```
squareSize = 37.3;  % in units of 'millimeters'
num_samples = 20;
num_images = 20;

stereo_params_array = {};
for i = 0 : num_samples - 1
    imageFileNames1 = {'','','','','','','','','','',...
        '','','','','','','','','',''};
    imageFileNames2 = {'','','','','','','','','','',...
        '','','','','','','','','',''};
    disp(i)
    for j = 0 : num_images - 1
        path1 = "C:\Users\Evgen\Documents\My\Projects\stereocamera\images_samples\" ↙
+ i + "\left\" + j + ".jpg";
        imageFileNames1{j+1} = char(path1);

        path2 = "C:\Users\Evgen\Documents\My\Projects\stereocamera\images_samples\" ↙
+ i + "\right\" + j + ".jpg";
        imageFileNames2{j+1} = char(path2);
    end

    % Detect checkerboards in images
    [imagePoints, boardSize, imagesUsed] = detectCheckerboardPoints ↙
(imageFileNames1, imageFileNames2);

    % Generate world coordinates of the checkerboard keypoints
    worldPoints = generateCheckerboardPoints(boardSize, squareSize);

    % Read one of the images from the first stereo pair
    I1 = imread(imageFileNames1{1});
    [mrows, ncols, ~] = size(I1);

    % Calibrate the camera
    [stereoParams, pairsUsed, estimationErrors] = estimateCameraParameters ↙
(imagePoints, worldPoints, ...
        'EstimateSkew', true, 'EstimateTangentialDistortion', true, ...
        'NumRadialDistortionCoefficients', 3, 'WorldUnits', 'millimeters', ...
        'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', [], ...
        'ImageSize', [mrows, ncols]);

    stereo_params_array{i+1} = stereoParams;

end

fx_L = zeros(1,20);
fy_L = zeros(1,20);
cx_L = zeros(1,20);
cy_L = zeros(1,20);

k1_L = zeros(1,20);
k2_L = zeros(1,20);
k3_L = zeros(1,20);
```

```
p1_L = zeros(1,20);
p2_L = zeros(1,20);


fx_R = zeros(1,20);
fy_R = zeros(1,20);
cx_R = zeros(1,20);
cy_R = zeros(1,20);

k1_R = zeros(1,20);
k2_R = zeros(1,20);
k3_R = zeros(1,20);
p1_R = zeros(1,20);
p2_R = zeros(1,20);

tx = zeros(1,20);
ty = zeros(1,20);
tz = zeros(1,20);

r11 = zeros(1,20); r12 = zeros(1,20); r13 = zeros(1,20);
r21 = zeros(1,20); r22 = zeros(1,20); r23 = zeros(1,20);
r31 = zeros(1,20); r32 = zeros(1,20); r33 = zeros(1,20);

for i = 1 : num_samples

    fx_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.FocalLength(1);
    fy_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.FocalLength(2);
    cx_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.PrincipalPoint↙
(1);
    cy_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.PrincipalPoint↙
(2);

    k1_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.RadialDistortion↙
(1);
    k2_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.RadialDistortion↙
(2);
    k3_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.RadialDistortion↙
(3);
    p1_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.↙
TangentialDistortion(1);
    p2_L(i) = stereo_params_array{i}.CameraParameters1.Intrinsics.↙
TangentialDistortion(2);

    fx_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.FocalLength(1);
    fy_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.FocalLength(2);
    cx_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.PrincipalPoint↙
(1);
    cy_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.PrincipalPoint↙
(2);

    k1_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.RadialDistortion↙
(1);
    k2_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.RadialDistortion↙
```

```
(2);
    k3_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.RadialDistortion↙
