Kuwahara filter

It is an edge-preserving noise reduction technique in image processing. First proposed by Kuwahara et al. in 1976, it has become one of the popular methods for smoothing images with a view to preserving important edge information. This filter will be particularly effective in medical imagers, computer vision, and digital art, where edges and structures are of importance.

The basic concept of the Kuwahara filter is to divide a window around each pixel into multiple, usually four quadrants, overlapping subregions. The mean and variance of the pixel intensities in each subregion are calculated by the filter. Subsequently, the output value for the central pixel is set at the mean value of the subregion that displays the lowest variance.

four square subregions centered on the corners of a 5 × 5 pixel neighborhood. Variants differ by the number of subregions, shape, and size.

Kuwahara Filter Implementation:

1. For every pixel in the image:

a. Define four overlapping subregions around the pixel.

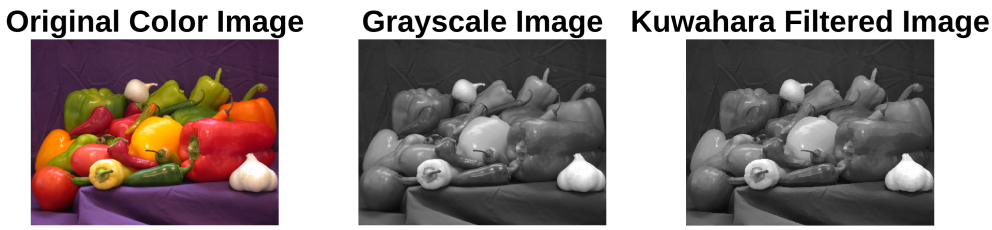
b. Calculate the average and variance of pixel intensities in each subregion.

c. Identify the subregion having minimum variance.

d. Use the average of the chosen subregion as the output value for the central pixel.

```
img = imread('peppers.png');

% Convert to grayscale
img_gray = rgb2gray(img);

% Define the Kuwahara filter function
function output = kuwaharaFilter(input, windowSize)
    % Ensure window size is odd
    if mod(windowSize, 2) == 0
        windowSize = windowSize + 1;
    end

    % Convert input to double
    input = double(input);

    % Pad the input image
    padSize = floor(windowSize / 2);
    paddedInput = padarray(input, [padSize padSize], 'replicate');

    % Initialize output
    output = zeros(size(input));

    % Apply Kuwahara filter
    for i = 1:size(input, 1)
        for j = 1:size(input, 2)
            % Extract window
            window = paddedInput(i:i+windowSize-1, j:j+windowSize-1);

            % Define four sub-regions
            subRegions = {
                window(1:padSize+1, 1:padSize+1),
                window(1:padSize+1, padSize+1:end),
                window(padSize+1:end, 1:padSize+1),
                window(padSize+1:end, padSize+1:end)
            };

            % Calculate mean and variance for each sub-region
            means = cellfun(@(x) mean(x(:)), subRegions);
            variances = cellfun(@(x) var(x(:)), subRegions);

            % Find sub-region with minimum variance
            [~, minIdx] = min(variances);

            % Set output pixel to mean of sub-region with minimum variance
            output(i, j) = means(minIdx);
        end
    end

    % Convert output back to uint8
    output = uint8(output);
```

```
end

% Apply the Kuwahara filter
filtered_img = kuwaharaFilter(img_gray, 5);  % 5 is the window size

% Display the results
figure;
subplot(1,3,1); imshow(img); title('Original Color Image');
subplot(1,3,2); imshow(img_gray); title('Grayscale Image');
subplot(1,3,3); imshow(filtered_img); title('Kuwahara Filtered Image');
```




```
% Optional: Apply to each color channel separately
filtered_img_color = img;
for channel = 1:3
    filtered_img_color(:,:,channel) = kuwaharaFilter(img(:,:,channel), 5);
end

% Display color results
figure;
subplot(1,2,1); imshow(img); title('Original Color Image');
subplot(1,2,2); imshow(filtered_img_color); title('Kuwahara Filtered Color
Image');
```

Original Color Image

Kuwahara Filtered Color Image