CS6790 : Assignment 1

Sumanth R Hegde, EE17B032

February 26, 2020

1 Aim

The aim of the assignment is to rectify various in-the-wild images by performing a perspective transform in four different methods.

2 Homography estimation using four point correspondences

For a general perspective transform, the homography matrix has 8 unknown parameters to be estimated. Thus, given four point correspondences

$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \qquad i = 1, 2, 3, 4$$

One would get 2 equations per point. This would yield 8 equations in the 9 variables of H. Let

$$\overrightarrow{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]$$

be the flattened version of matrix H. Then, by stacking all the co-efficients into a matrix A, we desire the solution for

$$A_{8\times 9}\overrightarrow{h} = \overrightarrow{0} \tag{2}$$

From the Singular Value Decomposition (SVD) of A, we obtain,

$$\begin{gathered} A = U_{8\times 8} D_{8\times 9} V^T_{9\times 9} \\ \Rightarrow UDV^T\overrightarrow{h} = \overrightarrow{0} \\ \Rightarrow V^T\overrightarrow{h} = \overrightarrow{0} \\ \text{For} \quad \text{V} = [\overrightarrow{v_1}, \overrightarrow{v_2}.....\overrightarrow{v_9}] \\ \text{We get} \quad ¡ \overrightarrow{v_i}, \overrightarrow{h} >= 0 \qquad i = 1, 2, ...8 \\ \Rightarrow \overrightarrow{h} = \overrightarrow{v_9} \qquad \text{(Upto scale)} \end{gathered}$$

Where $< ., . >$ is the dot product.

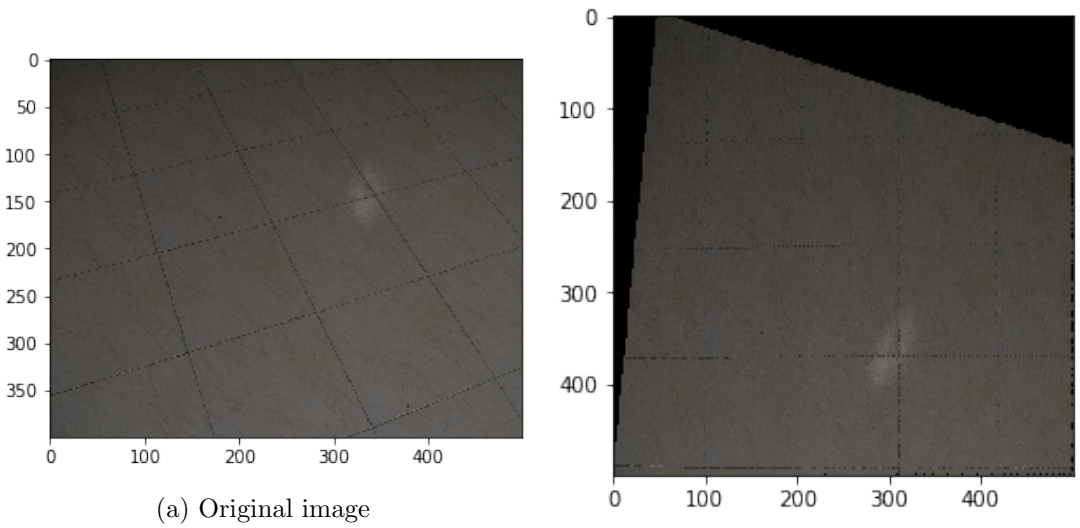
2.1 Implementation

In python, the `Opencv` library provides many convenient functionalities for computer vision-related tasks. For a better understanding of the rectification process, a custom implementation was first developed with the following procedure :

- Use the steps mentioned above to obtain H from four point correspondences
- For a grid of pixel coordinates of the original image, the transformed coordinates are obtained.

- The resulting array would have correspondences only in certain regions, with values at many points being unknown. To account for these "holes", interpolation is performed. For a small pixel area (3x3), interpolating using `max` function gave more visually pleasing results.
- Appropriate shifting is performed for display.

Following result was obtained on `Image1.JPG` using the custom implementation: For obtaining the four point correspondences, corners of a square (or a rectangle) were chosen. The length of the sides were measured to get an approximate idea about the scale of the original image. Based on this, four points were constructed such that the camera axis was perpendicular to the plane of the scene of interest.

