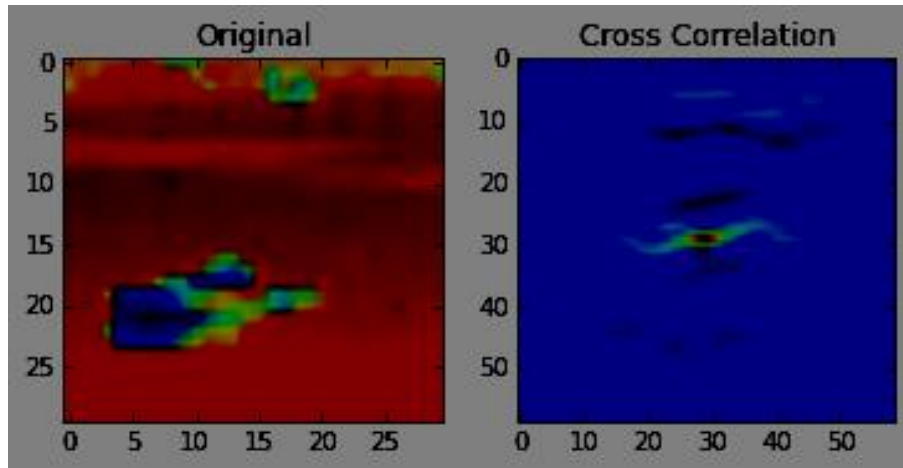
Separación y alineación de los canales R, G, B :

En la separación de imágenes no hemos tenido problema alguno, sin embargo para la parte de alinear las imágenes resultantes de la separación hemos encontrado problemas como.

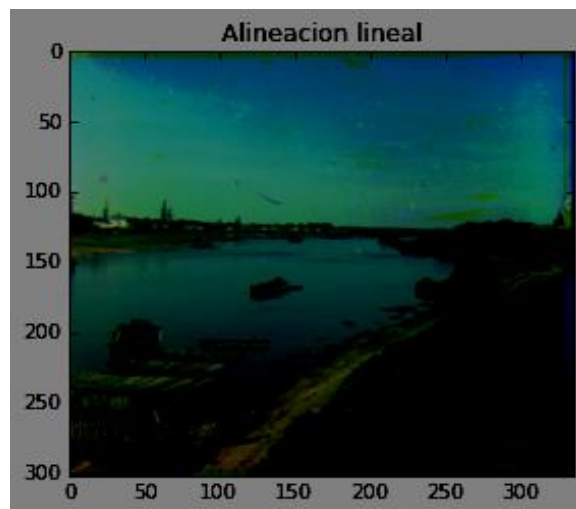
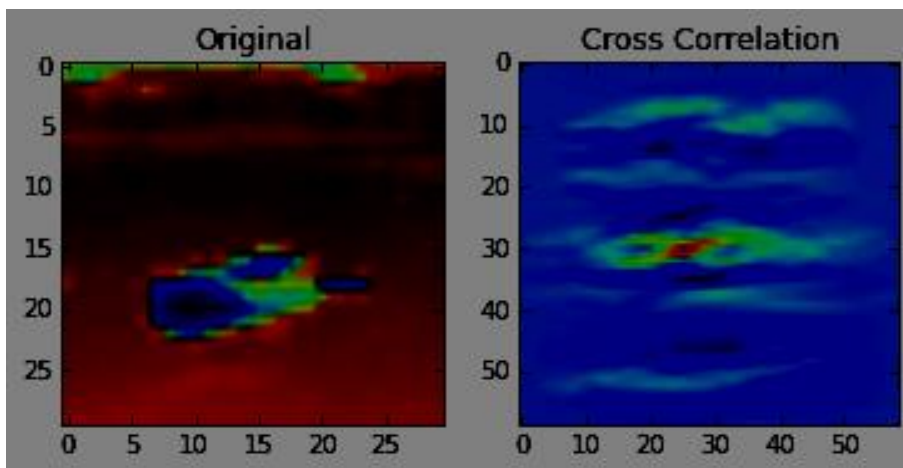
- Encontrar la diferencia de posiciones entre las imágenes de cada canal. Solventado con la función `signal.correlate2d`, que sin embargo no nos proporciona el valor de las distancia para cada eje correspondiente.
- La obtención de las diferencias exactas para cada eje. Esto lo hemos logrado primero restando del valor obtenido por el `correlate2d` la media del mismo, luego usando el máximo de lo obtenido del resultado anterior para poder encontrar la posición exacta para cada canal.



Desplazamiento en la altura 0
Desplazamiento en el ancho 0



Desplazamiento en la altura 2
Desplazamiento en el ancho -8

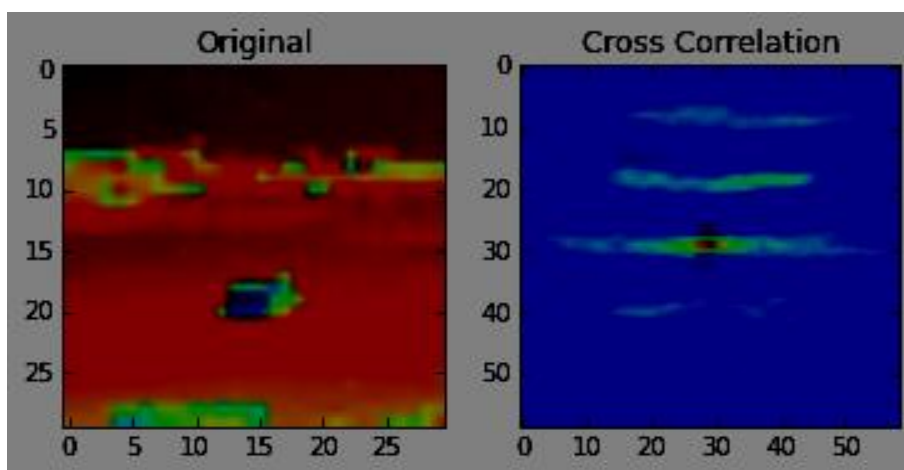


Alineación con descomposición piramidal:

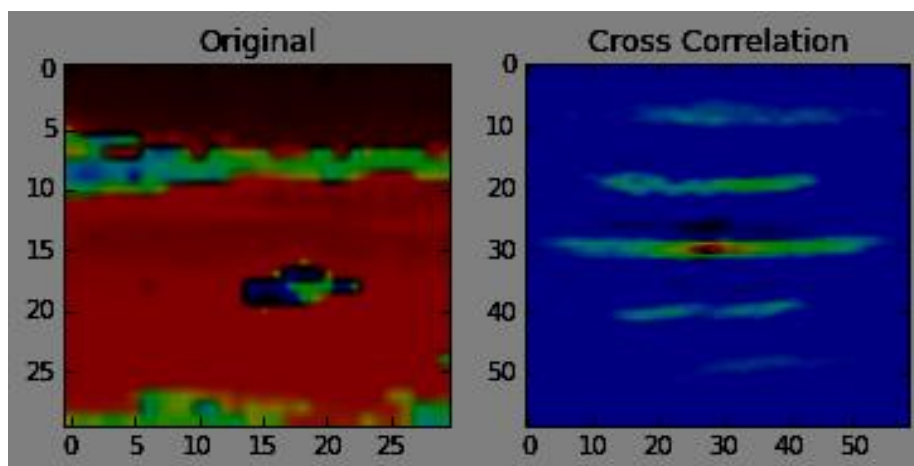
En este caso hemos tenido los siguientes problemas:

- La obtención del desplazamiento entre los canales, hemos usado la cross correlation con respecto a la gradiente, usando la piramide gaussiana.
- La realización del desplazamiento entre canales para la correcta alineación de los canales. Usando la función roll de numpy desplazamos las matrices según fila y columna.

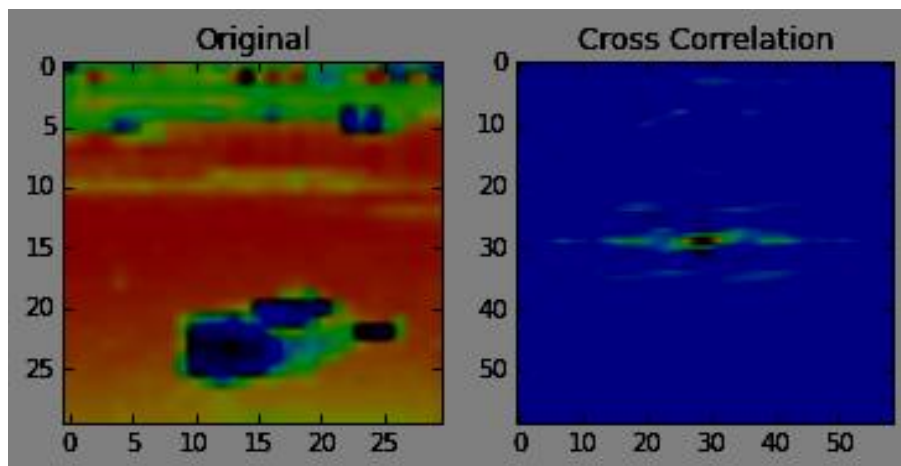
Desplazamiento en la altura 0
Desplazamiento en el ancho 0



