Computer Vision, Assignment 1
Elements of Projective Geometry

1 Instructions

In this assignment you will study the basics of projective geometry. You will study the representations of points lines and planes, as well as transformations and camera matrices.

Please see Canvas for detailed instructions on what is expected for a passing/higher grade. All exercises not marked **OPTIONAL** are “mandatory” in the sense described on Canvas.

The maximum amount of points for the theoretical exercises in Assignment 1 is 34. 50% of 34 is 17.

2 Points in Homogeneous Coordinates.

Theoretical Exercise 1 (5 points). What are the 2D Cartesian coordinates of the points with homogeneous coordinates

$$\mathbf{x}_1 = \begin{pmatrix} 4 \\ -16 \\ 2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} -3 \\ 7 \\ -1 \end{pmatrix} \text{ and } \mathbf{x}_3 = \begin{pmatrix} 9\lambda \\ -3\lambda \\ 6\lambda \end{pmatrix}, \lambda \neq 0? \tag{1}$$

What is the interpretation of the point with homogeneous coordinates

$$\mathbf{x}_4 = \begin{pmatrix} -6 \\ 3 \\ 0 \end{pmatrix}? \tag{2}$$

Is $\mathbf{x}_4$ the same point as

$$\mathbf{x}_5 = \begin{pmatrix} -4 \\ 3 \\ 0 \end{pmatrix}? \tag{3}$$

For the report: Answers are enough.

Computer Exercise 1. Write a matlab function `pflat` that takes as input an array of shape (m, n), representing n points in $\mathbb{P}^{m-1}$ in homogeneous coordinates. The output should be an (m, n)-array containing the same homogeneous points, but where each point is normalized so that its last coordinate is 1. (You may assume that none of the points have last homogeneous coordinate zero.) Apply the function to the points in `x2D` and `x3D` in the file `compEx1.mat`, and plot the result. It will be useful for later to also create a function `plot_points_2D` that takes as input a $(2, n)$-array representing n two dimensional points in Cartesian coordinates and plots them. And a similar `plot_points_3D` for three dimensional points.

```
Useful matlab commands:

x(end,:) %Extracts the last row of x

a./b %Elementwise division.
%Divides the elements of a by the corresponding element of b.

plot(a(1,:),a(2,:),'.')  %Plots a point at (a(1,i),a(2,i)) for each i.

plot3(a(1,:),a(2,:),a(3,:),'.') %Same as above but 3D.

axis equal %Makes sure that all axes have the same scale.
```

For the report: Submit the m-files, and the plots.

3 Lines

Theoretical Exercise 2 (5 points). Compute the homogeneous coordinates of the intersection (in $\mathbb{P}^2$) of the lines

$$l_1 = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} \text{ and } l_2 = \begin{pmatrix} 6 \\ 3 \\ 1 \end{pmatrix}. \tag{4}$$

What is the corresponding point in $\mathbb{R}^2$?

Compute the intersection (in $\mathbb{P}^2$) of the lines

$$l_3 = \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix} \text{ and } l_4 = \begin{pmatrix} 5 \\ 0 \\ 4 \end{pmatrix}. \tag{5}$$

What is the geometric interpretation in $\mathbb{R}^2$?

Compute the line that goes through the points with Cartesian coordinates

$$x_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \text{ and } x_2 = \begin{pmatrix} 6 \\ 3 \end{pmatrix}. \tag{6}$$

Hint: Re-use the calculations from the line intersections above.

For the report: Submit a complete solution.

Theoretical Exercise 3 (3 points). The nullspace of a $m \times n$ matrix A is the set

$$\mathcal{N}(A) = \{x \in \mathbb{R}^n; Ax = 0\}, \tag{7}$$

that is, all the x for which the multiplication Ax gives the zero vector. Explain why the intersection point (in homogeneous coordinates) of l_1 and l_2 (from Exercise 2) is in the null space of the matrix

$$M = \begin{pmatrix} 6 & 3 & 1 \\ -1 & 1 & 1 \end{pmatrix}. \tag{8}$$

Are there any other $\mathbb{P}^2$-points in the null space besides the intersection point?

For the report: Full solution.

Computer Exercise 2. Load and plot the image in `compEx2.jpg` (see figure 1).

Figure 1: compEx2.jpg

In the file `compEx2.mat` there are three pairs of image points. Plot the image points in the same figure as the image.