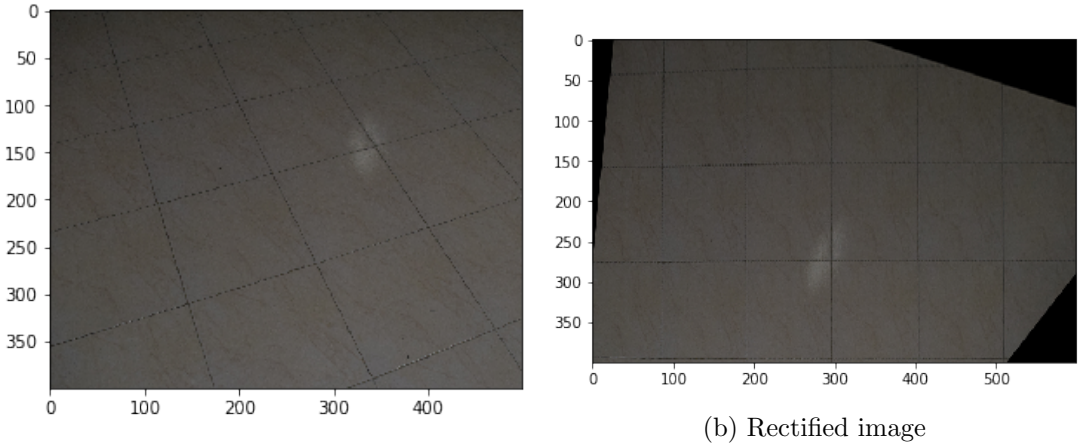


(a) Original image

(b) Rectified image

Figure 1: Image rectification with a custom implementation

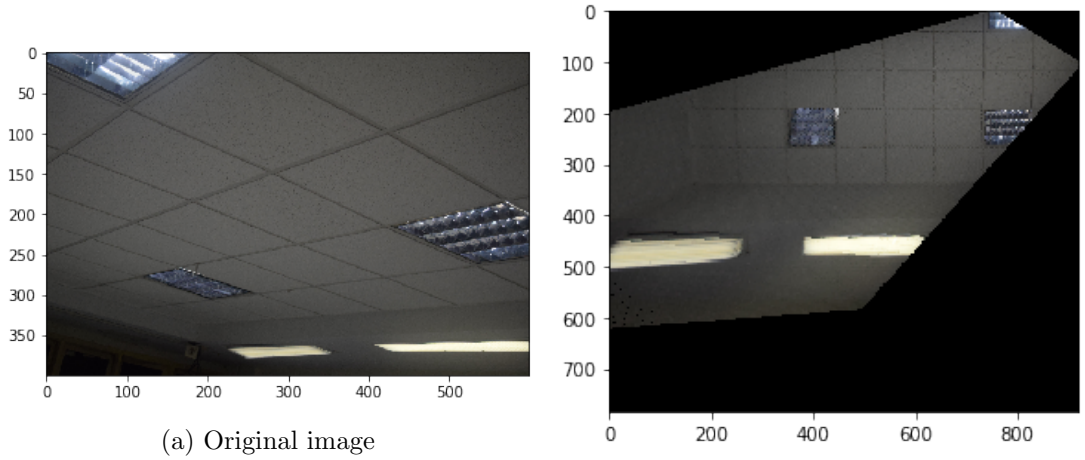
The following are the original-rectified image pairs obtained by using `cv2.warpPerspective()` function:



(a) Original image

(b) Rectified image

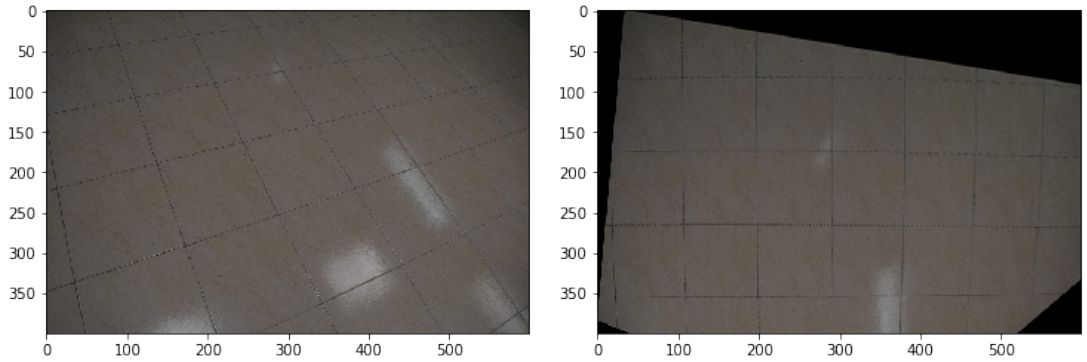
Figure 2: Rectification of Image1.JPG using the first method



