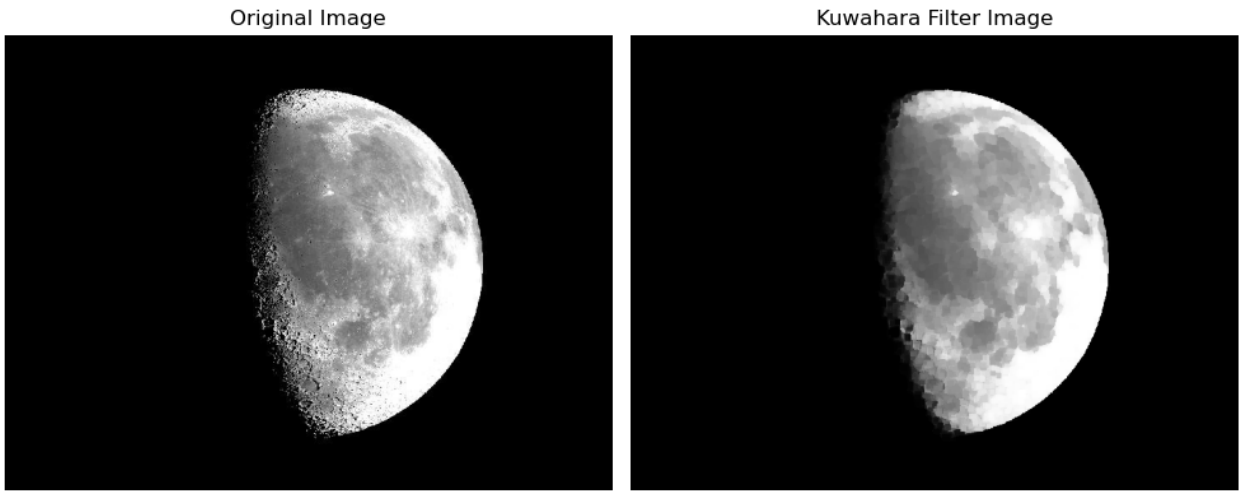
In [11]:

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from scipy.ndimage import uniform_filter
def kuwahara_filter(image, window_size=5):
    assert window_size % 2 == 1, "Window size must be odd."
    offset = window_size // 2
    padded_image = np.pad(image, pad_width=offset, mode='reflect')
    filtered_image = np.zeros_like(image)

    for y in range(image.shape[0]):
        for x in range(image.shape[1]):
            regions = []
            for dy in range(-offset, offset + 1, offset):
                for dx in range(-offset, offset + 1, offset):
                    subregion = padded_image[y + dy:y + dy + window_size,
                    regions.append(subregion)
            variances = [np.var(region) for region in regions]
            means = [np.mean(region) for region in regions]
            filtered_image[y, x] = means[np.argmin(variances)]

    return filtered_image


image_path = '/Users/karedlashilpa/Downloads/Moon_Image.jpg'
img = Image.open(image_path).convert('L')
image_np = np.array(img)

kuwahara_filtered_image = kuwahara_filter(image_np)
fig, axes = plt.subplots(1, 2, figsize=(10, 5))
axes[0].imshow(image_np, cmap='gray')
axes[0].set_title('Original Image')
axes[0].axis('off')

# I am plotting and setting the title below
axes[1].imshow(kuwahara_filtered_image, cmap='gray')
axes[1].set_title('Kuwahara Filter Image')
axes[1].axis('off')
plt.tight_layout()
plt.show()
```

```
/var/folders/2n/2_3thmc94nn7n3qf3vgj6h0w0000gn/T/ipykernel_2345/1395004213
.py:20: RuntimeWarning: invalid value encountered in cast
  filtered_image[y, x] = means[np.argmin(variances)]
```




In [ ]:
```
# Kuwahara filter is like a magic brush that smooths out noisy images wit
#and replacing the center pixel with that square's color. This gives imag

#Effects
#The Kuwahara filter effectively reduces noise while preserving edges and
#It can give images a painterly or cartoonish look, as it tends to smooth

#For Example- When we are painting a landscape. The Kuwahara filter is li
```