For each pair of points compute the line going through the points. Use the function `rital` to plot the lines in the same image. Do these lines appear to be parallel (in 3D)?

Compute the point of intersection between the second and third line (the lines obtained from the pairs `p2` and `p3`). Plot this point in the same image.

The distance between a 2D-point $x = (x_1, x_2)$ in Cartesian coordinates and a line $l = (a, b, c)$ can be computed using the distance formula

$$d = \frac{|ax_1 + bx_2 + c|}{\sqrt{a^2 + b^2}}, \tag{9}$$

see your linear algebra book. Implement a function `point_line_distance_2D` that computes the distance between a point and a line. Compute the distance between the first line and the the intersection point. Is it close to zero? Why/why not?

```
Useful matlab commands:

imread('compEx2.jpg') %Loads the image compEx2.jpg

imagesc(im) %Displays the image

colormap gray %changes the colormap of the current image to gray scale

hold on %Prevents the plot command from clearing the figure before plotting

hold off %Makes the plot command clear the figure before plotting

null(A) %computes the nullspace of A
```

For the report: Answer all the questions. Submit the m-files, the plot, and the computed distance between the intersection point and the first line.

4 Projective Transformations

Theoretical Exercise 4 (5 points). Let H be the projective transformation

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & -1 & 1 \end{pmatrix} \tag{10}$$

Compute the transformations $\mathbf{y}_1 \sim H\mathbf{x}_1$ and $\mathbf{y}_2 \sim H\mathbf{x}_2$ if

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \text{ and } \mathbf{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}. \tag{11}$$

Compute the lines l_1, l_2 containing $\mathbf{x}_1$,$\mathbf{x}_2$ and $\mathbf{y}_1$,$\mathbf{y}_2$ respectively.

Compute $(H^{-1})^T l_1$ and compare to l_2.

For the report: Submit the answers.

Theoretical Exercise 5 (4 points). Prove that every projective transformation H preserves lines. That is, for each line l_1 there is a corresponding line l_2 such that if x belongs to l_1 then the transformation $\mathbf{y} \sim H\mathbf{x}$ belongs to l_2. (Hint: If $l_1^T\mathbf{x} = 0$ then $l_1^T H^{-1}H\mathbf{x} = 0$.)

For the report: Submit the full (short) proof.

Theoretical Exercise 6 (6 points). Take a look at the following transformation matrices.

$$H_1 = \begin{pmatrix} \sqrt{3} & 0 & 1 \\ 1 & -\sqrt{3} & -1 \\ 0 & 0 & 1 \end{pmatrix}, \; H_2 = \begin{pmatrix} 1 & -\sqrt{3} & 1 \\ \sqrt{3} & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}, \tag{12}$$

$$H_3 = \begin{pmatrix} \sqrt{5} & 1 & 1 \\ -1 & \sqrt{5} & 1 \\ 1/2 & 1/4 & 1 \end{pmatrix} \text{ and } H_4 = \begin{pmatrix} 1 & -5 & 2 \\ 0 & 3 & 0 \\ \sqrt{3} & 0 & 2\sqrt{3} \end{pmatrix}, \tag{13}$$

(a) Which of the transformations are projective transformations?

(b) Which are affine transformations?

(c) Which are similarity transformations?

(d) Which are Euclidean?

(e) Which of the transformations preserve lengths between points?

(f) Which map lines to lines?

(g) Which map parallel lines to parallel lines?

For the report: Answers to the questions.

5 The Pinhole Camera

Theoretical Exercise 7 (6 points). Compute the projections of the 3D points with homogeneous coordinates

$$\mathbf{X}_1 = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \end{pmatrix}, \ \mathbf{X}_2 = \begin{pmatrix} 1 \\ 1 \\ 3 \\ 1 \end{pmatrix} \text{ and } \mathbf{X}_3 = \begin{pmatrix} 2 \\ 1 \\ -1 \\ 1 \end{pmatrix}, \tag{14}$$

in the camera with camera matrix

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}. \tag{15}$$

What is the interpretation of the projection of $\mathbf{X}_1$?

Compute the camera center (position) of the camera and the principal axis.

For the report: Answers are enough.

Computer Exercise 3. Load and plot the images `compEx3im1.jpg` and `compEx3im2.jpg` (see figure 2). The file `compEx3.mat` contains the camera matrices `P1, P2` and a point model `U` of the statue.

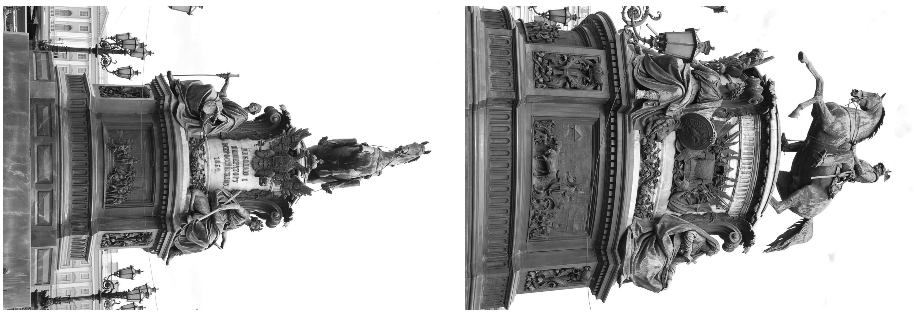

Figure 2: compEx3im1.jpg and compEx3im2.jpg

In order to visualize cameras in 3D it is useful to plot them as vectors pointing from the camera center in the direction of the principal axis. Implement a function `camera_center_and_axis` that takes a camera matrix as input and outputs the camera center and principal axis. The camera center should be in Cartesian coordinates and the principal axis should be normalized to length 1. Compute the camera centers and principal axes of the two cameras.

Next, implement a function `plot_camera` which takes a camera matrix and a scale s as input and plots the principal axis of the camera scaled by s, from the camera center. Plot the 3D-points in `U` and the camera centers in the same 3D plot (make sure that the 4th coordinate of `U` is one before you plot by using `pflat`).

Project the points in `U` into the cameras `P1` and `P2` and plot the result in the same plots as the images. Does the result look reasonable?

```
Useful matlab commands:

null(P) %computes the nullspace of P

P(3,1:3) %extracts elements P31, P32 and P33.

quiver3(a(1),a(2),a(3),v(1),v(2),v(3),s) %Plots a vector v starting from
                                          %the point a and rescales the size by s

plot(x1(1,:),x1(2,:),'.','Markersize',2);%Same as plot but with smaller points
```

For the report: Submit the m-files, the plots, the camera centers in Cartesian coordinates, and the principal axes normalized to length one.

Theoretical Exercise 8. (**OPTIONAL**, 10 optional points) Consider the calibrated camera pair $P_1 = [I\ 0]$ and $P_2 = [R\ t]$. If $\mathbf{x} \in \mathbb{P}^2$ is the 2D projection in P_1 of the 3D point $\mathbf{U} \in \mathbb{P}^3$, i.e. $\mathbf{x} \sim P_1\mathbf{U}$, verify that

$$\mathbf{U} \sim \begin{pmatrix} \mathbf{x} \\ s \end{pmatrix}, \tag{16}$$

where $s \in \mathbb{R}$. That is, for any s the point of the form $\mathbf{U}(s) = (\mathbf{x}^T, s)^T$ projects to $\mathbf{x}$. What kind of object is this collection of points? Is it possible to determine s using only information from P_1?

Assume that $\mathbf{U}$ belongs to the plane

$$\Pi = \begin{pmatrix} \pi \\ 1 \end{pmatrix}, \tag{17}$$

where $\pi \in \mathbb{R}^3$. Compute the s that makes $\mathbf{U}(s)$ belong to the plane, that is, find s such that $\Pi^T\mathbf{U}(s) = 0$.

Verify that if $\mathbf{x} \sim P_1\mathbf{U}$, $\mathbf{y} \sim P_2\mathbf{U}$ and $\Pi^T\mathbf{U} = 0$ then the homography

$$H = (R - t\pi^T), \tag{18}$$

where $P_2 = [R\ t]$, maps $\mathbf{x}$ to $\mathbf{y}$. (Hint: What is $P_2\mathbf{U}(s)$ for the s from above?)

For the report: Provide all answers, computations and verifications.

Computer Exercise 4. (**OPTIONAL**, 15 optional points) Figure 3 shows an image (`compEx4.jpg`) of a poster. (Here the axes units are pixels.)

The file `compEx4.mat` contains the inner parameters `K`, the corner points of the poster and the 3D plane `v` that contains the poster. The camera matrix for the camera that generated this image is

$$P_1 = K[I\ 0]. \tag{19}$$

The goal of this exercise is to create an image of the poster taken by a camera 2.5m to the right of the original camera.

