Kuwahara Filter

In [4]:

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
```

In [5]:

```
def kuwahara_filter(image, window_size):
    half_window = window_size // 2
    padded_image = np.pad(image, half_window, mode='reflect')
    height, width = image.shape
    output_image = np.zeros_like(image)

    for y in range(half_window, height + half_window):
        for x in range(half_window, width + half_window):
            regions = []
            for i in range(2):
                for j in range(2):
                    region = padded_image[y - i*half_window:y + (1 - i)*half_window
                                          x - j*half_window:x + (1 - j)*half_window
                    regions.append(region)
            variances = [np.var(region) for region in regions]
            min_variance_region = regions[np.argmin(variances)]
            output_image[y - half_window, x - half_window] = np.mean(min_variance_r

    return output_image
```

In [6]:

```
# Load and convert the image to grayscale
image = cv2.imread('img.jpg', cv2.IMREAD_GRAYSCALE)
```

In [7]:

```
kuwahara_result = kuwahara_filter(image, window_size=5)
```

In [8]:

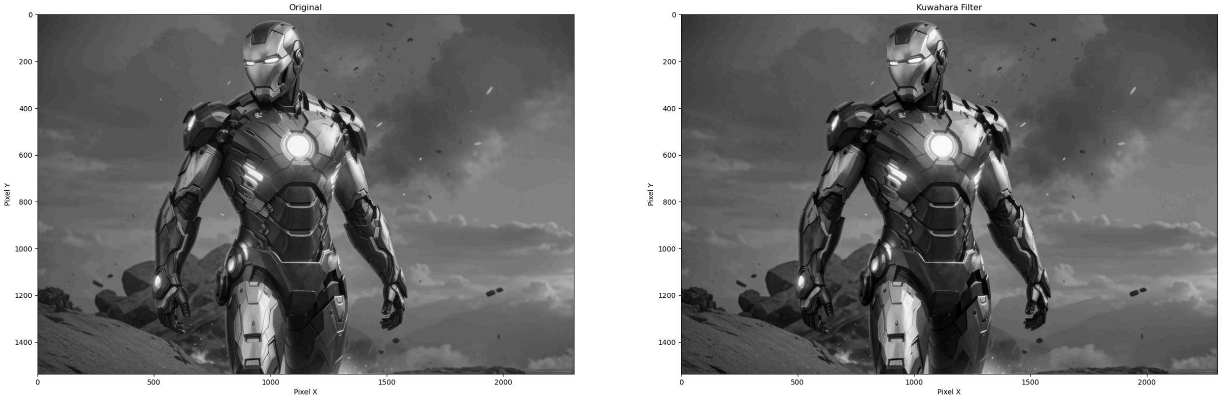
```
plt.figure(figsize=(30,20))

# Original Image
plt.subplot(1,2,1)
plt.imshow(image, cmap='gray')
plt.title('Original')
plt.xlabel('Pixel X')
plt.ylabel('Pixel Y')

# Kuwahara Filtered Image
plt.subplot(1,2,2)
plt.imshow(kuwahara_result, cmap='gray')
plt.title('Kuwahara Filter')
plt.xlabel('Pixel X')
plt.ylabel('Pixel Y')
```

Out[8]:

```
Text(0, 0.5, 'Pixel Y')
```



Original

Kuwahara Filter

In [ ]: