```
%==========================
%  Name:                hw3_2.m
%
%  Author:              Kairi Kozuma
%
%==========================

% Transformation matrix
R_WL = [0.913545457642601,-0.063627629171822,0.401729040058774;0.287606238475951,0.799453749866612,-0.527405302792764;-0.287606238475951,0.597348496800
R_WR = [0.994521895368273,-0.016351854232753,0.103241544429788;0.073912785203567,0.808411029059454,-0.583959337863936;-0.073912785203567,0.588391217607
T_WL = [-8.659258262890683;2.169872981077807;4.830127018922193];
T_WR = [10.659258262890683;5.830127018922193;1.169872981077807];

% Homogenous points in world frame
q_W = [16,78.3000000000000,33.1000000000000;-25.4000000000000,-20.9000000000000,-39.1000000000000;19.1000000000000,7.30000000000000,38.5000000000000;1

% a) Transformation giving camera's R frame relative to L frame

    G_WR = [R_WR,T_WR;0,0,0,1];
    G_WL = [R_WL,T_WL;0,0,0,1];
    G_LW = [R_WL',-R_WL'*T_WL;0,0,0,1];
    G_LR = G_LW * G_WR;

    fprintf('a) Transformation matrix of R frame relative to L frame:\n');
    disp(G_LR);

% b) Coordinates of points given in both frames

    % Convert to camera R frame
    qcR1 = transformToCamera(q_W(1:3,:), R_WR', -R_WR'*T_WR);
    fprintf('b1) Points in camera R frame:\n');
    disp (qcR1(1:3,:));

    % Convert to camera R frame
    qcL1 = transformToCamera(q_W(1:3,:), R_WL', -R_WL'*T_WL);
    fprintf('b2) Points in camera L frame:\n');
    disp (qcL1(1:3,:));

% c) Both cameras have horizontal FOV of 60deg, vertical FOV of 40deg
%    Specify if each point is visible by L only, R only, or both

    % Field of view +- the following value
    horiFOV = 30;
    vertFOV = 20;

    % Determine if in field of view
    inViewR = inFOV(qcR1(1:3,:),horiFOV, vertFOV);
    inViewL = inFOV(qcL1(1:3,:),horiFOV, vertFOV);

    inView = inViewR & inViewL;

    fprintf('b) Points in both fields of view:\n');
    count = 0;
    for n = 1:length(inView)
        if (inView(n))
            fprintf('Point q%d in field of view\n', n);
            count = count + 1;
        end
    end

    if (count == 0)
        fprintf('\tNo points in field of view\n\n');
    end

%======================== transformToCamera ==============================
%
%  qc = transformToCamera(pw, R_CW, T_CW)
%
%  INPUTS:
%    pw         - point in 3 dimension, world frame
%    R_CW       - rotation matrix
%    T_CW       - translation vector
%
%  OUTPUTS:
%    qc             - point in 3 dimensions, camera frame
%
%======================== transformToCamera ==============================
function [pc] = transformToCamera(pw, R_CW, T_CW)

transformMatrix = [R_CW,T_CW;0,0,0,1];

dim = size(pw);
lastRow = ones([1,dim(2)]);

qw = [pw; lastRow];

pc = transformMatrix * qw;

end
```

```
%================================== inFOV ================================
%
%  inView = inFOV(pc, horiFOV, vertFOV)
%
%  INPUTS:
%    pc          - point in 3 dimensions, camera frame
%    horiFOV     - horizontal field of view, +- value
%    vertFOV     - vertical field of view, +- value
%
%  OUTPUTS:
%    inView              - boolean vector of whether points are in FOV
%
%================================== inFOV ================================
function [inView] = inFOV(pc, horiFOV, vertFOV)

angleY = (180 / pi) * atan2(pc(2,:), pc(3,:));
angleX = (180 / pi) * atan2(pc(1,:), pc(3,:));

angles = [angleX; angleY];

inHoriView = (angles(1,:) >= -horiFOV & angles(1,:) <= horiFOV);
inVertView = (angles(2,:) >= -vertFOV & angles(2,:) <= vertFOV);

inView = inHoriView & inVertView;

end
```

```
a) Transformation matrix of R frame relative to L frame:
    0.9511    0.0483   -0.3052   19.7538
   -0.0483    0.9988    0.0076   -0.4894
    0.3052    0.0076    0.9523    3.0902
         0         0         0    1.0000

b1) Points in camera R frame:
    1.6779   64.8414   16.2377
  -14.7842  -19.1081  -14.7242
   33.2257   27.5286   58.6121

b2) Points in camera L frame:
   10.4940   72.0958   16.5958
  -15.0858  -22.5009  -15.5377
   35.1298   48.9503   63.7484

b) Points in both fields of view:
Point q3 in field of view
```

Published with MATLAB® R2016b