Homework 3

Guixiang Zhang

Describe:

1. Method 1 Using Linear Least-Square

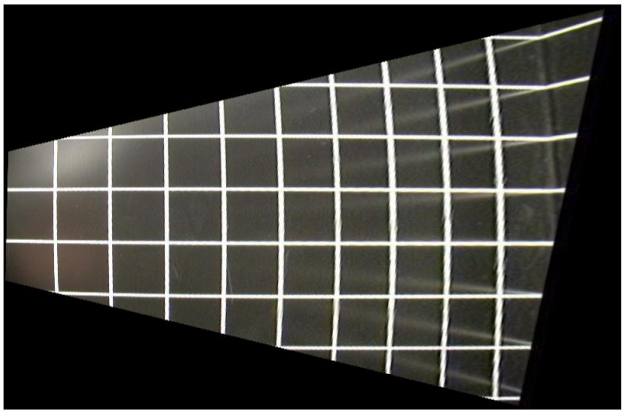
MATLAB code : Zhang_homework3LSmethod.m

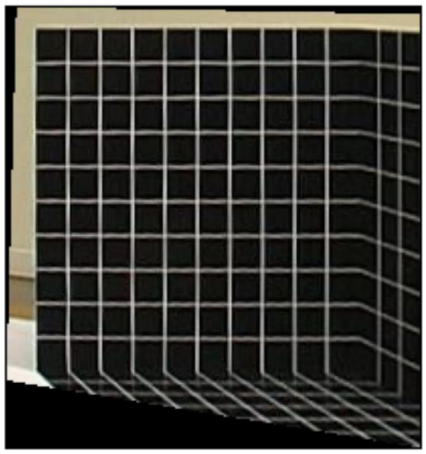
In this method, take the "Board, jpg" as example, I select 4 points in it. Every three of them are not collinear. Then I think where they should be in the front view picture. So I get another 4 points coordinate of the front view picture. Now I get 4 pair of points, I get the points correspondences. Now I can do LS and find the 9 parameters of the projection matrix H. they are h11,h12 ⋯, h33. They fit the constraint that the 2-norm of H vector equals 1.

Then I simply set h33=1 because H has 8 DOF. And Use the function "imwarp" to compute the new coordinates and get new image, the front view.

NOTE: we should get the transpose of H to be the "tform" because of the computation rule of MATLAB. The area that the 4 points determine should be large enough to represent the plane you want to show its the front view.

Results :

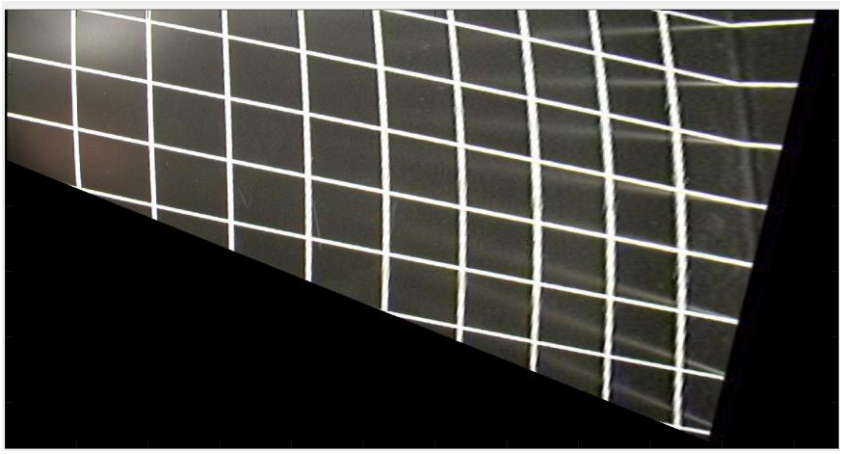




2. Method 2 Remove Perspective and Affine Properties

MATLAB code : Zhang_homework3Rectification.m

This is a method mentioned in Computer Vision class. First I select two pair of lines in the perspective piciture which they represent two pairs of parallel lines in the real world. Then I can get the line at infinity 'l'. If we want to remove the perspective properties 'v', we should get a transformation H map l to (0,0,1) which is the canonical coordinates of the line at infinity. If l=(l1,l2,l3) l3≠0. The transformation H should be [1,0,0 ; 0,1,0; l1,l2,l3]. Using H, we can remove the perspective properties and get an affinely rectified image. For example.



The parallel lines are parallel and line at infinity is (0,0,1).

Then, from the affinely rectified image, we select 2 pair of orthogonal lines, and they should not parallel. We suppose the translation is 0. l1=(x1,x2,x3) m1=(y1,y2,y3) are a pair of orthogonal lines. We have,

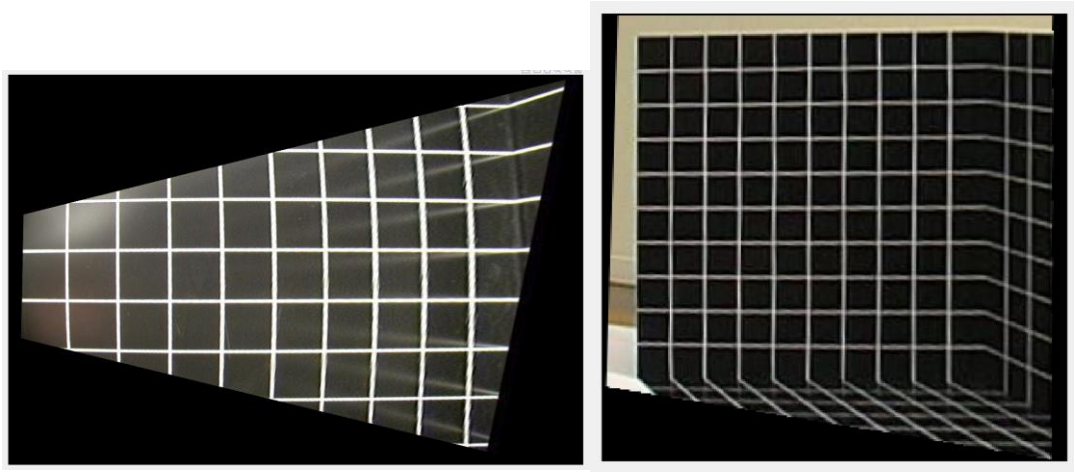
$$(x1y1, x1y2 + x2y1, x2y2)(S1, S2, S3)^T = 0$$

where, $S = \begin{bmatrix} S1 & S2 \\ S3 & S3 \end{bmatrix}$ is up to scale, so it only have 2 DOF. And $S = KK^T, K = \begin{bmatrix} k11 & k12 \\ 0 & k22 \end{bmatrix}$ is the upper-triangle matrix of the affine transformation $H_A = \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix}$. Then we can use the two pair of lines to get 2 equations to get $k11, k12, k22$.

Then we use the inverse of H_A to rectify the affine rectified image. This method is up to similarity transformation. So the result is the real front view after using similarity transformation. Thus I may need to rotate it for conveniently see.

NOTE: The lines we select should surround the area you focus to get its front view. In this project, I want them to be sparse enough because I want to get the front of whole picture.

Here are the results.



3. Compare two method:

In method 2, we don't need points correspondences. There are some error in both method, I think it may because I always choose the intersection points of the white lines, but they have widths.