```
function [corrX, corrY] = getCorners(image1, image2, rThreshold,
 nccThreshold)
    verticalPrewittFilter = [-1 0 1;
                             -1 0 1;
                             -1 0 1];

    horizontalPrewittFilter = [-1 -1 -1;
                               0 0 0;
                               1 1 1];

    % Harris Corner Detection
    verticalPrewitt1 = imfilter(image1,
 verticalPrewittFilter, 'replicate');
    horizontalPrewitt1 = imfilter(image1,
 horizontalPrewittFilter, 'replicate');

    verticalPrewitt2 = imfilter(image2,
 verticalPrewittFilter, 'replicate');
    horizontalPrewitt2 = imfilter(image2,
 horizontalPrewittFilter, 'replicate');

    R1 = CMatrix(verticalPrewitt1, horizontalPrewitt1);
    R2 = CMatrix(verticalPrewitt2, horizontalPrewitt2);

    magnitude1 = sqrt(horizontalPrewitt1.^2 + verticalPrewitt1.^2);
    orientation1 = atand(horizontalPrewitt1./verticalPrewitt1);
    magnitude2 = sqrt(horizontalPrewitt2.^2 + verticalPrewitt2.^2);
    orientation2 = atand(horizontalPrewitt2./verticalPrewitt2);

    R1(R1 < rThreshold) = 0;
    R1(R1 >= rThreshold) = 255;
    orientation1(isnan(orientation1)) = 90; % get rid of bad
 orientation values

    R2(R2 < rThreshold) = 0;
    R2(R2 >= rThreshold) = 255;
    orientation2(isnan(orientation2)) = 90; % get rid of bad
 orientation values

    % Non-Max Suppression
    nonMaxSuppress1 = nonMaxSuppression(orientation1, magnitude1, R1);
    nonMaxSuppress2 = nonMaxSuppression(orientation2, magnitude2, R2);

    % NCC
    [corrX, corrY] = normalizedCrossCorrelation(image1, image2,
 nonMaxSuppress1, nonMaxSuppress2, nccThreshold);
end
```

Published with MATLAB® R2017a