CSCE 643 Multi-View Geometry CV Homework I

I. Four point rectification

As we know, the pictures taken by a camera is actually projections of Euclidean prototype of real-world scenarios. Assuming we have a point in the Euclidean space whose homogenized coordinate is (x, y, z), and its counterpart in projective space, or in the picture, is (x', y', z). According to the definition of projective transformation, the planar projective transformation can be represented by a non-singular 3×3 matrix:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{1}$$

more briefly as $\mathbf{x}' = \mathbf{H}\mathbf{x}$, this can be further transformed into (note that by default we assume $h_{33} = 1$):

$$\begin{aligned} x' &= h_{11}x + h_{12}y + h_{13}z \\ y' &= h_{21}x + h_{22}y + h_{23}z \\ z' &= h_{31}x + h_{32}y + z \end{aligned} \tag{2}$$

However, this is up scale, and if we go down scale and set $z' = 1, z = 1$, it turns out that it can be transformed through equation 3–5:

$$\begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1} \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1} \end{aligned} \tag{3}$$

$$\begin{aligned} x'(h_{31}x + h_{32}y + 1) &= h_{11}x + h_{12}y + h_{13} \\ y'(h_{31}x + h_{32}y + 1) &= h_{21}x + h_{22}y + h_{23} \end{aligned} \tag{4}$$

$$\begin{aligned} x' &= h_{11}x + h_{12}y + h_{13} - h_{31}x'x - h_{32}xy \\ y' &= h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' \end{aligned} \tag{5}$$

which could also be represented by matrix forms:

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -yx' & -x' \\ 0 & 0 & 0 & x & y & 1 & -xy' & -yy' & -y' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0 \tag{6}$$

Now, say we have four points whose coordinate in Euclidean space is $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, and their corresponding coordinates in projective space is $(x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3), (x'_4, y'_4)$, let:

$$\begin{aligned} p_i &= \begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_ix'_i & -y_ix'_i & -x'_i \end{pmatrix} \\ p'_i &= \begin{pmatrix} 0 & 0 & 0 & x_i & y_i & 1 & -x_iy'_i & -y_iy'_i & -y'_i \end{pmatrix} \end{aligned} \tag{7}$$

we can then easily scale equation 6 for our current point set:

$$\begin{bmatrix} p_1 \\ p'_1 \\ p_2 \\ p'_2 \\ p_3 \\ p'_3 \\ p_4 \\ p'_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ 1 \end{bmatrix} = 0 \tag{8}$$

Then, we solve the above simultaneous linear equations thereby getting the homograpy $\mathbf{H}$ and apply the homography on the original picture to rectify it, hereby the question 1 is solved and a review of solution steps is listed as follows:

1) Pick up 4 apex points of a rectangle in the picture and get their coordinates.
2) Acquire the coordinates of those points picked in the actual Euclidean space (as we just want to rectify the image here we just arbitrarily select 4 coordinates that can form a rectangle and ignores things about scales and actual location).
3) Do reverse homography to get the homography matrix using those coordinates we get.
4) Apply the homography we get to all points in the picture space and get the rectified image.

To validate our solution, we did several experiments using our four point rectification approach and get the following results as shown in Figure 1, for every result we attached the original image with selected points highlighted by its side for convenience of comparison.

II. Affine rectification using parallelism

The key of using parallel lines in projective space to recover affine properties from images is the infinite line. In the affinity space, the infinite line is a fixed line $l_\infty = (0, 0, 1)^T$, however a projective transformation might maps l_∞ from the fixed line at infinity to a finite line l on the space after projection. Then, say we have the infinite line $l = (l_1, l_2, l_3)^T$ in a projective space, where $l_3 \neq 0$, and the homography of this current projection $\mathbf{H}$ can be divided as:

$$\mathbf{H} = \mathbf{H}_A\mathbf{H}_P \tag{9}$$

(a) Result of findHomography in OpenCV

(b) Result of Our Four Point Rectification Approach

(c) The Point Choice in Original Picture (Apexes of Highlighted Green Rectangle)

Fig. 1. Four Point Rectification vs. findHomography in OpenCV

where $\mathbf{H}_A$ is the affine homography and the last matrix $\mathbf{H}_P$ is the homography for transformation from affine space to current projective space:

$$\mathbf{H}_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \tag{10}$$

That is to say, the current projective transformation can be decomposed into two parts, one is the transformation to affine space and the other one is the transformation from affinity to current projective space, and the later one can be directly calculated if the infinite line is given.

Now that we figured out the infinite line could help us back to affinity, we can start to work on the details to calculate the infinite line. We know that in Euclidean space, two parallel line will intersect at an ideal point on infinite line, and if we can get two ideal points then we can easily calculate the infinite line as two points determine a line. Intuitively, we can identify two pairs of parallel lines from the distorted picture and calculate two ideal points through them to form the infinite line and then we can get back to affinity based on our discussion above.

Suppose we have four points p_1, p_2, p_3, p_4 and they form a rectangle similar to what we have in question 1, through those we can simply get two pairs of parallel lines:

$$\begin{aligned} \vec{l}_1 &= p_1 \times p_2 \\ \vec{l}_2 &= p_3 \times p_4 \\ \vec{m}_1 &= p_1 \times p_3 \\ \vec{m}_2 &= p_2 \times p_4 \end{aligned} \tag{11}$$

in which we have $l_1 /\!/ l_2$ and $m_1 /\!/ m_2$. Through those pairs of parallel lines, we can further compute two points at the infinite line as follows:

$$\begin{aligned} v_1 &= \vec{l}_1 \times \vec{l}_2 \\ v_2 &= \vec{m}_1 \times \vec{m}_2 \end{aligned} \tag{12}$$

And finally we can acquire the line at infinity $\vec{l}_\infty = (l_1, l_2, l_3)$ which can be calculated by:

$$\vec{l}_\infty = v_1 \times v_2 \tag{13}$$

According to our discussion above, now we can form a new matrix same as given in equation 10 based on the infinite line, and the new $\mathbf{H}_P$ can handle the transformation between the picture space and affinity.

To summarize, the following steps are needed when we are doing the affine rectification:

1) Find two physically parallel line pairs in the picture plane. (here we can simply use those points we picked for the rectangle in our first question, since those four apexes naturally provide two parallel line pairs)
2) Through those parallel lines we got, solve two intercepts of those parallel lines in picture plane, which are ideal points that should lie on the line of infinity in the world plane.
3) From the two ideal points we can now get line of infinity in the picture plane.
4) Since we know the coordinates of infinite line in the world plane, we can now solve the homography transformation from picture plane to affinity.

Both result we got from rectifying from the wall and the floor is show in Figure 2, from the wall rectification part we can see that parallelism is recovered. However, I still do not know how to transform Figure 2(b) to show the parallelism recovery of rectification result through the floor plane, sorry about that.

III. From Affinity to normal: a two step approach

Now that we have got the image $\mathbf{I}_A$ that is affinely rectified (1st step), we can further remove the affine distortion from it (step 2) through using C^*_∞, basically to find the $\mathbf{H}_A$ in equation 9 so that:

$$\mathbf{H}_A = \begin{bmatrix} A & \vec{t} \\ \vec{0} & 1 \end{bmatrix} \tag{14}$$

and if we apply $\mathbf{H}_A$ on the affinely rectified image we can get the real-world image:

$$\mathbf{I}_{real} = \mathbf{H}_A \mathbf{I}_A \tag{15}$$

To solve $\mathbf{H}_A$, suppose we have two orthogonal lines on the world plane $\vec{l} \perp \vec{m}$, where $\vec{l} = (l_1, l_2, l_3), \vec{m} = (m_1, m_2, m_3)$, and $\vec{l'} = (l'_1, l'_2, l'_3), \vec{m'} = (m'_1, m'_2, m'_3)$ are the two projected lines in the affine plane. If we do a dehomogenization for $\vec{l}, \vec{m}$, we have:

$$\begin{aligned} \vec{l} &= (l_1/l_3, l_2/l_3) \\ \vec{m} &= (m_1/m_3, m_2/m_3) \end{aligned} \tag{16}$$

After which we can use the orthogonality so that we have:

$$\begin{aligned} (l_1/l_3, l_2/l_3)(m_1/m_3, m_2/m_3)^T = 0 \\ \Leftrightarrow l_1 m_1 + l_2 m_2 = 0 \end{aligned} \tag{17}$$

Now we introduce the dual degenerate conic:

$$C^*_\infty = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{18}$$

So that we have:

$$l_1 m_1 + l_2 m_2 = \vec{l}^T C^*_\infty \vec{m} = 0 \tag{19}$$

Since we know the fact that:

$$\begin{aligned} \vec{l}^T &= \vec{l'}^T H_A \\ \vec{m} &= H_A^T \vec{m'} \end{aligned} \tag{20}$$

we can transform equation 19 to:

$$\begin{aligned} \vec{l}^T C^*_\infty \vec{m} &= \vec{l'}^T H_A C^*_\infty H_A^T \vec{m'} \\ &= \vec{l'}^T \begin{pmatrix} A & \vec{t} \\ \vec{0} & 1 \end{pmatrix} \begin{pmatrix} \vec{I} & \vec{0} \\ \vec{0} & \vec{1} \end{pmatrix} \begin{pmatrix} A^T & \vec{0} \\ \vec{t}^T & 1 \end{pmatrix} \vec{m'} \\ &= \vec{l'}^T \begin{pmatrix} AA^T & \vec{0} \\ \vec{0} & 0 \end{pmatrix} \vec{m'} \end{aligned} \tag{21}$$

If we plug $\vec{l'} = (l'_1, l'_2, l'_3), \vec{m'} = (m'_1, m'_2, m'_3)$ into equation 21 we will have:

$$(l'_1, l'_2) AA^T (m'_1, m'_2)^T = 0 \tag{22}$$

Since AA^T is definitely a symmetric matrix, we can assume $S = AA^T = \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & 1 \end{pmatrix}$.

