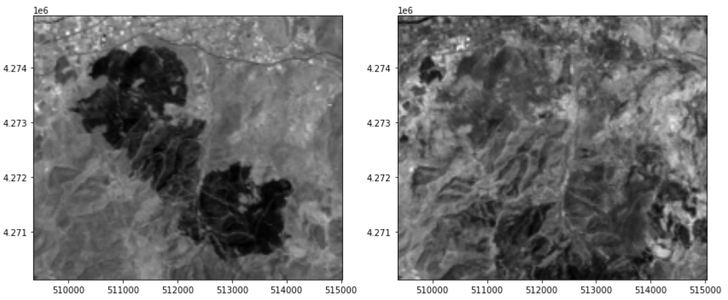
In [1]:
```
#importing required libraries
import rasterio
from rasterio import plot
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

In [2]:
```
#openning the study area images
band4=rasterio.open("Test Images/nir.tif")
band5=rasterio.open("Test Images/swir.tif")
```

In [3]:
```
#multiple band representation
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))
plot.show(band4, ax=ax1, cmap='gray') #nir
plot.show(band5, ax=ax2, cmap='gray') #swir
fig.tight_layout()
```

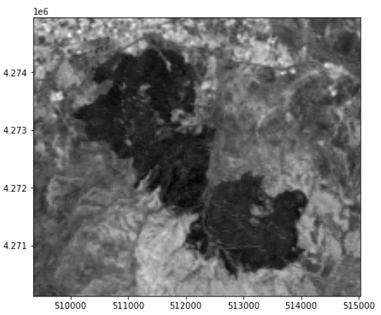


In [4]:
```
#generating nir and swir objects as arrays in float64 format
nir=band4.read(1).astype('float64')
swir=band5.read(1).astype('float64')
```

In [5]:
```
#Normalized Burn Ratio calculation(empty cells or nodata cells are reported as 0)
nbr=np.where(
    (nir+swir)==0.,
    0,
    (nir-swir)/(nir+swir))
```

In [6]:
```
#exporting the NBR image
nbr_image = rasterio.open('Outputs/nbr_image.tiff','w',driver='Gtiff',
                          width=band4.width,
                          height = band4.height,
                          count=1, crs=band4.crs,
                          transform=band4.transform,
                          dtype='float64')
nbr_image.write(nbr,1)
nbr_image.close()
```

In [7]:
```
#plotting the NBR image
nbrimg = rasterio.open('Outputs/nbr_image.tiff')
fig = plt.figure(figsize=(12,6))
plot.show(nbrimg, cmap='gray')
```



Out[7]:
```
<AxesSubplot:>
```

In [8]:
```
#raster sytem of reference
nbrimg.crs
```

Out[8]:
```
CRS.from_epsg(32635)
```

In [9]:
```
#raster transform parameters
nbrimg.transform
```

Out[9]:
```
Affine(30.0, 0.0, 509355.0,
       0.0, -30.0, 4274955.0)
```

In [10]:
```
#type of raster byte
nbrimg.dtypes[0]
```

Out[10]:
```
'float64'
```

In [11]:
```
#number of raster rows
nbrimg.height
```

Out[11]:
```
161
```

In [12]:
```
#number of raster columns
nbrimg.width
```

Out[12]:
```
189
```

In [13]:
```
#importing skimage library in order to show the histogram of NBR image
from skimage import io, exposure
import skimage.io
```

In [14]:
```
#defining a function in order to show the histogram of NBR image
def image_histogram(nbrimg):
    """
    Plot image histogram
    Input:
    img - 2D array of uint16 type
    """
    co, ce = exposure.histogram(nbr)

    fig = plt.figure(figsize=(10, 7))
    fig.set_facecolor('white')
    plt.plot(ce[1::], co[1::])
    plt.show()
```

In [15]:
```
image_histogram(nbrimg)
```