(a) Original image

(b) Rectified images

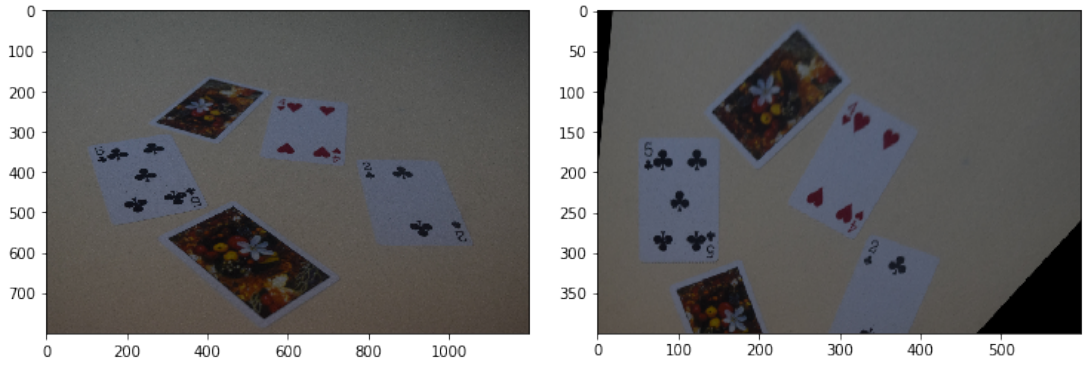
Figure 3: Rectification of Image2.JPG using the first method



(a) Original image

(b) Rectified image

Figure 4: Rectification of Image3.JPG using the first method



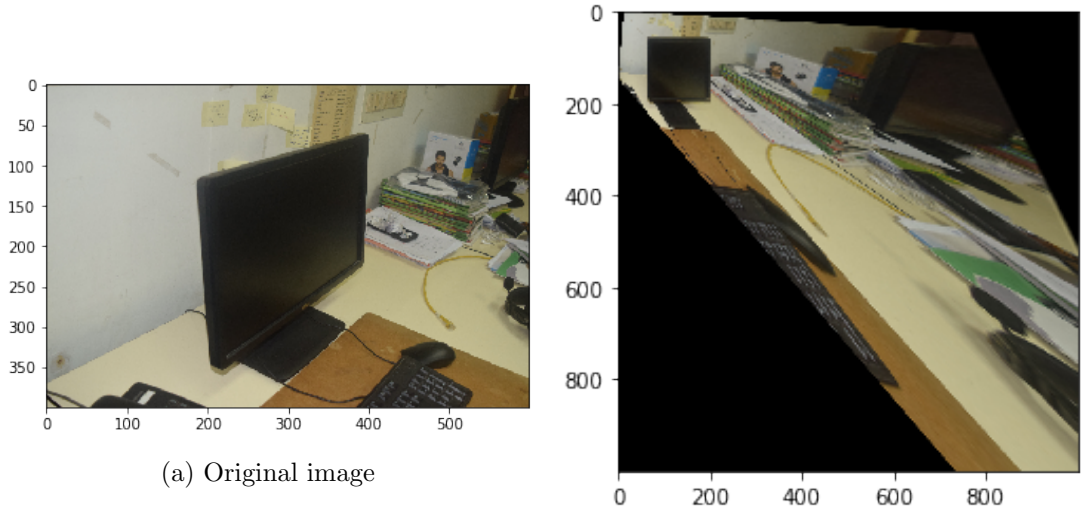
(a) Original image

(b) Rectified image

Figure 5: Rectification of Image4.JPG using the first method

2.2 Interpreting the results

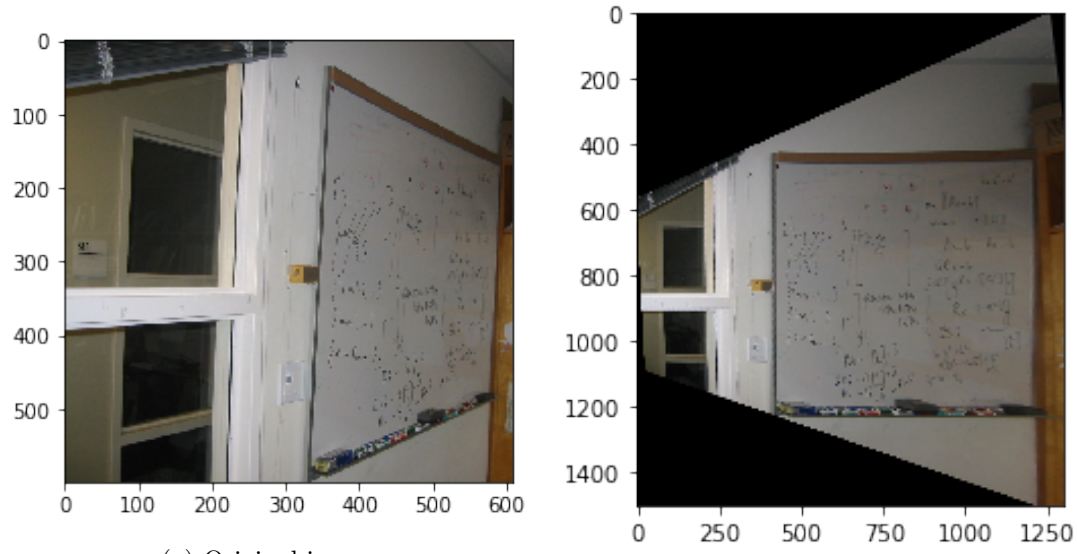
The rectified images for `Image2.JPG`, `Image3.JPG` and `Image5.JPG` provide insights into the different distortions that can remain in the result. In `Image2.JPG` , we see that the ceiling lights not in the plane of the tiled roof appears distorted, because of the "top-down" view in the rectified image. In `Image3.JPG`, there is minor radial distortion which might have been due



(a) Original image

(b) Rectified image

Figure 6: Rectification of Image5.JPG using the first method



(a) Original image

(b) Rectified image

Figure 7: Rectification of image6.png using the first method

to camera optics. This cannot be corrected with a homography. In `Image5.JPG`, the plane of computer monitor was considered while rectifying the image, and thus the keyboard and other objects are now stretched out and distorted.