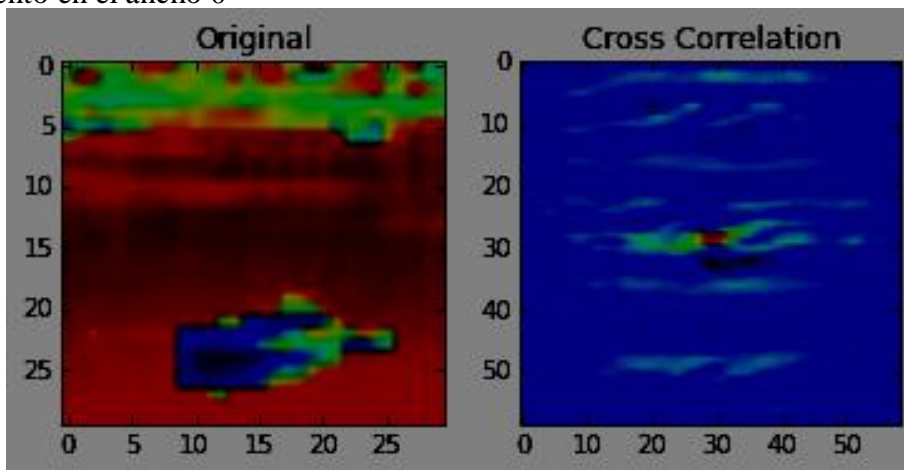
Desplazamiento en la altura 4
Desplazamiento en el ancho -8



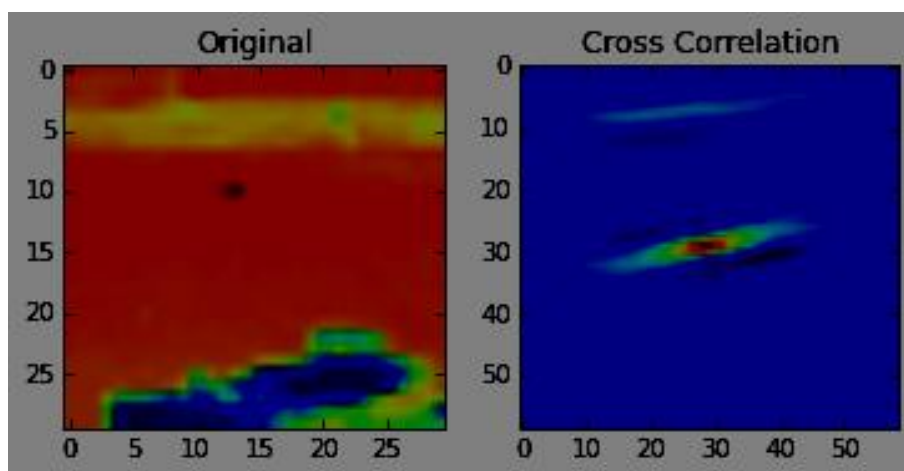
Desplazamiento en la altura 0
Desplazamiento en el ancho 0



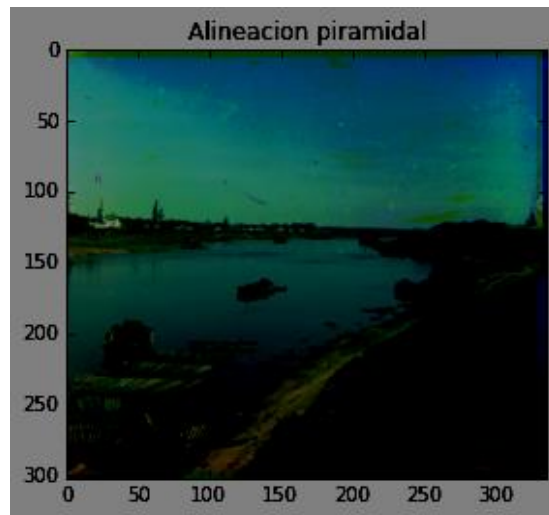
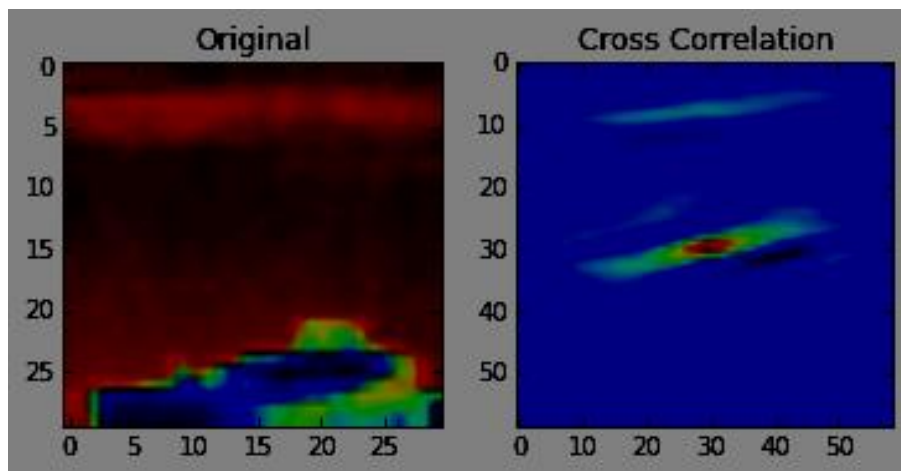
Desplazamiento en la altura -2
Desplazamiento en el ancho 0



Desplazamiento en la altura 0
Desplazamiento en el ancho -1



Desplazamiento en la altura 1
Desplazamiento en el ancho 0



y por lo último procederemos a recortar las filas y columnas no coincidentes de las imágenes finales de cada tipo de alineación:

