Assignment -1

1 Problem-1

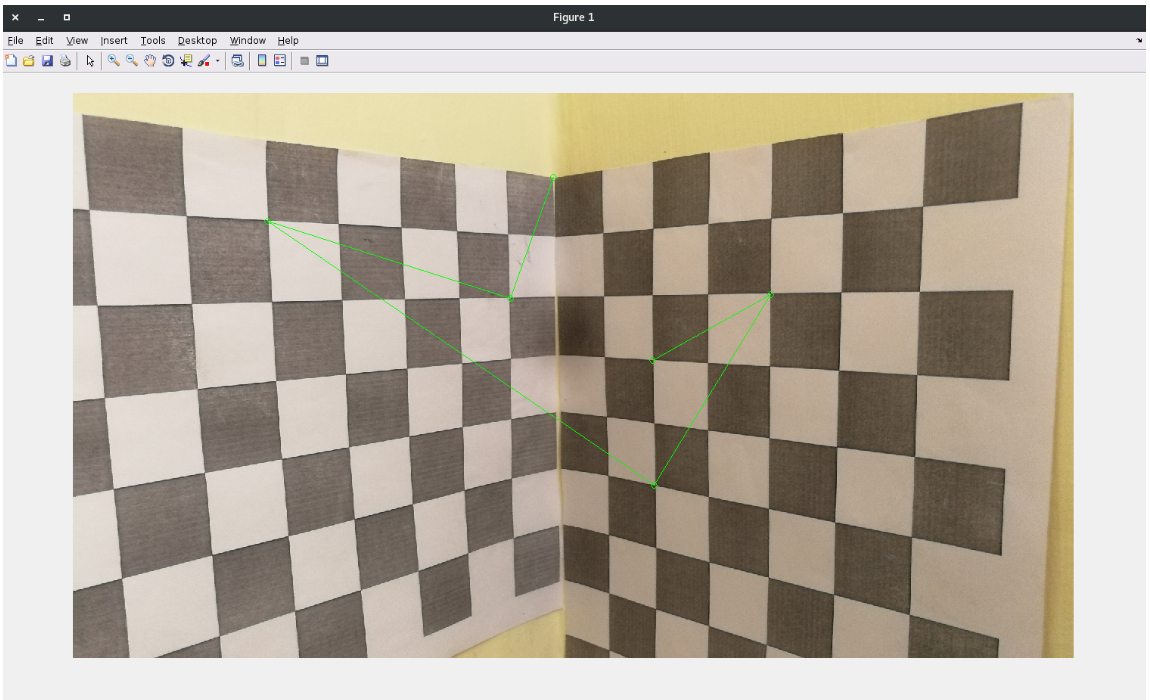


Figure 1: Green line shows the selected points for camera calibration

The 6 marked points (connected by green lines) in the figure are taken for camera calibration. Pixel location and spatial coordinates are measured and the method discussed in class using Direct Linear Transform is applied to the data.

Need for Normalization:
The DLT (Direct Linear Transform) method tries to use SVD estimation of $A = UDV^T$ to obtain a solution to the system of of equations $A\mathbf{h} = 0$. If we had exact data and infinite precision then we could find exact solution to the equation without need of SVD by finding null space of A. So we try to minimize the $||A\mathbf{h}||$ (subject to $||\mathbf{h}|| = 1$). This is in a way equivalent to finding a singular matrix $\hat{A}$ closest to A in norm and obtaining exact solution to $||\hat{A}\mathbf{h}|| = 0$

This approach is not invariant to similarity transforms. Let object coordinates be (x, y, z, w) and image coordinates be represented as (x^*, y^*, w^*). Then in matrix A the magnitude of the terms xx^*, yy^*, yx^*, xy^* could be much larger than the other terms like xw^* because x,y coordinates can be much greater than w, since in our coordinate system, w is 1 if x, y or z are real world coordinates and this causes large differences in the magnitudes of the values of these coordinates. Replacing A by $\hat{A}$ means that some entries of A must be increased and some should be decreased (to keep norm of A and $\hat{A}$ nearly equal).

So this is why normalization is important, a change of 100 in term xx^* would be small change but in xw^* term change of 100 would be of same magnitude as number itself and not feasible. If all coordinates are normalized and scaled properly this will not occur.

The Code:
We used a script we found a function online that gives us the image coordinates of the point we click on in the image and draws a line joining the consecutive points. It is present in the file readPoints.m. The initial part of the code in myMainScript.m (which has been commented out) calls this function and stores the values in 'loctions.mat'. The object coordinates have been measured using the length of the side of each block of the checkerboard as one unit.

We got a root mean square error of 0.2748 when the image pixel locations were of the order of the third power of 10. You can check this value by uncommenting the comment block present at the end of myMainScript.m file. It would also plot the following image containing the actual image points as well as the predicted image points.

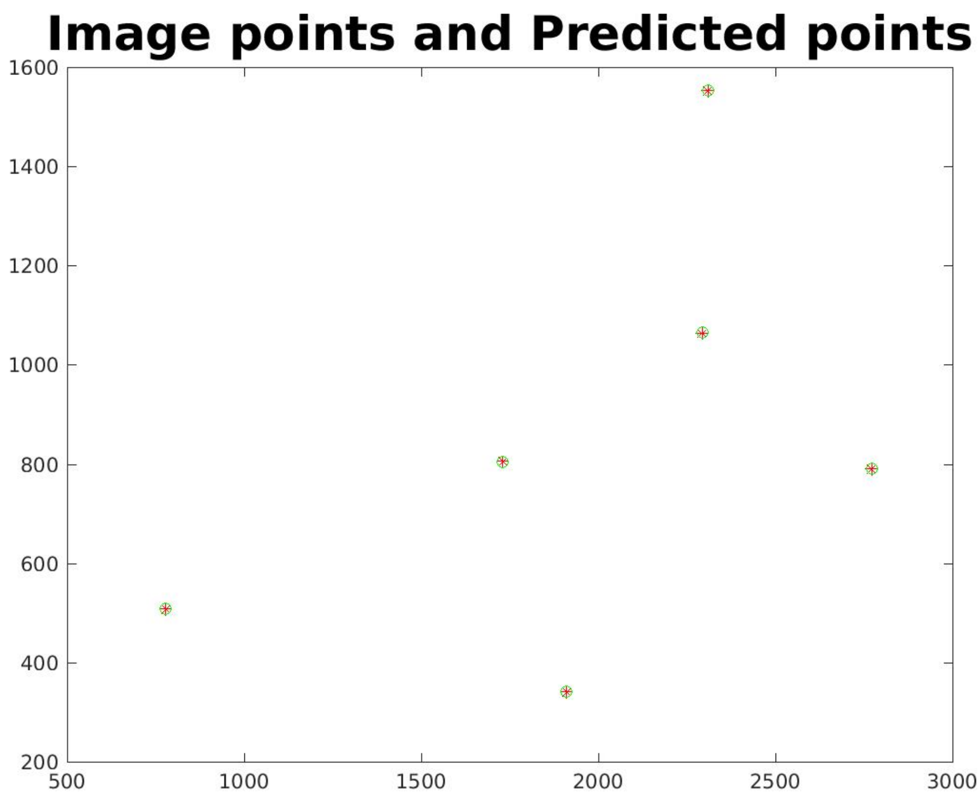


Figure 2: The original selected points are marked in red and our predictions are in green

2 Problem-2

As per the given equation, if we have $\vec{x_d} = (x_d, y_d, 1)$ and $\vec{x_u} = (x_u, y_u, 1)$, then:

$$x_d = x_u * (1 + q_1 r + q_2 r^2), y_d = y_u * (1 + q_1 r + q_2 r^2)$$

The second (matrix) equation gives us:

$$x_u = x_d + \Delta x(\vec{x_d}, \vec{q}), y_u = y_d + \Delta y(\vec{x_d}, \vec{q})$$

Substituting the values $x_u = \frac{x_d}{(1+q_1 r+q_2 r^2)}$ and $y_u = \frac{y_d}{(1+q_1 r+q_2 r^2)}$, we get:

$$\Delta x(\vec{x_d}, \vec{q}) = x_d(\frac{1}{(1 + q_1 r + q_2 r^2)} - 1)$$

$$\Delta y(\vec{x_d}, \vec{q}) = y_d(\frac{1}{(1 + q_1 r + q_2 r^2)} - 1)$$

We then substitute this to get the matrix with which to multiply $\vec{x_d}$ to get $\vec{x_u}$. However, in this matrix the value of r is calculated by using the coordinated of the undistorted image, that is, $\vec{x_u}$. Since this is not known initially, we start by using the value of $\vec{x_d}$ in its place and run iterations, updating the value of $\vec{x_u}$ at every iteration. This method converges quickly.
The resulting image using this method is:

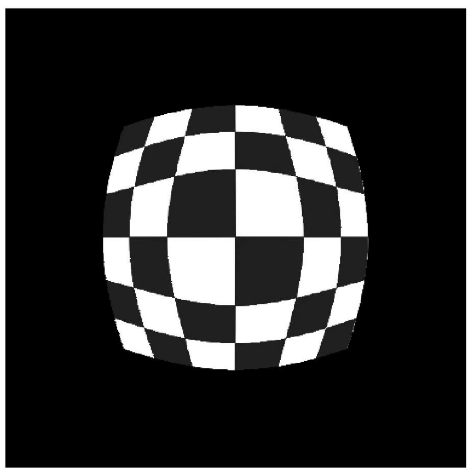


Figure 3: This is the recovered image found after iterations

However, this image is more distorted (in the subjective sense of the word) than the one given to us. Nonetheless, we feel this is the correct image because we have written code (at the bottom of the submitted script) to recover the distorted image from the undistorted one and that gives the following image, which is identical to the image given to us in the questions. (The black lines are because we round the non-integral coordinates in the destination image to the closest integral ones). If we do not assume that the original ("undistorted") image is a perfect checkerboard, this method has no inconsistencies.

We also tried other approaches like the one given below. Those approaches assumed that the undistorted image is actually undistorted (in subjective sense of word).
We iterate over the pixels in the undistorted image, for each pixel in undistorted image we find out to which value would it map to in the distorted image. Consider a pixel p in the undistorted image: it is at a distance r from the center. Due to the distortion it gets mapped to a point at a distance $\frac{r}{1+q1*r+q2*r^2}$. So we map the intensity $I_{undistorted}(p) = I_{distorted}(\frac{\vec{p}}{1+q1*r+q2*r^2})$. This method seems intuitive but the image burst out of desired image range. Here we use $\vec{x_u}$ to calculate

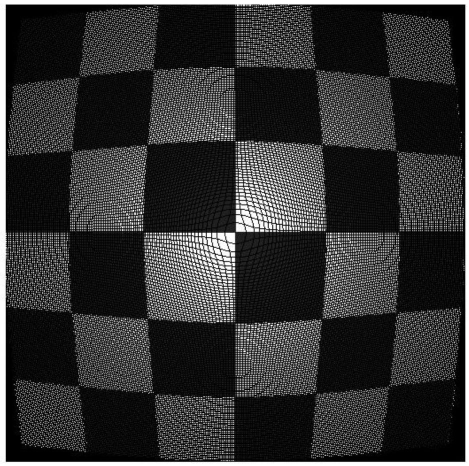


Figure 4: This is obtained on distorting the recovered Image

the value of r, as required by the problem.

GENERAL DOUBT: If we assume the first equation to hold in the case that the undistorted image is actually a uniform checkerboard, then a point radially farther away in the undistorted image, should move even farther away in the distorted image, that is, the corners should move even farther away from the centre as compared to points on the axes. This should give a distorted image opposite to the one given to us. This would be a case of pincushion distortion rather than barrel distortion.