3 Homography estimation using the dual conic

Similarity rectification of an image can be done as follows :

- Computing the line at infinity : We first calculate the transformed line at infinity l_∞ Since parallel lines meet at l_∞ , one only needs to compute the intersection of two pairs of parallel lines :

$$\overrightarrow{p_1} = \overrightarrow{l_1} \times \overrightarrow{l_2}$$

$$\overrightarrow{p_2} = \overrightarrow{l_3} \times \overrightarrow{l_4}$$
$$\mathrm{l}'_\infty = \overrightarrow{p_1} \times \overrightarrow{p_2}$$

- The image is now affine rectified using the Homography H given by:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \qquad \text{where } \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix} = l'_\infty$$

- Computing the dual conic : On applying a affine transform

$$H_A = \begin{bmatrix} KR & \overrightarrow{t} \\ 0 & 1 \end{bmatrix}$$

  The dual conic to the circular points C^*_∞ transforms to

$$C^{*'}_\infty = \begin{bmatrix} KK^T & 0 \\ 0 & 0 \end{bmatrix}$$

  Now, this matrix would have 4 unknowns. These can be found using the angle condition on a pair of orthogonal lines:

$$\overrightarrow{l_i}^T C^{*'}_\infty \overrightarrow{m_i} = 0 \qquad i = 1, 2$$

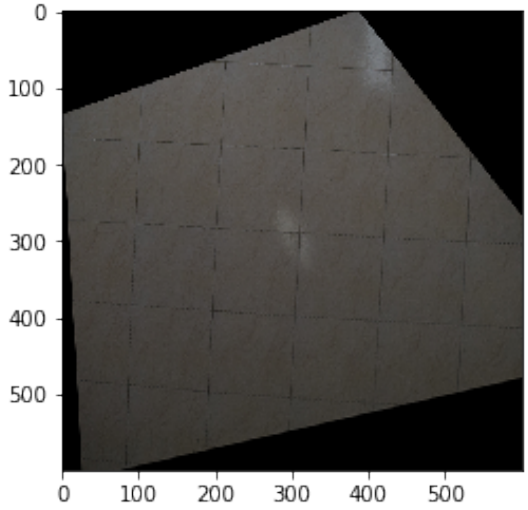
  where l_i, m_i is a pair of (originally) orthogonal lines. Once the matrix of $C^{*'}_\infty$ is found, we can obtain K using Cholesky decomposition (using `scipy.linalg.cholesky`) or use SVD.

3.1 Implementation

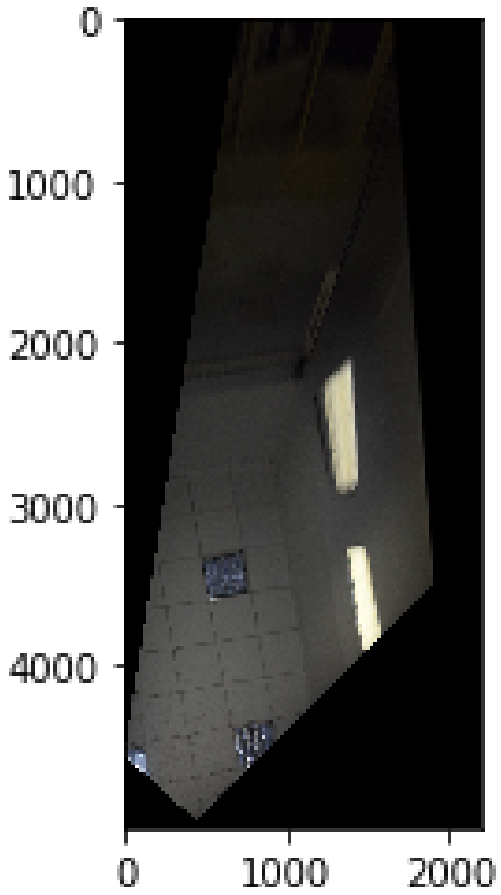
The choice of the pair of perpendicular lines is crucial in obtaining the desired solution. After affine rectification, the squares/rectangles in the original image would now be parallelograms. One angle constraint on a corner of the parallelogram would in fact ensure that all adjacent sides are perpendicular (This can be proved using geometry and affine transform properties). Thus, one should consider angle at the diagonals (for a square) or the angle of intersection of sides of a different parallelogram (for a rectangle).

