```
image = rgb2gray(imread('pisa.jpg'));

figure(1), clf, hold on
subplot(1,2,1)
imshow(image);

degree = 5.5;

switch mod(degree, 360)
    % Special cases
    case 0
        imagerot = image;
    case 90
        imagerot = rot90(image);
    case 180
        imagerot = image(end:-1:1, end:-1:1);
    case 270
        imagerot = rot90(image(end:-1:1, end:-1:1));

    % General rotations
    otherwise

        % Convert to radians and create transformation matrix
        a = degree*pi/180;
        R = [+cos(a) +sin(a); -sin(a) +cos(a)];

        % Figure out the size of the transformed image
        [m,n,p] = size(image);
        dest = round( [1 1; 1 n; m 1; m n]*R );
        dest = bsxfun(@minus, dest, min(dest)) + 1;
        imagerot = zeros([max(dest) p],class(image));

        % Map all pixels of the transformed image to the original image
        for ii = 1:size(imagerot,1)
            for jj = 1:size(imagerot,2)
                source = ([ii jj]-dest(1,:))*R.';
                if all(source >= 1) && all(source <= [m n])

                    % Get all 4 surrounding pixels
                    C = ceil(source);
                    F = floor(source);

                    % Compute the relative areas
                    A = [...
                        ((C(2)-source(2))*(C(1)-source(1))),...
                        ((source(2)-F(2))*(source(1)-F(1)));
                        ((C(2)-source(2))*(source(1)-F(1))),...
                        ((source(2)-F(2))*(C(1)-source(1)))];

                    % Extract colors and re-scale them relative to area
                    cols = bsxfun(@times, A, double(image(F(1):C(1),F(2):C(2),:)))
```

```
                    % Assign
                    imagerot(ii,jj,:) = sum(sum(cols),2);

                end
            end
        end
end

subplot(1,2,2)
imshow(imagerot);
```

Conclusion

Job done

Published with MATLAB® R2014a