3 Problem-3

Parametric Euclidean Coordinates of point are given by $\boldsymbol{x} = (t, 1/t)$ and the corresponding homogeneous coordinates are given by $\mathbf{x} = (t, 1/t, 1)$
We are given the transformation matrix M =

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Applying transformation to the homogeneous coordinates we get

$$\mathbf{x'} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t \\ 1/t \\ 1 \end{bmatrix}$$

$$\mathbf{x'} = \begin{bmatrix} 1 \\ 1/t \\ t \end{bmatrix}$$

So $\mathbf{x'} = (1, 1/t, t)$ which is equivalent to $(1/t, 1/t^2, 1)$. Converting the homogeneous coordinates back to euclidean coordinates we get $\boldsymbol{x'} = (1/t, 1/t^2)$ which is a parametric equation of parabola $y = x^2$
Shape of image is **parabola** given by the equation $y = x^2$

4 Problem-4

For sake of compact equations let

$f_p = f_1$, $S_1 = S_p$, $(P_{ix}, P_{iy}) \equiv (P_{1ix}, P_{1iy})$

$f_2 = f_q$, $S_2 = S_q$, $(q_{ix}, q_{iy}) \equiv (P_{2ix}, P_{2iy})$

let the center of 2 images be (C_{1x}, C_{1y}) and (C_{2x}, C_{2y})

Since we know the 3 vanishing pt corresponding to mutually $\perp$ lines in each image, we can find the line joining vanishing pt and center in each image as:

$$m_{ij} = \left(S_j (P_{jix} - C_{jx}),\ S_j (P_{jiy} - C_{jy}),\ f_j \right)$$

i represents the i^{th} vanishing pt $\in \{1,2,3\}$

j represent the j^{th} image

the unit vector along each lines can be found as

$$\hat{m}_{ij} = \frac{m_{ij}}{norm(m_{ij})}$$

Now, we know unit line in one image is related to unit line (corresponding) in othe image throug a Rotation (R). i.e $\hat{m}_{i1} = R \hat{m}_{i2} \quad \forall i$

i.e $\left[\hat{m}_{11} \mid \hat{m}_{21} \mid \hat{m}_{31} \right] = R \left[\hat{m}_{12} \mid \hat{m}_{22} \mid \hat{m}_{32} \right]$

R can be found through inverting; assuming all other parameters are known. The translation information is not required and not available in this process hence we can not recover t, since we only have points at infinity.

Now, for the case where we don't know the intrinsic parameters, we can apply the following property of the image:

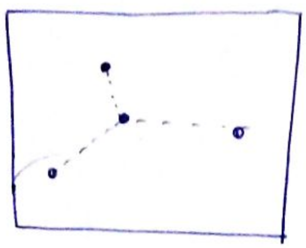


image center $\equiv$ orthocenter of Δ formed by vanishing pts of 3 mutully $\perp$ lines.

Now, to figure out (C_{jx}, C_{jy}), we know that

$$\hat{m}_{k_1,j} \perp \hat{m}_{k_2,j} \quad \forall\ k_1 \neq k_2 \quad \text{i.e } \hat{m}_{k_1,j} \cdot \hat{m}_{k_2,j} = 0$$

to find C_{jx}, C_{jy}

$$S_1^2 (P_{11x} - C_{1x})(P_{12x} - C_{1x}) + S_1^2 (P_{11y} - C_{1y})(P_{12y} - C_{1y}) + f_1^2 = 0$$

$$\Rightarrow \left(\frac{S_1}{f_1}\right)^2 \left(\text{term1} \right) + 1 = 0 \quad \left(\frac{S_1}{f_1} \text{ can be evaluated} \right)$$

Let $\frac{S_1}{f_1} = \lambda_1$ from above equation (i.e $S_1 = \lambda_1 f_1$)