3.2 Results

The results obtained for the six images are shown in the next page. For the sake of brevity, the original images have not been displayed again. Please refer to the previous section for the same. Some of the images have not been cropped and thus may not look aesthetically pleasing. This has been done only to highlight the complete effects of the transformation. In these as well as the later results, for better visualization, some of the images have been subjected to additional similarity transformations (reflection, scaling, rotation).

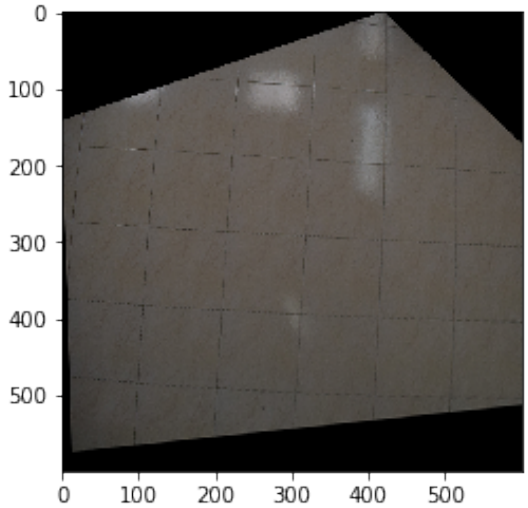


(a) Rectified image of Image1.JPG

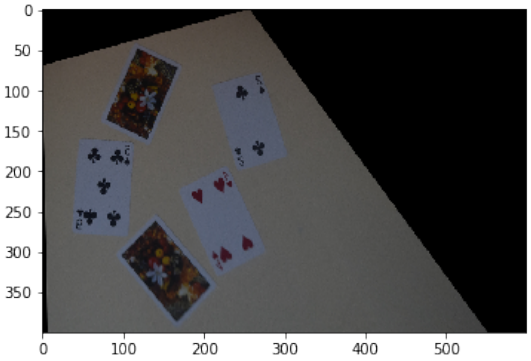


(b) Rectified image of Image2.JPG

Figure 8: Image rectification using the dual conic of circular points



(a) Rectified image of Image3.JPG

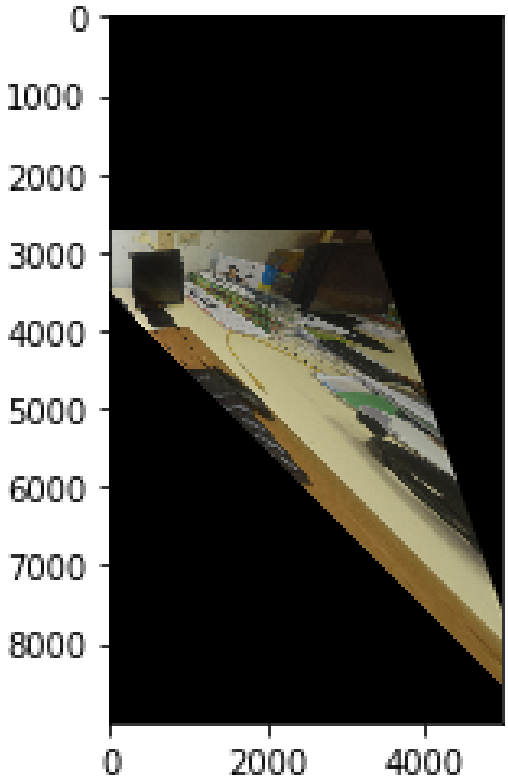


(b) Rectified image of Image4.JPG

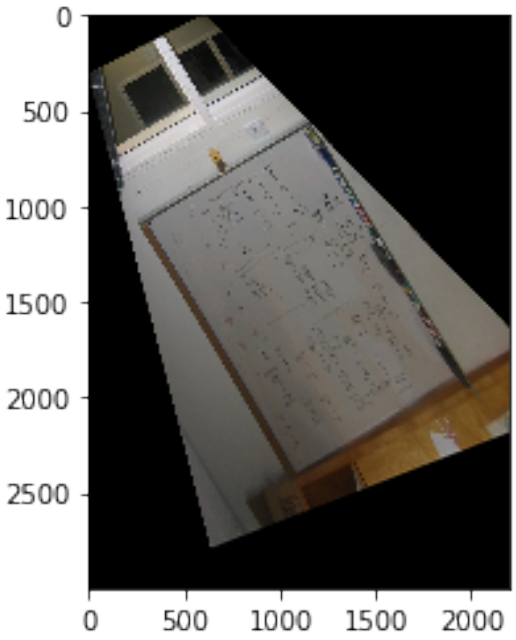
Figure 9: Image rectification using the dual conic of circular points

4 Homography estimation using five perpendicular lines

One can estimate the homography H using five perpendicular lines as follows:



(a) Rectified image of Image5.JPG



(b) Rectified image of Image6.png

Figure 10: Image rectification using the dual conic of circular points

- The transformed dual conic $C_\infty^{*\prime}$ has 6 parameters (5 upto scale).
- Using the angle criterion on five pairs of perpendicular lines, we have

$$\overrightarrow{l_i}^T C_\infty^{*\prime} \overrightarrow{m_i} = 0 \qquad\qquad i = 1, 2, 3, 4, 5$$

- If the flattened version of the matrix is $\overrightarrow{c}$, then the co-efficients of the above equations can be stacked to form the equation

$$A_{5\times 6}\overrightarrow{c} = \overrightarrow{0}$$

  $\overrightarrow{c}$ is thus the vector spanning the nullspace of A .
- Once we obtain $\overrightarrow{c}$ (and thus $C_\infty^{*\prime}$) , we can find the homography H as follows :

$$C = UDV^T$$
$$\text{For} \quad \text{D} = \text{diag}(\text{d}_1, d_2, d_3),$$
$$\text{Let} \quad \text{D'} = \text{diag}(\sqrt{d_1}, \sqrt{d_2}, 1)$$

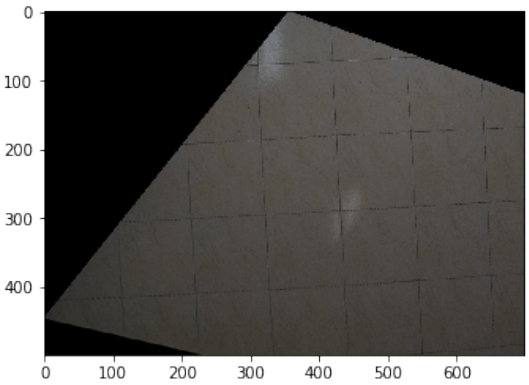
  Then

$$H = UD' \tag{1}$$

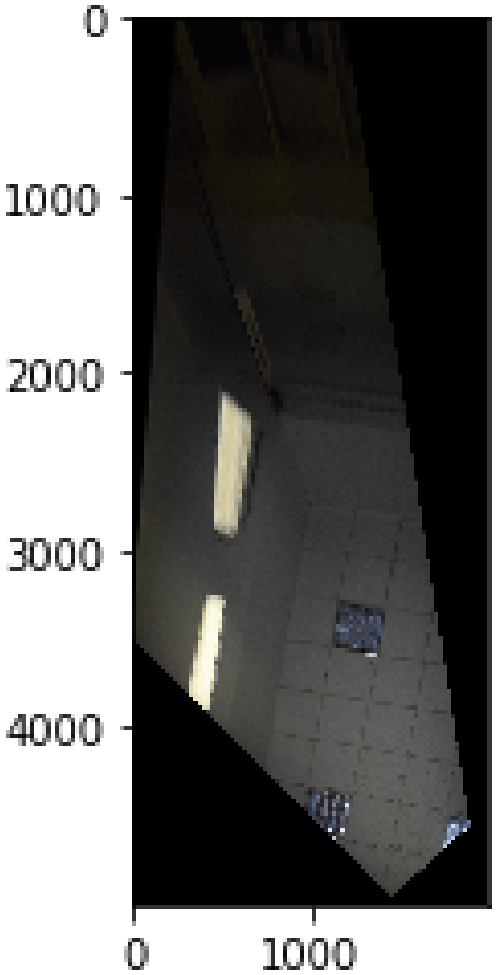
  where the first step is the SVD of C.

4.1 Results

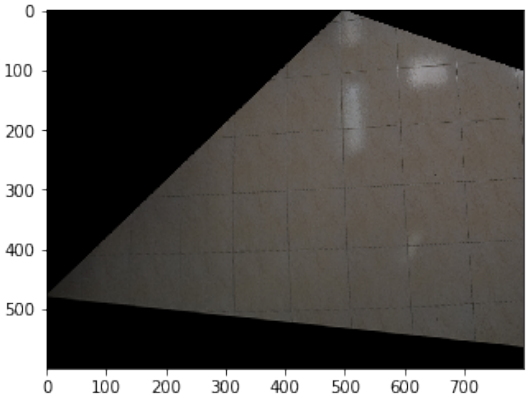
The following are the results obtained using this method:



