~/Downloads/lab16.m

```
% Read the input image
img = imread('rgb.jpeg'); % Replace 'your_image.jpg' with the actual image file
name

% Convert the image to grayscale if it is RGB
if size(img, 3) == 3
    img = rgb2gray(img);
end

% Resize image to a square if not already
[N, M] = size(img);
if N ~= M
    N = min(N, M); % Use the smaller dimension for resizing
    img = imresize(img, [N, N]);
end

% Display the original image
figure;
subplot(1,2,1);
imshow(img);
title('Original Image');

% Convert image to double for computations
img = double(img);

% Define parameters
angles = 0:1:179; % Angles in degrees (0° to 179°)
num_angles = length(angles);
projections = zeros(num_angles, N); % Initialize projection matrix
center = ceil(N / 2); % Center of the image

% Loop through each angle
for a = 1:num_angles
    theta = angles(a); % Current angle in degrees
    radon_sum = zeros(1, N); % Projection at this angle

    % For each column in the projection
    for col = 1:N
        x_rot = (1:N) - center; % X-coordinates centered at the origin
        y_rot = col - center;   % Y-coordinate for the current projection line

        % Rotate coordinates (inverse rotation to align with projection axis)
        x_orig = x_rot * cosd(theta) - y_rot * sind(theta) + center;
        y_orig = x_rot * sind(theta) + y_rot * cosd(theta) + center;

        % Interpolate values from the image
        for idx = 1:N
            x = x_orig(idx);
            y = y_orig(idx);
```

```
            % Check if indices are within the image bounds
            if x >= 1 && x <= N && y >= 1 && y <= N
                % Nearest neighbor interpolation
                x_nn = round(x);
                y_nn = round(y);
                radon_sum(col) = radon_sum(col) + img(x_nn, y_nn);
            end
        end
    end

    % Store the projection for the current angle
    projections(a, :) = radon_sum;
end

% Display the Radon transform
subplot(1,2,2);
imshow(projections', [], 'XData', angles, 'YData', 1:N, 'InitialMagnification',
'fit');
xlabel('Angle (degrees)');
ylabel('Projection Position');
title('Radon Transform');
colormap(gca, hot);
colorbar;

```