Similarly $S_2 = \lambda_2 f_2$

$$\therefore \hat{m}_{ij} = \frac{(\lambda_j (P_{jix} - C_{jx}), \lambda_j (P_{jiy} - C_{jy}), 1)\cancel{f_j}}{\cancel{f_j} \sqrt{\text{term\_independent\_of } f_j}} \quad \left(\text{from previous equations}\right)$$

hence all directions can be found even if we don't know f_j

5 Problem-5

a)

Consider a coordinate system such that the image plane is parallel to the x-y plane and the image plane is located at $z = 0$. Now, let L_1 and L_2 be two parallel lines in R^3 in this coordinate system. Since they are parallel if $L_1 = \vec{A} + \lambda\vec{D}$, then L_2 will be of the form $\vec{B} + \lambda\vec{D}$.
Given our coordinate sytem, using the projection formula, if $\vec{X} = (X_x, X_y, X_z)$ is a point on the line L_1 in R^3 and $\vec{x} = (x_x, x_y)$ is its projection in the image plane, we have:

$$\vec{x} = \frac{\vec{X}}{X_z} = \frac{\vec{A} + \lambda\vec{D}}{A_z + \lambda D_z}$$

Now, the vanishing point for this line will be when λ tends to infinity, and hence if v_1 is the vanishing point for the line L_1, we have:

$$\vec{v_1} = \lim_{\lambda\to\infty} \frac{\vec{A} + \lambda\vec{D}}{A_z + \lambda D_z} = \frac{\vec{D}}{D_z} = (\frac{D_x}{D_z}, \frac{D_y}{D_z})$$

Similarly, for line L_2, the projection of any point in the image plane will be given by $\frac{\vec{B}+\lambda\vec{D}}{B_z+\lambda D_z}$ and the vanishing point will be:

$$\vec{v_2} = \lim_{\lambda\to\infty} \frac{\vec{B} + \lambda\vec{D}}{B_z + \lambda D_z} = \frac{\vec{D}}{D_z} = (\frac{D_x}{D_z}, \frac{D_y}{D_z})$$

Hence, we see that the two parallel lines have the same vanishing point and hence, we prove that they intersect and their point of intersection is the vanishing point.

b)

Let the three non-parallel vectors in R^3 be represented by $\vec{D_1}, \vec{D_2}, \vec{D_3}$.
As we found in the previous question (i.e., part (a)), the vanishing points for lines parallel to these vectors will be given by:

$$\vec{v_1} = (\frac{D_{1x}}{D_{1z}}, \frac{D_{1y}}{D_{1z}})$$

$$\vec{v_1} = (\frac{D_{2x}}{D_{2z}}, \frac{D_{2y}}{D_{2z}})$$

$$\vec{v_1} = (\frac{D_{3x}}{D_{3z}}, \frac{D_{3y}}{D_{3z}})$$

In homogenous coordinates, this will become,

$$\vec{v_1} = (D_{1x}, D_{1y}, D_{1z}) = \vec{D_1}$$

$$\vec{v_2} = (D_{2x}, D_{2y}, D_{2z}) = \vec{D_2}$$

$$\vec{v_3} = (D_{3x}, D_{3y}, D_{3z}) = \vec{D_3}$$

In homogenous coordinates, the line passing though $\vec{v_1}$ and $\vec{v_2}$ will be $\vec{(l)} = \vec{v_1} \times \vec{v_2} = \vec{D_1} \times \vec{D_2}$
Now to prove that the line $\vec{l}$ passes through the point $\vec{v_3}$, we need to show that their dot product is zero, i.e., $\vec{l} \cdot \vec{v_3} = 0$ This is the same as