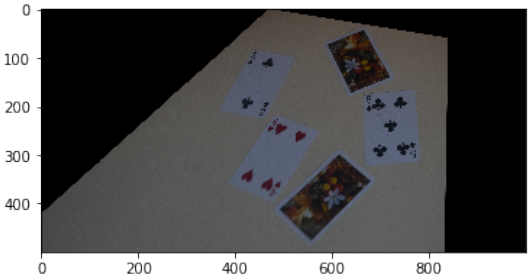
(a) Rectified image of Image1.JPG



(b) Rectified image of Image2.JPG



(c) Rectified image of Image3.JPG

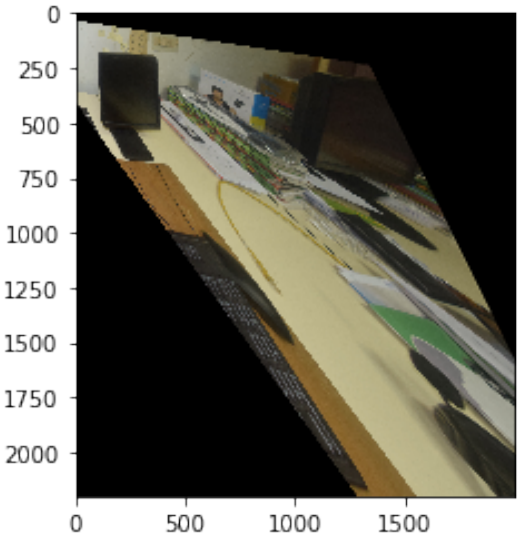


(d) Rectified image of Image4.JPG

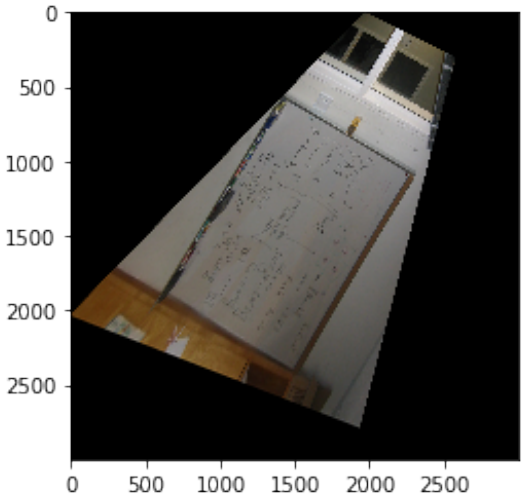
Figure 11: Image rectification using the five perpendicular lines

4.2 Interpreting the results

Images like `Image5.JPG` have undergone unwanted anisotropic scaling. One can also observe that `Image4.JPG` has not been rectified completely. One possible reason for these problems is that the "right" lines were not chosen in order to obtain the desired rectification.



(a) Rectified image of Image5.JPG



(b) Rectified image of Image6.png

Figure 12: Image rectification using five perpendicular lines

5 Homography estimation using a transformed circle

- We first perform affine rectification by finding the line at infinity.
- Every circle intersects the line at infinity l_∞ at the circular points I and J. An affine transformation of a circle will, in general, yield an ellipse. We first compute the equation of the ellipse using five constraints.
- The ellipse is now intersected with l_∞ to yield two points I' and J'. These would be the transformed circular points
- The transformed dual conic $C_\infty^{*'}$ is computed as :

$$C_\infty^{*'} = I'(J')^T + J'(I')^T$$

- Using $C_\infty^{*'}$, one can find H, as in 1 .

5.1 Implementation

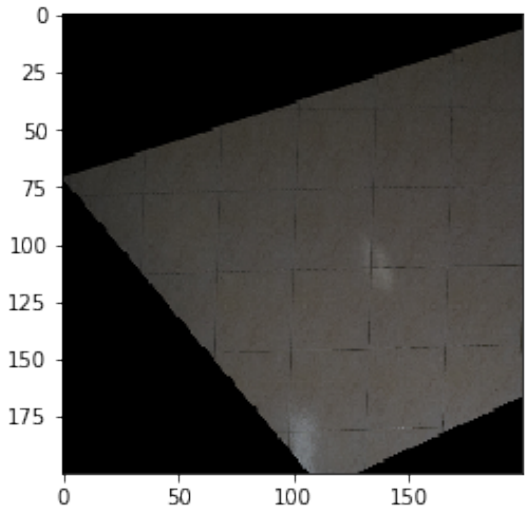
The above algorithm works when we have atleast 5 constraints on the ellipse. In all of the images, we can only work with squares/rectangles. Thus, when we consider the transformed square/rectangle, we obtain 4 constraints on the parameters of the ellipse (corresponding to four corners). Since infinite number of ellipses can pass through four points, we can as well choose a random point close to the transformed square and use this as the fifth constraint.

5.2 Results

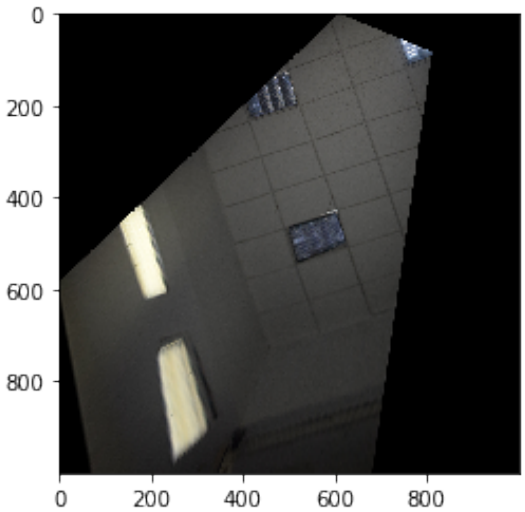
The results obtained are shown in the next page.