(3);
    p1_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.↙
TangentialDistortion(1);
    p2_R(i) = stereo_params_array{i}.CameraParameters2.Intrinsics.↙
TangentialDistortion(2);

    tx(i) = stereo_params_array{i}.TranslationOfCamera2(1);
    ty(i) = stereo_params_array{i}.TranslationOfCamera2(2);
    tz(i) = stereo_params_array{i}.TranslationOfCamera2(3);

    r11(i) = stereo_params_array{i}.RotationOfCamera2(1,1);
    r12(i) = stereo_params_array{i}.RotationOfCamera2(1,2);
    r13(i) = stereo_params_array{i}.RotationOfCamera2(1,3);

    r21(i) = stereo_params_array{i}.RotationOfCamera2(2,1);
    r22(i) = stereo_params_array{i}.RotationOfCamera2(2,2);
    r23(i) = stereo_params_array{i}.RotationOfCamera2(2,3);

    r31(i) = stereo_params_array{i}.RotationOfCamera2(3,1);
    r32(i) = stereo_params_array{i}.RotationOfCamera2(3,2);
    r33(i) = stereo_params_array{i}.RotationOfCamera2(3,3);
end


% average_fx_L = averageMinMax(fx_L, 274.5, 276)
% average_fx_R = averageMinMax(fx_R, 275, 276.5)
%
% average_fy_L = averageMinMax(fy_L, 274.5, 275.6)
% average_fy_R = averageMinMax(fy_R, 274.9, 276.5)
%
% average_cx_L = averageMinMax(cx_L, 0, 157)
% average_cx_R = averageMinMax(cx_R, 0, 169.3)
%
% average_cy_L = averageMinMax(cy_L, 252.3, 254)
% average_cy_R = averageMinMax(cy_R, 258, 259.3)
%
% average_k1_L = averageMinMax(k1_L, -0.435, -0.425)
% average_k2_L = averageMinMax(k2_L, 0.21, 0.24)
% average_k3_L = averageMinMax(k3_L, -1, -0.05)
%
% average_k1_R = averageMinMax(k1_R, -1, -0.4268)
% average_k2_R = averageMinMax(k2_R, 0.21, 1)
% average_k3_R = averageMinMax(k3_R, -1, -0.005)
%
%
% average_p1_L = averageMinMax(p1_L, -1, 0)
% average_p2_L = averageMinMax(p2_L, 0, 1)
%
% average_p1_R = averageMinMax(p1_R, -1, 0)
% average_p2_R = averageMinMax(p2_R, -1*10^(-3), 1)
average_fx_L = averageMinMax(fx_L, -1000, 1000)
```

```
average_fx_R = averageMinMax(fx_R, -1000, 1000)

average_fy_L = averageMinMax(fy_L, -1000, 1000)
average_fy_R = averageMinMax(fy_R, -1000, 1000)

average_cx_L = averageMinMax(cx_L, -1000, 1000)
average_cx_R = averageMinMax(cx_R, -1000, 1000)

average_cy_L = averageMinMax(cy_L, -1000, 1000)
average_cy_R = averageMinMax(cy_R, -1000, 1000)

average_k1_L = averageMinMax(k1_L, -1000, 1000)
average_k2_L = averageMinMax(k2_L, -1000, 1000)
average_k3_L = averageMinMax(k3_L, -1000, 1000)

average_k1_R = averageMinMax(k1_R, -1000, 1000)
average_k2_R = averageMinMax(k2_R, -1000, 1000)
average_k3_R = averageMinMax(k3_R, -1000, 1000)


average_p1_L = averageMinMax(p1_L, -1000, 1000)
average_p2_L = averageMinMax(p2_L, -1000, 1000)

average_p1_R = averageMinMax(p1_R, -1000, 1000)
average_p2_R = averageMinMax(p2_R, -1000, 1000)

figure();
subplot(2,1,1);
plot(1:num_samples, fx_L, "red", 1:num_samples, fx_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_fx_L, "--red", 1:num_samples, ones↙
(num_samples)*average_fx_R, "--blue");
grid('on');
title('Фокусное расстояние по направлению x');
ylabel('f_x [пиксели]');
xlabel('Набор №');
legend('левая камера', 'правая камера');

subplot(2,1,2);
plot(1:num_samples, fy_L, "red", 1:num_samples, fy_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_fy_L, "--red", 1:num_samples, ones↙
(num_samples)*average_fy_R, "--blue");
grid('on');
title('Фокусное расстояние по направлению y');
ylabel('f_y [пиксели]');
xlabel('Набор №');
legend('левая камера', 'правая камера');

figure();
subplot(2,1,1);
plot(1:num_samples, cx_L, "red", 1:num_samples, cx_R, "blue");
hold on;
```

