In [1]:

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

Floyd-Steinberg Dithering

In [3]:

```python
def floyd_steinberg_dithering(image):
    height, width = image.shape
    for y in range(height):
        for x in range(width):
            old_pixel = image[y, x]
            new_pixel = np.round(old_pixel / 255.0) * 255
            image[y, x] = new_pixel
            quant_error = old_pixel - new_pixel
            if x + 1 < width:
                image[y, x + 1] += quant_error * 7 / 16
            if x - 1 >= 0 and y + 1 < height:
                image[y + 1, x - 1] += quant_error * 3 / 16
            if y + 1 < height:
                image[y + 1, x] += quant_error * 5 / 16
            if x + 1 < width and y + 1 < height:
                image[y + 1, x + 1] += quant_error * 1 / 16
    return image
```

Jarvis-Judice-Ninke Dithering

In [5]:

```python
def jjn_dithering(image):
    height, width = image.shape
    filter_matrix = np.array([[0, 0, 0, 7, 5],
                              [3, 5, 7, 5, 3],
                              [1, 3, 5, 3, 1]]) / 48.0

    for y in range(height):
        for x in range(width):
            old_pixel = image[y, x]
            new_pixel = np.round(old_pixel / 255.0) * 255
            image[y, x] = new_pixel
            quant_error = old_pixel - new_pixel

            for dy in range(3):
                for dx in range(-2, 3):
                    ny, nx = y + dy, x + dx
                    if 0 <= nx < width and 0 <= ny < height:
                        image[ny, nx] += quant_error * filter_matrix[dy, dx + 2]
    return image
```

In [7]:

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

In [8]:

```python
floyd_steinberg_result = floyd_steinberg_dithering(image.copy())
jjn_result = jjn_dithering(image.copy())
```

In [9]:

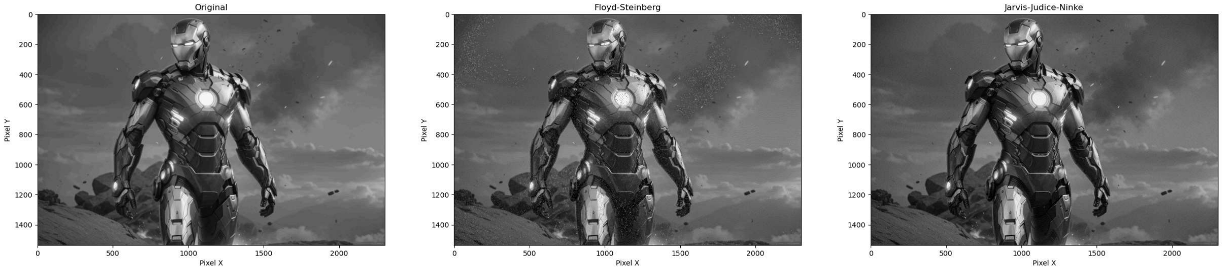
```
plt.figure(figsize=(30,10))
# Original Image
plt.subplot(1,3,1)
plt.imshow(image, cmap='gray')
plt.title('Original')
plt.xlabel('Pixel X')
plt.ylabel('Pixel Y')

# Floyd-Steinberg Dithering Result
plt.subplot(1,3,2)
plt.imshow(floyd_steinberg_result, cmap='gray')
plt.title('Floyd-Steinberg')
plt.xlabel('Pixel X')
plt.ylabel('Pixel Y')

# Jarvis-Judice-Ninke Dithering Result
plt.subplot(1,3,3)
plt.imshow(jjn_result, cmap='gray')
plt.title('Jarvis-Judice-Ninke')
plt.xlabel('Pixel X')
plt.ylabel('Pixel Y')
```

Out[9]:

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



Original

Floyd-Steinberg

Jarvis-Judice-Ninke

In [ ]:

```

```