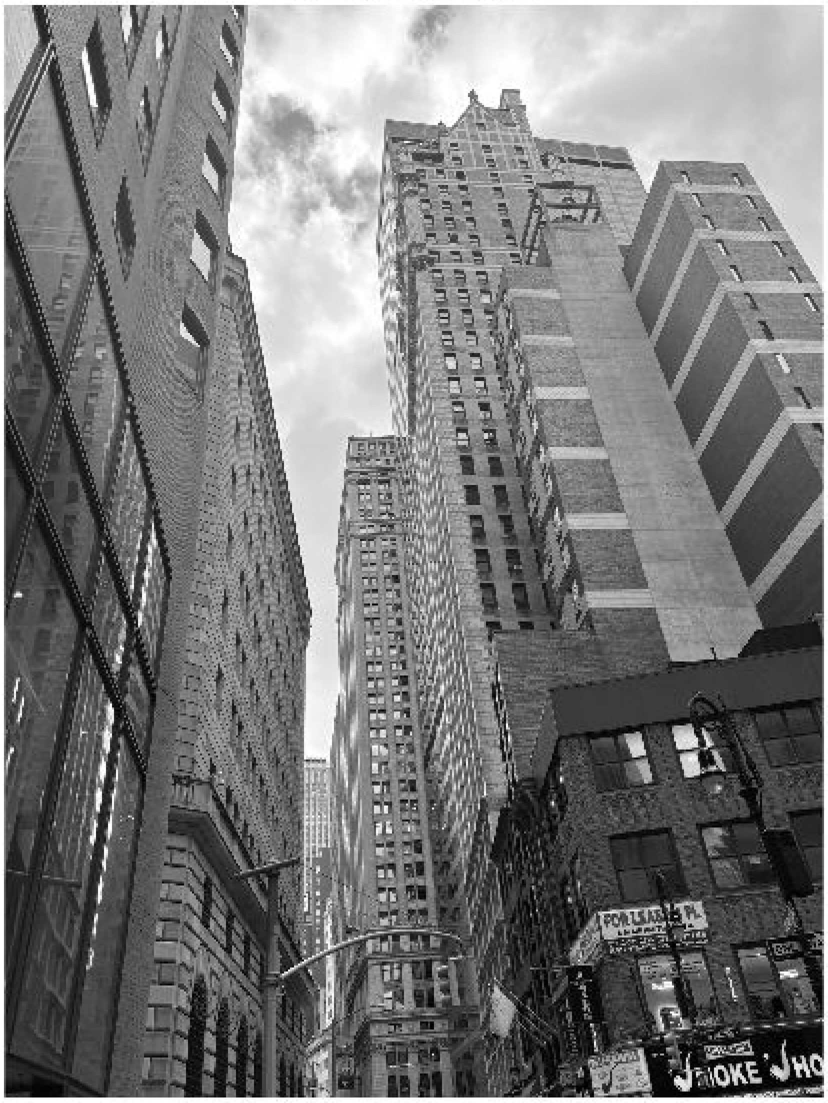
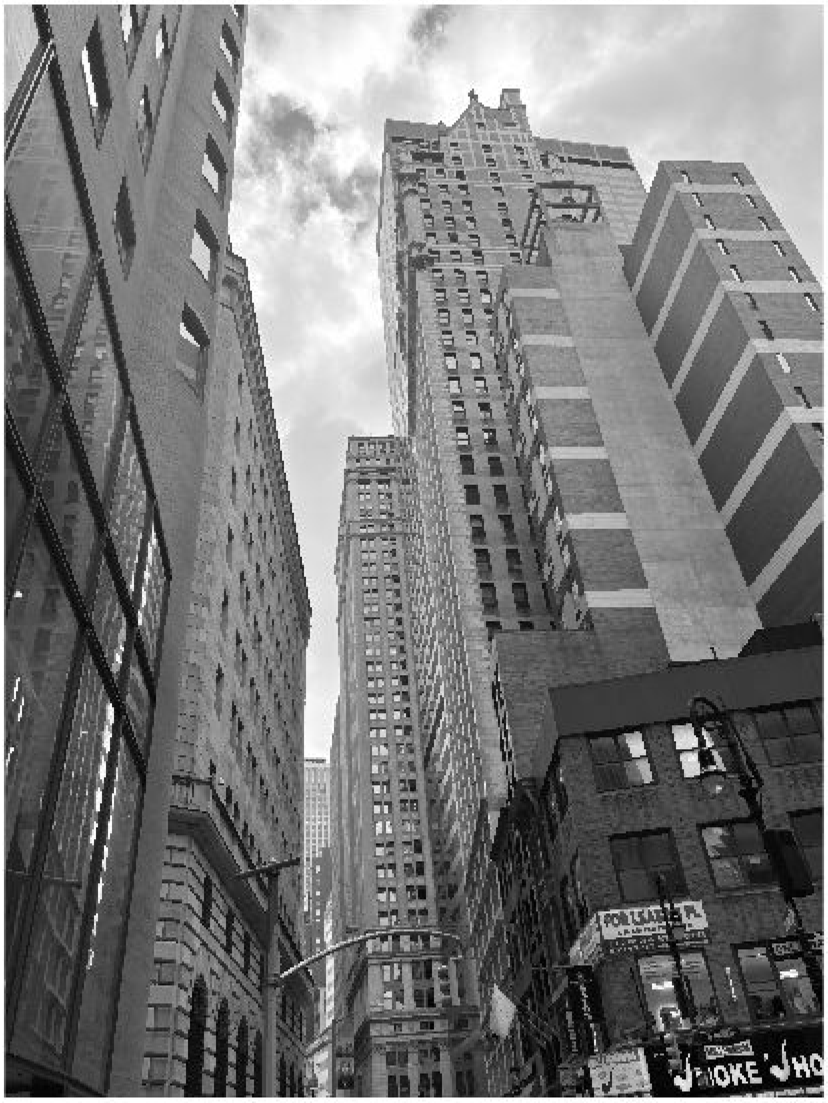
```
clc;


% The Kuwahara filter is a nonlinear smoothing filter used in image
processing that
% preserves edges while also reducing noise. Unlike traditional smoothing
filters
% (such as Gaussian or average filters) that tend to blur both noise and
important details
% like edges, the Kuwahara filter selectively smooths regions
% while maintaining the sharpness of edges.


% Loading and Reading an Image
img = imread('new_york.jpeg');
my_img = rgb2gray(img);  % Convert to grayscale
imshow(my_img);
title('My Original Image');

window_size = 5;  % Set Window size for the Kuwahara filter
filtered_img = kuwahara_filter(my_img, window_size);

% Display the original and filtered images
figure;
imshow(my_img);
title('My Original Image');

figure;
imshow(filtered_img);
title('Image After Applying Kuwahara Filter');

function kuwahara_filtered = kuwahara_filter(img, window_size)

    [rows, cols] = size(img);
    kuwahara_filtered = zeros(rows, cols);
    offset = floor(window_size / 2);  % Offset for subregions
    %Offset defines the boundaries of the quadrants around the center pixel
in the filtering window.

    % Zero-padding the image
    padded_img = padarray(img, [offset, offset], 'symmetric');

    for i = 1:rows
        for j = 1:cols
            % Extract the window around the pixel (i, j)
            window = padded_img(i:i + window_size - 1, j:j + window_size -
1);

            % Define 4 subregions (quadrants)
            quad_a = window(1:offset+1, 1:offset+1);  % Top-left
            quad_b = window(1:offset+1, offset+1:end);  % Top-right
            quad_c = window(offset+1:end, 1:offset+1);  % Bottom-left
            quad_d = window(offset+1:end, offset+1:end);  % Bottom-right

            % Calculate mean and variance of each quadrant
            means = [mean(quad_a(:)), mean(quad_b(:)), mean(quad_c(:)),
mean(quad_d(:))];
            variances = [var(double(quad_a(:))), var(double(quad_b(:))),
var(double(quad_c(:))), var(double(quad_d(:)))];

            % Find the subregion with the lowest variance
            [y, k] = min(variances);

            % Assign the center pixel the mean of the subregion with the
lowest variance
            kuwahara_filtered(i, j) = means(k);
        end
    end

    % Convert the result back to the uint8 format
    kuwahara_filtered = uint8(kuwahara_filtered);
end
```

My Original Image



My Original Image

Image After Applying Kuwahara Filter



Published with MATLAB® R2023b