After which we have:

$$(l'_1, l'_2) \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & 0 \end{pmatrix} (m'_1, m'_2)^T = 0 \tag{23}$$

Which is a simultaneous linear equations with 2 DoFs, thus we can solve s_{11}, s_{12} if we have two pairs of orthogonal lines in real-world plane.

So basically what we need to do in this question is:

1) Find two pairs of orthogonal lines in real-world plane.
2) Transform those orthogonal lines pairs from picture space to affined space.
3) Build simultaneous linear equations of form in equation 23 and solve it to get S.
4) Use Singular Value Decomposition APIs in OpenCV to solve A from S and then construct the homography from affined space to world space.

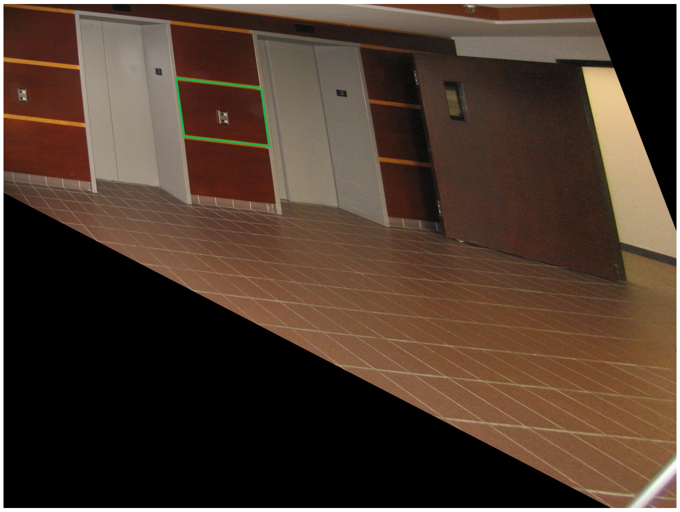
To validate the correctness of our approach, we did experiment on the picture 1 and rectified it leveraging the squares on the floor plane, we can see from the result Figure 3(a) that the squares on the floor is correctly recovered.

IV. Important Results

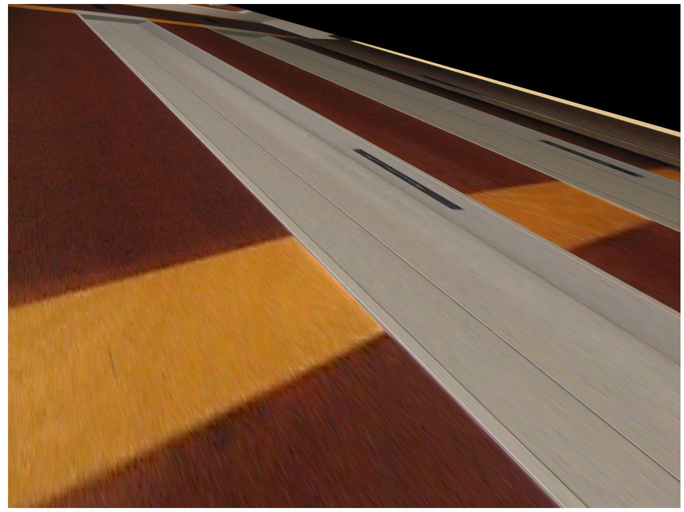
A. Problem I

In problem I, we have solved the homography using four apexes of the rectangle and get:

$$\mathbf{H}_1 = \begin{pmatrix} 0.253003 & -0.0815367 & 554.457 \\ -0.121283 & 0.502774 & 236.837 \\ -0.000246957 & -5.84335e-05 & 1 \end{pmatrix} \tag{24}$$



(a) Result of Affine Rectification from the Wall



(b) Result of Affine Rectification from Squares on the Floor

(c) The Point Choice in Original Picture (Apexes of Highlighted Green Rectangle)

Fig. 2. Affinely Rectification through Wall vs. Floor

B. Problem II

In problem II, the first and most important result we get is the vanishing line (line at infinity) at the picture plane, I solved it to be:

$$\vec{l_\infty} = \begin{pmatrix} -8.46e+11 & -2.00e+11 & 3.42e+15 \end{pmatrix} \tag{25}$$

And if we dehomogenize the line of infinity so that we can actully apply it into homography, we can get:

After solving the line of infinity, we can get the homography from picture plane to affinity:

$$\mathbf{H}_A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.000740871 & -0.0001753 & 3 \end{pmatrix} \tag{26}$$

C. Problem III

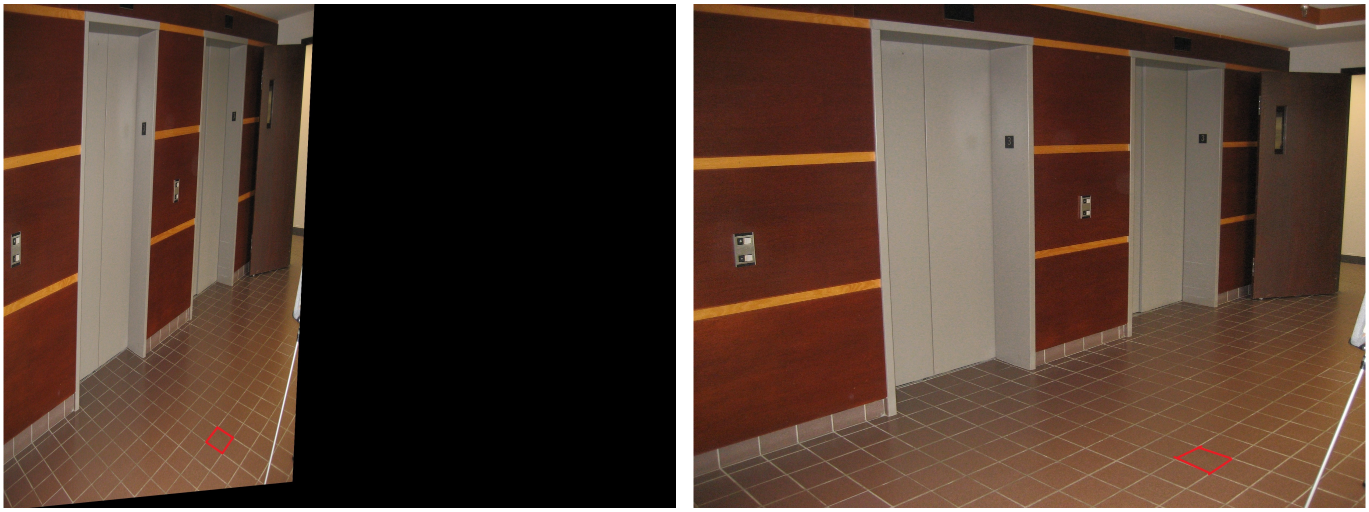
For problem III we can just transform the orthogonal lines back to affinity using $\mathbf{H}_A$ and then get the homography (from affinity to world plane) by using coordinates of orthogonal lines, the result of homography we get is:

$$\mathbf{H}_2 = \begin{pmatrix} 0.953997 & -0.00396995 & 0 \\ -0.00396995 & 0.999992 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{27}$$

Due to time and space limitations, we ignored some irrelevant and simple intermediate results like the coordinates of parallel or orthogonal lines.

V. Conclusion & Discussions

As we have researched and implemented, all the methods above can successfully transform a distorted image from the picture plane back to the world plane (The unimplemented approach in our challenge also does the job). For me I think those approaches are actually using similar quantity of informations:



(a) Two-Step Rectification through Floor Square

(b) The Point Choice in Original Picture (Apexes of Highlighted Green Rectangle)

Fig. 3. Two-Step Rectification through Floor Plane

- Problem 1: we used four known points that form a rectangle.
- Problem 2: we used two pairs of parallel line to rectify the image back to affinity.
- Problem 3: based on the same points (lines) we used in 2, we transform the picture from affinity to world plane.

Be noticed that problem 2 and problem 3 are actually combined together for us to get the image back from picture plane to world plane, and in the whole process the points we used are exactly the same with problem 1. Therefore, in reality though we have different approaches to rectify images but as long as the start plane and the target plane we want in the rectification is the same, we will need the same amount of information from the world planes (e.g., same amount of points, same amount of lines, etc).

And if we analyze the difference between the two step approch implemented in this paper and the challenge part (one-step approach), both of them can remove projective and affine distortions while preserving the similarity. The two-step approach should be more robust since in every step we only need to solve a 2D simultaneous linear function while the one-step approach is numerically unstable due to the need of solving 5 DoFs equations, which is also one of the reason I didn't succeed in it (it seems to rely a lot on the orthogonal line pairs we are choosing). Further questions may be resolved by email and the source code of all the questions is pushed to the Github repository: https://github.com/vincenttsang/multiview-geo-cv.