```
plot(1:num_samples, ones(num_samples)*average_cx_L, "--red", 1:num_samples, ones(num_samples)*average_cx_R, "--blue");
grid('on');
title('x координата главной точки (c_x)');
ylabel('c_x [пиксели]');
xlabel('Набор №');
legend('левая камера', 'правая камера');

subplot(2,1,2);
plot(1:num_samples, cy_L, "red", 1:num_samples, cy_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_cy_L, "--red", 1:num_samples, ones(num_samples)*average_cy_R, "--blue");
grid('on');
title('y координата главной точки (c_y)');
ylabel('c_y [пиксели]');
xlabel('Набор №');
legend('левая камера', 'правая камера');


figure();
subplot(3,1,1);
plot(1:num_samples, k1_L, "red", 1:num_samples, k1_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_k1_L, "--red", 1:num_samples, ones(num_samples)*average_k1_R, "--blue");
grid('on');
title('Коэффициент радиальной дисторсии k_1');
ylabel('k_1');
xlabel('Набор №');
legend('левая камера', 'правая камера');

subplot(3,1,2);
plot(1:num_samples, k2_L, "red", 1:num_samples, k2_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_k2_L, "--red", 1:num_samples, ones(num_samples)*average_k2_R, "--blue");
grid('on');
title('Коэффициент радиальной дисторсии k_2');
ylabel('k_2');
xlabel('Набор №');
legend('левая камера', 'правая камера');

subplot(3,1,3);
plot(1:num_samples, k3_L, "red", 1:num_samples, k3_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_k3_L, "--red", 1:num_samples, ones(num_samples)*average_k3_R, "--blue");
grid('on');
title('Коэффициент радиальной дисторсии k_3');
ylabel('k_3');
xlabel('Набор №');
legend('левая камера', 'правая камера');
```

```
figure();
subplot(2,1,1);
plot(1:num_samples, p1_L, "red", 1:num_samples, p1_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_p1_L, "--red", 1:num_samples, ones
(num_samples)*average_p1_R, "--blue");
grid('on');
title('Коэффициент тангенциальной дисторсии p_1');
ylabel('p_1');
xlabel('Набор №');
legend('левая камера', 'правая камера');

subplot(2,1,2);
plot(1:num_samples, p2_L, "red", 1:num_samples, p2_R, "blue");
hold on;
plot(1:num_samples, ones(num_samples)*average_p2_L, "--red", 1:num_samples, ones
(num_samples)*average_p2_R, "--blue");
grid('on');
title('Коэффициент тангенциальной дисторсии p_2');
ylabel('p_2');
xlabel('Набор №');
legend('левая камера', 'правая камера');

figure();
subplot(2,1,1);
plot(1:num_samples, tx, 1:num_samples, ty, 'green', 1:num_samples, tz, 'red');
grid('on');
title('Элементы вектора переноса от правой камеры к левой');
ylabel('Миллиметры');
xlabel('Набор №');
legend('t_x', 't_y', 't_z');

subplot(2,1,2);
plot(1:num_samples, r11, 1:num_samples, r12, 1:num_samples, r13,...
    1:num_samples, r21, 1:num_samples, r22, 1:num_samples, r23,...
    1:num_samples, r31, 1:num_samples, r32, 1:num_samples, r33);
grid('on');
title('Элементы матрицы поворота от правой камеры к левой');
xlabel('Набор №');
legend('r11', 'r12', 'r13',...
    'r21', 'r22', 'r23',...
    'r31', 'r32', 'r33');

average_tx = averageMinMax(tx, -1000, 1000)
average_ty = averageMinMax(ty, -1000, 1000)
average_tz = averageMinMax(tz, -1000, 1000)

average_r11 = averageMinMax(r11, -1000, 1000)
average_r12 = averageMinMax(r12, -1000, 1000)
average_r13 = averageMinMax(r13, -1000, 1000)
average_r21 = averageMinMax(r21, -1000, 1000)
```

```
average_r22 = averageMinMax(r22, -1000, 1000)
average_r23 = averageMinMax(r23, -1000, 1000)
average_r31 = averageMinMax(r31, -1000, 1000)
average_r32 = averageMinMax(r32, -1000, 1000)
average_r33 = averageMinMax(r33, -1000, 1000)
```