Start by plotting the corner points and the image in the same 2D-figure. (Use `axis equal` to get the correct aspect ratio.) Note the scale on the axis. Where is the origin of the image coordinate system located?

To be able to use the formulas derived in Exercise 8 we must first ensure that we have calibrated cameras. To do this normalize the corner points by multiplying with K^{-1} and plot them in a new 2D-figure. (Use `axis` ij to make the y-axis point downwards (as for the previous image) and `axis equal`.

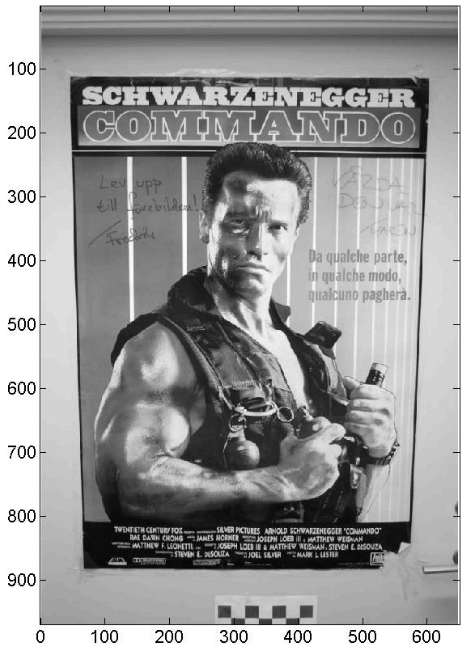

Figure 3: Arnold looking mean.

Note the difference in scale compared to the previous figure. Where is the origin of the image coordinate system located?

Since the inner parameters K have been removed our calibrated camera is $[I\ 0]$. Using the results from Exercise 8 compute the 3D points in the plane `v` that project onto the corner points. Compute the camera center and principal axis, and plot together with the 3D-points. Does it look reasonable?

Next compute a new camera with camera center in $(-2, 0, 0)$ (2 units to the left of P_1) and orientation

$$R = \begin{pmatrix} \cos(\pi/6) & 0 & -\sin(\pi/6) \\ 0 & 1 & 0 \\ \sin(\pi/6) & 0 & \cos(\pi/6) \end{pmatrix} \tag{20}$$

(30 degrees rotation around the y-axis. Note that the y-axis points down in the first image.) Compute the new camera and plot in the same figure. (Don't forget to use `axis equal`.)

Compute the homography in Exercise 8 and transform the normalized corner points to the new (virtual) image. Plot the transformed points (don't forget to divide by the third coordinate) in a new 2D-figure. Does the result look like you would expect it to when moving the camera like this? Also project the 3D points into the same image using the camera matrix. Does this give the same result?

We now have a homography that takes normalized points in the first camera and transforms them to normalized points in the second camera. To remove the need for normalization we simply include the normalization in the homography. If $\tilde{\mathbf{x}} = K^{-1}\mathbf{x}$ and $\tilde{\mathbf{y}} = K^{-1}\mathbf{y}$ then for the normalized points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ we have

$$\tilde{\mathbf{y}} \sim H\tilde{\mathbf{x}} \Leftrightarrow K^{-1}\mathbf{y} \sim HK^{-1}\mathbf{x} \Leftrightarrow \mathbf{y} \sim KHK^{-1}\mathbf{x} \tag{21}$$

Therefore the total transformation is $H_{tot} = KHK^{-1}$. Transform the original image and the corner points using the homography H_{tot} and plot both in a new 2D-figure.

```
Useful matlab commands:

plot(corners(1,[1:end 1]), corners(2,[1:end 1]),'*-');
%Plots the cornerpoints and connects them with lines.
```

```
axis ij %Makes the y-axis point down (as in an image)

%tform = projective2d(Htot.'); % pre MATLAB 2022b
tform = projtform2d(Htot); % recommended post MATLAB 2022b
%Creates a projective transformation that can be used in imwarp

[im_new, RB] = imwarp(im, tform);
%Creates a transformed image (using tform)
%of the same size as the original one.
%RB is an imref2d object which
%"stores the relationship between the intrinsic coordinates
%anchored to the rows and columns of a 2-D image and the
%spatial location of the same row and column locations
%in a world coordinate system." (see documentation)

imshow(im_new, RB);
%plots the new image in the correct coordinate system
```

For the report: Provide answers, plots and submit an m-file creating the transformed image.