```
% Load the image
img = imread('https://cdn.pixabay.com/photo/2015/04/23/22/00/
tree-736885_640.jpg');
img = rgb2gray(img); % Convert to grayscale
img = im2double(img); % Convert to double precision

function filtered_img = kuwahara_filter(img, window_size)
    half_size = floor(window_size / 2);
    [rows, cols] = size(img);
    filtered_img = zeros(rows, cols);

    % Loop through each pixel in the image
    for i = 1:rows
        for j = 1:cols
            % Define boundaries for neighborhoods
            x_start = max(i-half_size, 1);
            x_end = min(i+half_size, rows);
            y_start = max(j-half_size, 1);
            y_end = min(j+half_size, cols);

            % Create neighborhoods
            neighborhoods = {
                img(x_start:i, y_start:j),
                img(x_start:i, j:y_end),
                img(i:x_end, y_start:j),
                img(i:x_end, j:y_end)
            };

            % Initialize minimum variance and corresponding mean
            min_variance = inf;
            best_mean = 0;

            % Iterate through neighborhoods
            for k = 1:length(neighborhoods)
                neighborhood = neighborhoods{k};
                mean_val = mean(neighborhood(:));
                variance_val = var(neighborhood(:));

                % Update the best mean if variance is lower
                if variance_val < min_variance
                    min_variance = variance_val;
                    best_mean = mean_val;
                end
            end

            % Assign the best mean to the filtered image
            filtered_img(i, j) = best_mean;
        end
    end
end
```

```
% Apply the Kuwahara filter with a window size of 5
window_size = 5;
img_filtered = kuwahara_filter(img, window_size);

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

What is the Kuwahara Filter?

The **Kuwahara filter** is a non-linear smoothing filter designed to reduce noise in images while preserving important structural details, especially edges. This makes it particularly valuable in scenarios where maintaining edge clarity is crucial, such as in medical imaging, remote sensing, and artistic effects.

How the Kuwahara Filter Works

1. **Dividing the Neighborhood**:
    - The filter examines a pixel's local neighborhood, typically defined by a square window (for example, 5x5 pixels).
    - This neighborhood is subdivided into four overlapping rectangular regions. The way these regions are divided is typically into quadrants:
        - **Top-left** (Region 1)
        - **Top-right** (Region 2)
        - **Bottom-left** (Region 3)
        - **Bottom-right** (Region 4)
    - The overlapping nature ensures that each pixel in the neighborhood contributes to multiple regions, which helps in smoothing out variations.
2. **Calculating Mean and Variance**:
    - For each of the four regions, the filter computes two key statistics:
        - **Mean**: This is the average value of the pixel intensities in that region. The mean provides a central value around which the pixel intensities in that region are distributed.

        - **Variance**: This measures the spread of pixel values around the mean. A lower variance indicates that the pixel values in that region are more similar (homogeneous), while a higher variance suggests greater variability.
    - The filter calculates these statistics separately for each color channel (e.g., RGB) if working with color images.
3. **Selecting the Region with Lowest Variance**:
    - After calculating the means and variances for all four regions, the filter identifies the region with the lowest variance. The rationale behind this step is that a region with lower variance is more likely to represent a homogeneous area of the image—meaning that it is less affected by noise and has more consistent pixel values.
    - This step helps in effectively distinguishing between textured areas (high variance) and smooth areas (low variance).
4. **Assigning the Mean Value to the Output Pixel**:
    - Once the region with the lowest variance is identified, the filter assigns the mean value of that region to the output pixel in the filtered image.
    - This process is repeated for each pixel in the image, resulting in a new image that has reduced noise while preserving important edge details.

Benefits of the Kuwahara Filter

- **Edge Preservation**: Unlike traditional smoothing filters (e.g., Gaussian blur), the Kuwahara filter effectively maintains edges, making it ideal for applications where detail is crucial.
- **Non-linear Behavior**: The non-linear nature of the filter allows it to adaptively respond to different areas in the image, providing more sophisticated smoothing than linear filters.

Summary

In essence, the Kuwahara filter is a robust technique for noise reduction that focuses on maintaining the structural integrity of an image. By carefully analyzing local neighborhoods and leveraging statistical properties like mean and variance, it provides a smart balance between smoothing and edge preservation, making it a powerful tool in image processing tasks.