5.3 Interpreting the results

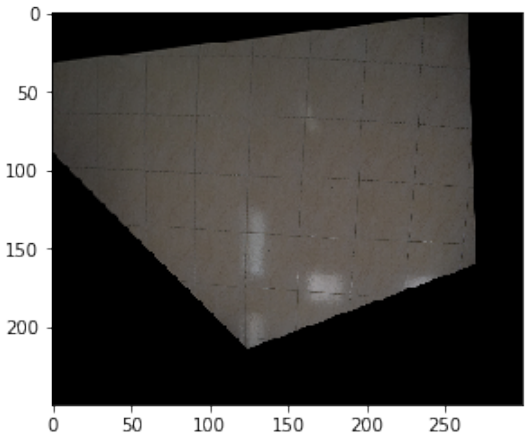
The above results have more abnormalities than the previous methods. This is because of the fact that the random point chosen might have been sub-optimal. For example,



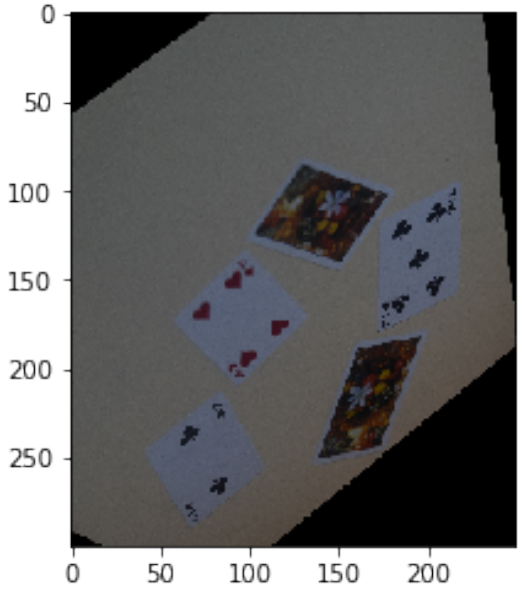
(a) Rectified image of Image1.JPG



(b) Rectified image of Image2.JPG



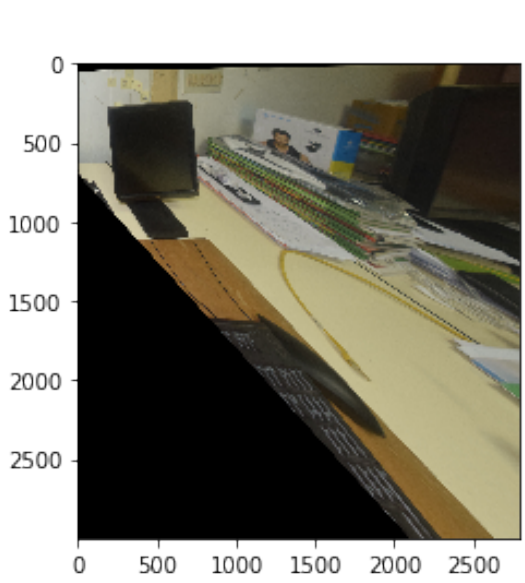
(c) Rectified image of Image3.JPG



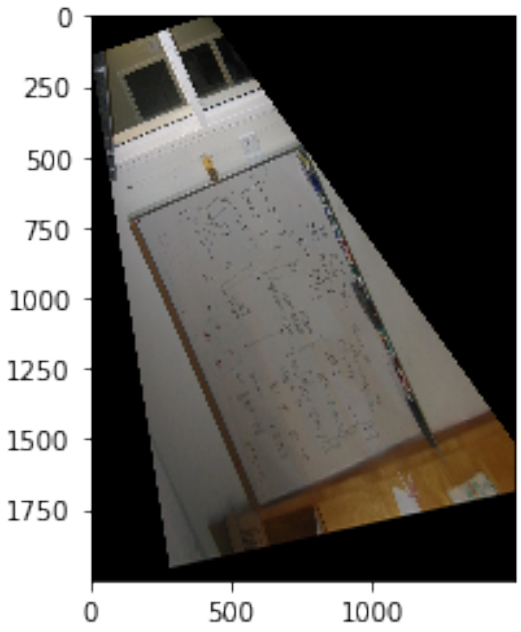
(d) Rectified image of Image4.JPG

Figure 13: Image rectification using the dual conic of circular points

different points led to different anisotropic scaling effects for `Image1.JPG`. A similar effect is observed in `Image6.JPG` .Some of the above results can thus be improved by performing a better search over possible points. One can also observe that in `Image4.JPG` some cards are still distorted, which could again be solved by choosing a better point.



(a) Rectified image of Image5.JPG



(b) Rectified image of Image6.png

Figure 14: Image rectification using the dual conic of circular points