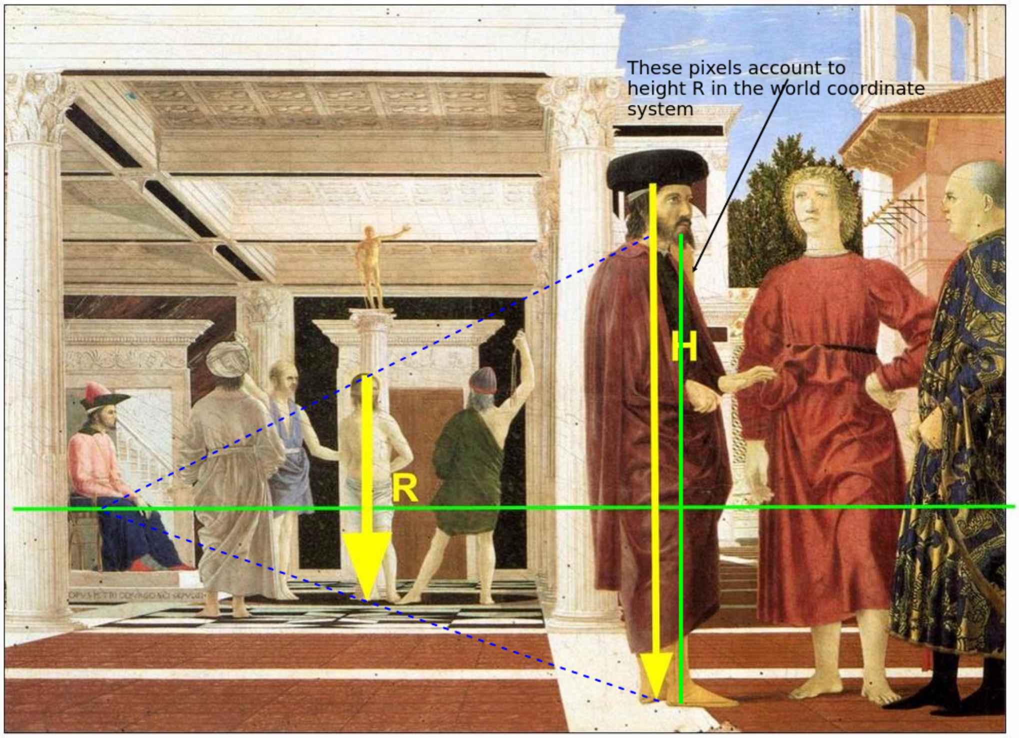
$$(\vec{D_1} \times \vec{D_2}) \cdot \vec{D_3} = [\vec{D_1}\vec{D_2}\vec{D_3}]$$

where $[\vec{D_1}\vec{D_2}\vec{D_3}]$ is the vector triple product and since it has been given that the three sets of lines are planar, their vector triple product will be zero. Therefore,

$$\vec{l} \cdot \vec{v_3} = 0$$

and hence we prove that the vanishing points for all are collinear.

6 Problem-6



We make the line joining the feet of the two people and find its point of intersection with the horizon. The line joining the feet (L1) will be parallel to ground, and as the green line is horizon all lines parallel to this line meet a that point of intersection. Now we mark the line L2 joining the vanishing point and head of person in left and further extend it to intersect with person on right.

The point of intersection of L2 with the person on right will be at a height 'R' above the ground in the world coordinate system (L1 and L2 and the ground are parallel and so height above the ground on that line remains constant). Essentially, we draw a parallel line from Christ's head to "project" Christ onto the same spot as the man whose height we want to estimate.

We have 4 points on the line vertically upwards (in the image) from the foot of the person on right namely, the foot of person (A), the intersection of the line and person(B), head of person on right(C), and infinity(D). So we can compute the cross ratio in both frames and equate them (as it is a projective transform) :

$$\frac{H}{R} = \frac{number\ of\ pixels\ from\ ground\ to\ head\ of\ person}{number\ of\ pixels\ from\ ground\ to\ point\ of\ intersection}$$

Note that in this case, in both the image and the ground frame of reference, AD and BD are both infinite and hence the ratio simplifies to AC/BC. We use the ratio AC/AB which will also be the same for both reference frames.
Solving this

$$\mathbf{H} = 200.0942$$

The Code :
In this problem we noted down the pixel locations of points by using the readPoints.m function we used in the first problem. After noting down the pixel coordinates it is simple problem involving similar triangles and cross ration as described above. In myMainScript.m we have performed the computation for H and on running the answer is displayed