```
% Calculate the stereoParams and other camera instrinsics
% This was run in the Stereo Camera Calibrator App and saved
%  to the workspace.  It can be re-run via the script if needed.
% run("stereoParam_script.m")

% Read in images
i1_o = imread('StereoImages\8.jpg');
i2_o = imread('StereoImages\9.jpg');

% Copy images to allow for manipulation
i1 = i1_o;
i2 = i2_o;

% Check for grayscale and convert
if (size(i1,3)>1)
    i1 = rgb2gray(i1); % ensure grayscale
end

if (size(i2,3)>1)
    i2 = rgb2gray(i2); % ensure grayscale
end


% Use SURF to detect features in both images
points1 = detectSURFFeatures(i1);
[features1, points1] = extractFeatures(i1, points1);

points2 = detectSURFFeatures(i2);
[features2, points2] = extractFeatures(i2, points2);


% Match the points across the image pair
indexPairs = matchFeatures(features1, features2);
% I had origionally tried to use the exhaustive methods and maxRatio
%  Using them provided me with an empty matrix for indexPairs
%  resulting in no triangulation.  Removing them allowd for points
%  to be matched.
%, 'Method','Exhaustive','MaxRatio',0.02)

% Assign the matched points
matchedPoints1 = points1(indexPairs(:,1), :);
matchedPoints2 = points2(indexPairs(:,2), :);



% Drop in RANSAC here to remove outliers
[f1MedS, inliers] = estimateFundamentalMatrix(matchedPoints1, ...
    matchedPoints2, Method="RANSAC", NumTrials=2000);

% Set the inliers for matched points 1 and 2
matchedPoints1_inliers = matchedPoints1(inliers,:);
```

```
matchedPoints2_inliers = matchedPoints2(inliers,:);


% Look at the matched inlier points and run triangulation
for i=1:height(matchedPoints1_inliers)

    center1 = round(matchedPoints1_inliers.Location(i,:));
    center2 = round(matchedPoints2_inliers.Location(i,:));

    % define some area around the feature point (the calc'd center)
    area1 = [center1(1)-10, center1(2)-10, 20, 20];
    area2 = [center2(1)-10, center2(2)-10, 20, 20];

    % compute distance from camera to center point
    point3d = triangulate(center1, center2, stereoParams);
    distance = norm(point3d); % I'm not sure what units this is in yet...

    % Display the detected point and distance
    distanceString = sprintf('%0.1f units', distance);
    i1 = insertObjectAnnotation(i1, 'rectangle', area1, distanceString, ...
        'FontSize',60, Color='black', TextColor='white');
    i2 = insertObjectAnnotation(i2, 'rectangle', area2, distanceString, ...
        'FontSize',60, Color='black', TextColor='white');
end

% output results
imshowpair(i1,i2, 'montage')
```