```
%==========================
%  Name:                hw5_1.m
%
%  Author:              Kairi Kozuma
%
%==========================

% Points in the world
qWpts = [2,8,3,4,6,5,5;7,6,2,8,8,5,4;5,4,4,4,2,2,2;1,1,1,1,1,1,1];

% Left camera points
rpL = [384,189,22,358,313,215,165;137,169,287,172,243,298,316;1,1,1,1,1,1,1];

% Right camera points
rpR = [579,433,253,597,580,457,408;178,250,375,220,309,379,399;1,1,1,1,1,1,1];

% psi matrix
psi = [500,0,320;0,-500,240;0,0,1];
psi = [psi,zeros(3,1)];

% 1) R, T pairs for each camera, with respect to the world frame
disp('1) R, T pairs for each camera, with respect to the world frame');
[R_LW, T_LW] = extrinsicCalib(rpL, qWpts, psi);
[R_RW, T_RW] = extrinsicCalib(rpR, qWpts, psi);

disp('Left camera R with respect to world frame:');
scaleL = nthroot(det(R_LW),3);
R_LW = R_LW ./ scaleL;
disp(R_LW);

disp('Left camera T with respect to world frame:');
T_LW = T_LW ./ scaleL;
disp(T_LW);

disp('Right camera R with respect to world frame:');
scaleR = nthroot(det(R_RW),3);
R_RW = R_RW ./ scaleR;
disp(R_RW);

disp('Right camera T with respect to world frame:');
T_RW = T_RW ./ scaleR;
disp(T_RW);

% Check if transformation matrix is correct
G_LW = [R_LW, T_LW; 0,0,0,1];
G_RW = [R_RW, T_RW; 0,0,0,1];

% rpL2 = psi * G_LW * qWpts;
% rpL2 = round(rpL2 ./ rpL2(3,:));
% disp(rpL);
% disp(rpL2);
% rpR2 = psi * G_RW * qWpts;
% rpR2 = round(rpR2 ./ rpR2(3,:));
% disp(rpR);
% disp(rpR2);

% 2) R, T pair for the right camera relative to the left camera
disp('2) R, T pair for the right camera relative to the left camera');
```

```matlab
% Obtain G_LR
G_LR = G_LW * inv(G_RW);

% Extract R and T
R_LR = G_LR(1:3,1:3);
T_LR = G_LR(1:3,end);

disp('Right camera R with respect to left camera frame');
disp(R_LR);

disp('Right camera T with respect to left camera frame');
disp(T_LR);

%============================== extrinsicCalib ============================
%
%
%  Given a set of image points plus the world coordinates that they came
%  from, and thirdly the intrinsic camera matrix, solve for the extrinsic
%  parameters associated to the camera rig.
%
%
%  function [R_CW, T_CW] = extrinsicCalib(rp, qWpts, psi)
%
%  Inputs:
%    rp      - Points in the image as homogeneous rays.
%    qWpts   - Points in the world in homogeneous form.
%    psi     - psi matrix
%
%  Outputs:
%    R_CW       - The rotation of the camera frame relative to the world frame.
%    T_CW       - The translation of the camera frame relative to the world frame.
%
%============================== extrinsicCalib ============================
function [R_CW, T_CW] = extrinsicCalib(rp, qWpts, psi)

% Convert image pts (in pixels) to rays (in world length units)
sz = size(qWpts);
masterMatrix = zeros(2*sz(2),12);

for index = 1:sz(2)
    rmat = makeRMat(rp(:,index));
    Qmat = makeQMat(qWpts(:,index));
    mat = rmat * Qmat;
    index2 = 2*index;
    masterMatrix(index2:index2 + 1,:) = mat(1:2,:);
end

% % Run for each column in rp and qWpts to construct 12 x 12 matrix.
% for i = 1:size(rp,2)
% end

% Perform SVD and reconstruct extrinsic parameters (scale by nth root of det)

[UU SS VV] = svd(masterMatrix);
szvv = size(VV);
M = VV(:,szvv(2));
M = (reshape(M, 4, 3))';

G_CW = inv(psi(1:3,1:3)) * M;

R_CW = G_CW(1:3,1:3);
```

```
T_CW = G_CW(1:3,end);

end


%=============================== makeRMat ==============================
%
%  function mat = makeRMat(vector)
%
%
%  INPUT:
%    vector - 3 x 1 vector
%
%=============================== makeRMat ==============================
function mat = makeRMat(vector)
mat = [0, vector(3), -vector(2); -vector(3), 0, vector(1); vector(2), -vector(1), 0];
end


%=============================== makeQMat ==============================
%
%  function mat = makeQMat(vector)
%
%
%  INPUT:
%    vector - 4 x 1 vector
%
%=============================== makeQMat ==============================
function mat = makeQMat(qPts)
mat = [qPts',zeros(1,8); zeros(1,4), qPts', zeros(1,4);zeros(1,8), qPts'];
end
```

```
1) R, T pairs for each camera, with respect to the world frame
Left camera R with respect to world frame:
   -0.5632    0.8209   -0.1026
    0.2305    0.2799    0.9365
    0.7827    0.5062   -0.3482

Left camera T with respect to world frame:
   -3.1524
   -5.5655
    4.0902

Right camera R with respect to world frame:
    0.0069    0.9341   -0.3888
    0.0470    0.3825    0.9309
    0.9795   -0.0147   -0.0546

Right camera T with respect to world frame:
   -1.3297
   -6.6396
    4.7487

2) R, T pair for the right camera relative to the left camera
Right camera R with respect to left camera frame
    0.7932    0.1864   -0.5895
   -0.0973    0.9765    0.1892
    0.5879   -0.0817    0.7989
```

```
Right camera T with respect to left camera frame
    1.9397
   -0.1099
    0.5357
```

Published with MATLAB® R2016b