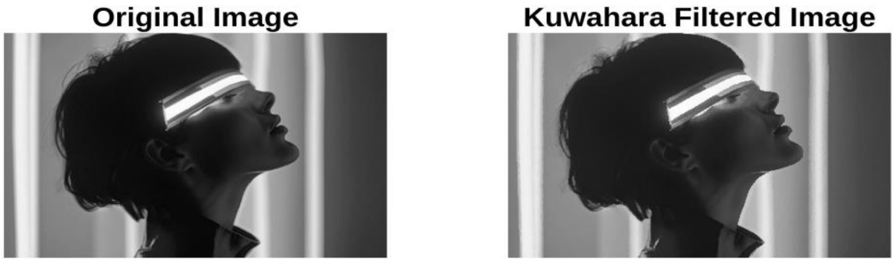
```
function output = kuwahara_filter(img, window_size)
    % Ensure window size is odd
if mod(window_size, 2) == 0
        error('Window size must be odd.');
end

    % Pad the image for edge handling
pad_size = floor(window_size / 2);
    img_padded = padarray(img, [pad_size pad_size], 'symmetric');

    % Initialize the output
    [rows, cols] = size(img);
output = zeros(size(img));

    % Loop through each pixel
    for i = 1:rows
for j = 1:cols
            % Extract the window around the current pixel
            region = img_padded(i:i+window_size-1, j:j+window_size-1);

            % Define the four quadrants
            top_left     = region(1:pad_size+1, 1:pad_size+1);
top_right    = region(1:pad_size+1, pad_size+1:end);
bottom_left  = region(pad_size+1:end, 1:pad_size+1);
bottom_right = region(pad_size+1:end, pad_size+1:end);

            % Calculate mean and variance for each region
            regions = {top_left, top_right, bottom_left, bottom_right};
            means = cellfun(@mean2, regions);
            variances = cellfun(@(x) var(x(:)), regions);

            % Choose the region with the minimum variance
            [~, min_idx] = min(variances);
output(i, j) = means(min_idx);

end
end end

% Load the image and convert to grayscale if necessary
img = imread('DIP_assign_img.jpg'); % Replace with actual file name

if size(img, 3) == 3
img_gray = rgb2gray(img);
else
    img_gray = img;
end

% Apply the Kuwahara filter
window_size = 5;
filtered_img = kuwahara_filter(double(img_gray), window_size);

% Display results
```

```
figure;
subplot(1, 2, 1), imshow(img_gray, []), title('Original Image');
subplot(1, 2, 2), imshow(uint8(filtered_img)), title('Kuwahara Filtered
Image');
```



Original Image

Kuwahara Filtered Image

Kuwahara Filter:

In image processing, the Kuwahara filter is a noise-reducing, edge-preserving filter. By maintaining boundaries and lowering noise in regions that are flat, it soothes the image. Because of this, it is especially helpful for jobs where preserving boundaries is crucial, such as picture segmentation or creative stylization.

The operation of the Kuwahara filter:
Window Division: Four overlapping subregions (quadrants) are created from a square window (often 5x5 or 7x7) that is centered around each pixel.
Calculation of Mean and Variance: The mean and variance are determined for each of the four quadrants.
Pixel Value Assignment: To select the smoothest (least noisy) area, the mean intensity of the quadrant with the lowest variation is used to replace the pixel in the window's center.
In this manner, the Kuwahara filter maintains the image's crisp edges while lowering noise.